Applying NLP Technologies to the Collection and Enrichment of Language Data on the Web to Aid Linguistic Research

Fei Xia
University of Washington
Seattle, WA 98195, USA
fxia@u.washington.edu

William D. Lewis
Microsoft Research
Redmond, WA 98052, USA
wilewis@microsoft.com

Abstract

The field of linguistics has always been reliant on language data, since that is its principal object of study. One of the major obstacles that linguists encounter is finding data relevant to their research. In this paper, we propose a three-stage approach to help linguists find relevant data. First, language data embedded in existing linguistic scholarly discourse is collected and stored in a database. Second, the language data is automatically analyzed and enriched, and language profiles are created from the enriched data. Third, a search facility is provided to allow linguists to search the original data, the enriched data, and the language profiles in a variety of ways. This work demonstrates the benefits of using natural language processing technology to create resources and tools for linguistic research, allowing linguists to have easy access not only to language data embedded in existing linguistic papers, but also to automatically generated language profiles for hundreds of languages.

1 Introduction

Linguistics is the scientific study of language, and the object of study is language, in particular *language data*. One of the major obstacles that linguists encounter is finding data relevant to their research. While the strategy of word of mouth or consulting resources in a library may work for small amounts of data, it does not scale well. Validating or reputing key components of a linguistic theory realistically requires analyzing data across a large sample of languages. For instance, in linguistic typology a well-known implicational universal states that if the demonstrative follows the noun, then the relative clause also follows the noun (Croft, 2003). Although this particular universal is well-researched and widely accepted, identifying this tendency anew—as an example of what one must do when researching a new universal—would require a significant amount of work: in order to be relatively sure that the universal holds, the linguist would need to identify a substantial number of true positives (those that support the universal), and ensure that there are not a sufficient number of negatives that would act as a refutation. The only way a linguist could be completely sure would be to conduct a thorough literature review on the subject or go through data from a representative and significant sample of data from the approximately seven thousand languages that are or have been spoken (and for which data exists).

There have been much effort by the linguistic community to address the issue. For instance, LinguistList compiles a long list of linguistic resources[1], making it easier to find electronically available resources. Likewise, the Open Language Archives Community (OLAC) acts as an online virtual library of language resources, and provides a search tool that searches several dozen online linguistic resources. Further, the World Atlas of Language Structures (WALS), which was recently made available online, is a large database of structural (phonological, grammatical, lexical) properties of languages gathered from descriptive materials (Haspelmath et al., 2005).[2]

[1] http://www.linguistlist.org/langres/index.html

[2] There are other online resources for searching for linguistic data, in particular typological data. Two of note include Autotyp (Bickel and Nichols, 2002) and the Typological Database System (Dimitriadis et al., forthcoming), among others. The former has limited online availability (much of

Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education –LaTeCH – SHELT&R 2009, pages 51–59,
Athens, Greece, 30 March 2009. ©2009 Association for Computational Linguistics

We propose a three-stage approach to help linguists in locating relevant data. First, language data embedded in existing linguistic scholarly discourse is collected and stored in a database. Second, the language data is automatically analyzed and enriched and language profiles are created from the enriched data. Third, a search facility is provided to allow linguists to search the original data, the enriched data, and the language profiles.

This is an on-going research project. While the first stage is completed, the second and third stages are partially completed and still undergoing development. In this paper, we will describe each stage and report results.

2 Related work

In this section, we briefly discuss a few projects that are most relevant to our work.

2.1 Ethnologue

The purpose of the Ethnologue is to provide a comprehensive listing of the known living languages of the world. The most recent version, version 15, covers more than six thousand languages. Information in the Ethnologue comes from numerous sources and is confirmed by consulting both reliable published sources and a network of field correspondents, and has been built to be consistent with ISO standard 639-3; the information is compiled under several specific categories (e.g., countries where a language is spoken and their populations) and no effort is made to gather data beyond those categories (Gordon, 2005).

2.2 WALS

The World Atlas of Language Structures (WALS) is a large database of structural (phonological, grammatical, lexical) properties of languages gathered from descriptive materials (such as reference grammars) by a team of more than 40 linguists (Haspelmath et al., 2005). WALS consists of 141 maps with accompanying text on diverse features (such as vowel inventory size, noun-genitive order, passive constructions, and *hand/arm* polysemy). Each map corresponds to a feature and the map shows the feature values for between 120 and 1370 languages. Altogether there are 2,650 languages and more than 58,000 data points; each data point is a (language, feature, feature value) tuple that specifies the value of the feature in a particular language. For instance, *(English, canonical word order, SVO)* means that the canonical word order of English is SVO.

2.3 OLAC

The Open Languages Archive Community (OLAC), described in (Bird and Simons, 2003), is part of the Open Archives Initiative, which promotes interoperability standards for linguistic data.[3] The focus of OLAC has been to facilitate the discovery of linguistic resources through a common metadata structure for describing digital data and by providing a common means for locating these data through search interfaces housed at Linguist List and the Linguistics Data Consortium (LDC). Our work shares with OLAC the need for resource discovery, and moves beyond OLAC by enriching and manipulating the content of linguistic resources.

3 Building ODIN

The first stage of the three-stage approach is to collect linguistic data and store it in a database. In linguistics, the practice of presenting language data in interlinear form has a long history, going back at least to the time of the structuralists. Interlinear Glossed Text, or *IGT*, is often used to present data and analysis on a language that the reader may not know much about, and is frequently included in scholarly linguistic documents. The canonical form of an IGT consists of three lines: a *language line* for the language in question, a *gloss line* that contains a word-by-word or morpheme-by-morpheme gloss, and a *translation line*, usually in English. The grammatical markers such as *3sg* on the gloss line are called *grams*. Table 1 shows the beginning of a linguistic document (Baker and Stewart, 1996) 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.

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

(1) Crawling: crawling the Web to retrieve documents that may contain IGTs

the data is not directly accessible through query, but requires submitting requests to the site owners), however, and the latter is still under development.

[3] http://www.language-archives.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

(2) IGT detection: extracting IGTs from the retrieved documents

(3) Language ID: identifying the language code of the extracted IGTs.

The identified IGTs are then extracted and stored in a database (the ODIN database), which can be easily searched with a GUI interface.[4] In this section, we briefly describe the procedure, and more detail about the procedure can be found in (Xia and Lewis, 2008) and (Xia et al., 2009).

3.1 Crawling

In the first step, linguistic documents that may contain instances of IGT are harvested from the Web using metacrawls. Metacrawling involves throwing queries against an existing search engine, such as Google and Live Search, and crawling only the pages returned by those queries. We found that the most successful queries were those that used strings contained within IGT itself (e.g. grams such as 3sg). In addition, we found precision increased when we included two or more search terms per query, with the most successful queries being those which combined grams and language names.

Other queries we have developed include: queries by language names and language codes (drawn from the Ethnologue database (Gordon, 2005), which contains about 40,000 language names and their variants), by linguists names and the languages they work on (drawn from the Linguist Lists linguist database), by linguistically relevant terms (drawn from the SIL linguistic glossary), and by particular words or morphemes found in IGT and their grammatical markup.

3.2 IGT detection

The canonical form of IGT consists of three parts and each part is on a single line. However, many IGT instances, 53.6% of instances in ODIN, do not follow the canonical form for various reasons. For instance, some IGTs are missing gloss or translation lines as they can be recovered from context (e.g., other neighboring examples or the text surrounding the instance); some IGTs have multiple translations or language lines (e.g., one part in the native script, and another in a latin transliteration); still others contain additional lines of annotation and analysis, such as phonological alternations, underlying forms, etc.

We treat IGT detection as a sequence labeling problem. First, we train a learner and use it to label each line in a document with a tag in a pre-defined tagset. The tagset is an extension of the standard BIO tagging scheme and it has five tags: they are *BL* (any blank line), *O* (outside IGT that is not a BL), *B* (the first line in an IGT), *E* (the last line in an IGT), and *I* (inside an IGT that is not a B, E, or BL). After the lines in a document are tagged by the learner, we identify IGT instances by finding all the spans in the document that match the "B [I | BL]* E" pattern; that is, the span starts with a B line, ends with an E line, and has zero or more I or BL lines in between.

To test the system, we manually annotated 51 documents to mark the positions of the IGTs. We trained the system on 41 documents (with 1573 IGT instances) and tested it on 10 documents (with 447 instances). The F-score for exact match (i.e., two spans match iff they are identical) was 88.4%, and for partial match (i.e., two spans match iff they overlap), was 95.4%. The detail of the system can be found in (Xia and Lewis, 2008).

3.3 Language ID

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 several 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. For

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

instance, Cavnar and Trenkle's N-gram-based algorithm produced an accuracy of as high as 99.8% when tested on newsgroup articles in eight languages (Cavnar and Trenkle, 1994). However, when we ran the same algorithm on the IGT data, the accuracy fell as low as 2% when the training set was very small.

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).[5] 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, language code) pair 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 Markov Logic Network (Richardson and Domingos, 2006). Both systems outperform existing, general-purpose language identification algorithms significantly. The detail of the algorithm and experimental results is described in (Xia et al., 2009).

3.4 The current ODIN database

We ran the IGT detection and language ID systems on three thousand IGT-bearing documents crawled from the Web and the extracted IGTs were stored in the ODIN database. Table 2 shows the language distribution of the IGT instances in the database according to the output of the language ID system. For instance, the third row says that 122 languages each have 100 to 999 IGT instances, and the 40,260 instances in this bin account for 21.27% of all instances in the ODIN database.[6]

Table 2: Language distribution of the IGTs in ODIN

Range of IGT instances	# of languages	# of IGT instances	% of IGT instances
> 10000	3	36,691	19.39
1000-9999	37	97,158	51.34
100-999	122	40,260	21.27
10-99	326	12,822	6.78
1-9	838	2,313	1.22
total	1326	189,244	100

In addition to the IGTs that are already in the ODIN database, there are more than 130,000 additional IGT-bearing documents that have been crawled but have not been fully processed. Once these additional documents have been processed, the database is expected to expand significantly, growing to a million or more IGT instances.

4 Analyzing IGT data and creating language profiles

The second stage of the three-stage approach is to analyze and enrich IGT data automatically, to extract information from the enriched data, and to create so-called *language profiles* for the many languages in the database. A *language profile* describes the main attributes of a language, such as its word order, case markers, tense/aspect, number/person, major syntactic phenomena (e.g., scrambling, clitic climbing), etc.[7]

An example profile is shown below. The profile says that in Yoruba the canonical word order is SVO, determiners appear after nouns, and the language has Accusative case, Genitive case, Nominative case, and so on. The concepts such as AccusativeCase come from the GOLD Ontology (Farrar, 2003; Farrar and Langendoen, 2003).

```
<Profile>
  <language code="WBP">Yoruba</language>
  <ontologyNamespace prefix="gold">
     http://linguistic-ontology.org/gold.owl#
  </ontologyNamespace>
  <feature="word_order"><value>SVO</value></feature>
  <feature="det_order"><value>NN-DT</value></feature>
  <feature="case">
     <value>gold:AccusativeCase</value>
     <value>gold:GenitiveCase</value>
     <value>gold:NominativeCase</value>
                    . . .
</Profile>
```

Given a set of IGT examples for a language, the procedure for building a profile for the language has several steps:

(1) Identifying and separating out various fields

[5] A language code is a 3-letter code that *uniquely* identifies a language. In contrast, the mapping between language name and a language is not always one-to-one: some languages have multiple names, and some language names map to multiple languages.

[6] Some IGTs are marked by the authors as ungrammatical (usually with an asterisk "*" at the beginning of the language line). These IGTs are kept in ODIN because they may contain information useful to linguists (for the same reason that they were included in the original linguistic documents).

[7] A thorough discussion on the definition and content of language profiles is beyond the scope of the paper. The reader is referred to (Farrar and Lewis, 2006) for more discussion on the topic.

(language data, gloss, translation, citation, construction name, etc.) in an IGT.

(2) Enriching IGT by processing the translation line and projecting the information onto the language line.

(3) Identifying grams in the gloss line and mapping them to the concepts defined in GOLD Ontology or the like.

(4) Answering questions in the language profile.

In this section, we explain each step and report some preliminary results.

4.1 Identifying fields in IGT

In addition to the language data (*L*), gloss (*G*), and translation (*T*) parts of IGT, an IGT often contains other information such as language name (*-LN*), citation (*-AC*), construction names (*-CN*), and so on. An example is in (1), in which the first line contains the language name and citation,[8] the third line includes coindexes *i* and *i/j*, and the last two lines show two possible translations of the sentence. Here, the language line is displayed as two lines due to errors made by the off-the-shelf converter that converted the crawled pdf documents into text.

```
(1) Haitian CF (Lefebvre 1998:165)
                  ak
    Jani   pale         lii/j
    John   speak  with   he
    (a) 'John speaks with him' (b) 'John
        speaks with himself'
```

The goal of this step is to separate out different fields in an IGT, fix display errors caused by the pdf-to-text converter, and store the results in a uniform data structure such as the one in Ex (2) for the example in Ex (1). The task is not trivial partially because the IGT detector marks only the span of an instance. For instance, the coindex *i* in *Jani* and *lii/j* on the third line of Ex (1) could easily be mistaken as being part of the word.

```
(2) Language: Haitian CF
    Citation: (Lefebvre 1998:165)
    L:        Jan   pale   ak     li
    Coindx:   (Jan, i), (li, i/j)
    G:        John  speak  with   he
    T1:       'John speaks with   him'
    T2:       'John speaks with himself'
```

There has been much work on extracting database records from text or semi-structured sources, and the common approach is breaking the text into multiple segments and labeling each segment with a field name (e.g., (Wellner et al., 2004; Grenager et al., 2005; Poon and Domingos, 2007)). Our task here is slightly different from their tasks (e.g., extracting author/title/journal from citations) in that the fields in IGT could overlap[9] and corrupted lines need to be re-constructed and re-stored in a particular way (e.g., pasting the second and third lines in Ex (1) back together).

Due to the differences, we did not create annotated data by segmenting IGT into separate fields and labeling each field. Instead, we used a refined tagset to indicate what information is available at each line of IGT instances. The tagset includes six main tags (*L*, *G*, *T*, etc.) and nine secondary tags (e.g., *-CR* for corruption and *-SY* for syntactic information). Each line in each IGT instance is labeled with one main tag and zero or more secondary tags. The labeled lines in Ex (1) are shown in (3).

```
(3) M-LN-AC:  Haitian CF (Lefebvre 1998:165)
    L-CR:                    ak
    L-SY-CR:  Jani   pale         lii/j
    G:        John   speak  with   he
    T-DB:     (a) 'John speaks with him' (b) 'John
    C:         speaks with himself'
```

The labeling of the data is done semi-automatically. We have created a tool that takes the IGT spans produced by the current IGT detector and labels IGT lines by using various cues in an IGT instance, and designed a GUI that allows annotators to correct the system output easily. The annotation speed is about 320 IGT instances per hour on average. We are currently experimenting with different ways of re-training the IGT detector with the new data.

We have built a rule-based module that identifies fields in IGT using the enriched tagset (i.e., creating Ex (2) from Ex (3)), relying on the knowledge about the conventions that linguists tend to follow when specifying citations, construction names, coindexation and the like. The initial result of field extraction looks promising. We are also studying whether existing unsupervised statistical systems for information extraction (e.g., (Poon and Domingos, 2007)) could be extended to handle this task while taking advantage of the enriched tagset for IGTs. We plan to complete the study and report the results in the near future.

4.2 Enriching IGT

Since the language line in IGT data typically does not come with annotations (e.g., POS tags, phrase

[8] *CF* here stands for French-lexified creole.

[9] For instance, in some IGTs, a syntactic structure is added on top of the language line; for instance, the language line in Ex (1) could become something like *[IP Jani [VP pale [PP ak lii/j]]]*

structures), we developed a method to enrich IGT data and then extract syntactic information (e.g., context-free rules) to bootstrap NLP tools such as POS taggers and parsers. The enrichment algorithm first parses the English translation with an English parser, then aligns the language line and the English translation via the gloss line, and finally projects syntactic information (e.g., POS tags and phrase structures) from English to the language line. For instance, given the IGT example in Ex (4), the enrichment algorithm would produce the word alignment in Figure 1 and the phrase structures in Figure 2. The algorithm was tested on 538 IGTs from seven languages and the word alignment accuracy was 94.1% and projection accuracy (i.e., the percentage of correct links in the projected dependency structures) was 81.5%. Details of the algorithm and the experiments are discussed in (Xia and Lewis, 2007).

```
(4) Rhoddodd yr  athro   lyfr i'r    bachgen ddoe
    gave-3sg the teacher book to-the boy     yesterday
    ''The teacher gave a book to the boy yesterday''
    (Bailyn, 2001)
```

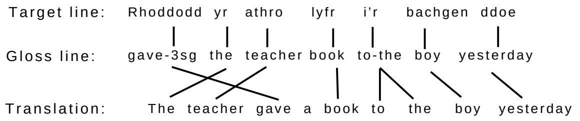
Target line:
Rhoddodd yr athro lyfr i'r bachgen ddoe
Gloss line:
gave-3sg the teacher book to-the boy yesterday
Translation:
The teacher gave a book to the boy yesterday

Figure 1: Aligning the language line and the English translation with the help of the gloss line

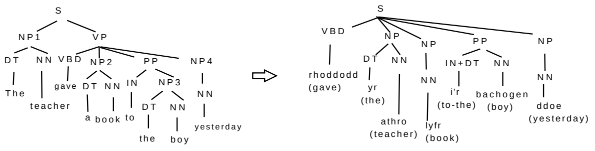

Figure 2: Projecting phrase structure from the translation line to the language line

4.3 Identifying and mapping grams

The third step of Stage 2 identifies grams on the gloss line of an IGT and mapping them to some common semantic so that they can reliably be searched. The gloss line of IGT has two types of glosses: those representing grammatical information (*grams*) such as *NOM*, *3sg*, *PERF*, and standard glosses such as *book* or *give*. Early work in ODIN involved significant manual effort to map grams to GOLD concepts.[10]

The base of several hundred manually mapped grams has provided a reasonably reliable "semantic search" facility in ODIN, which allows linguists to find instances with particular kinds of markup. For example, searching for Perfective Aspect finds instances of data where the data was marked up with PERF, PFV, etc., but also excludes instances that map to "Perfect Tense". While the manually created mapping table covers many common grams, it is far from complete, especially since linguists can coin new grams all the time. We are currently automating the mapping by using the grams in the table as labeled data or seeds and classifying new grams using supervised or semi-supervised methods. This work, however, is still too preliminary to be included in this paper.

4.4 Answering questions in language profiles

The final step of Stage 2 is answering questions in language profiles. Some questions are easier to answer than others. For instance, to determine what grammatical or lexical cases are available in a language according to the data in ODIN, we simply need to look at the grams in the data that map to the case category in GOLD. Other questions are more complex; for instance, to determine whether multiple wh-questions are allowed in a language, we need to examine the projected syntactic structure for the language line and look for the positions of any wh-words that were projected relative to one another. A case study is reported next.

4.5 A case study: Answering typological questions

Two biases are prevalent in IGT data, due to the opportunistic way in which it is harvested and enriched: The first is what we call the *IGT-bias*, that is, the bias produced by the fact that IGT examples are used by authors to illustrate a particular fact about a language, causing the collection of IGT for the language to suffer from a potential lack of representativeness. The second we call the *English-bias*, an English-centrism resulting from the fact that most IGT examples provide an English translation which is used to enrich the language line: as discussed in Section 4.2, the enrichment algorithm assigns a parse tree to the English translation which is then projected onto the langauge line. Since the original parse is built over English data, the projected parse suffers from a bias caused by

[10] See (Lewis, 2006) for more background on mapping grams to GOLD concepts, and (Farrar, 2003) and (Farrar and Langendoen, 2003) for more detailed background on GOLD.

the English source. Because of these biases and errors introduced at various stages of processing, automatically generated language profiles and associated examples should be treated as preliminary and unattested, subject to verification by the linguist. The question is how reliable the profiles are.

To answer the question, we ran a case study in which we evaluated the accuracy of our system in answering a number of typological questions, such as the canonical order of constituents (e.g., sentential word order, order of constituents in noun phrases) or the existence of particular constituents in a language (e.g., determiners). The list of questions and their possible answers are shown in Table 3 (the *WALS #* is a reference number used in WALS (Haspelmath et al., 2005) which uniquely identifies each typological parameter).

In one experiment, we automatically found the answer to the canonical word order question by looking at the context free rules extracted from enriched IGT data. When tested on about 100 languages, the accuracy was 99% for all the languages with at least 40 IGT instances.[12] Not surprisingly, the accuracy decreased for languages with fewer instances (e.g., 65% for languages with 5-9 IGTs). In another experiment, our system answered all the 13 typological questions in Table 3 for 10 languages and the accuracy was 83.1% on average across the questions.

This study shows that, despite potential biases and errors, we can automatically discover certain kinds of linguistic knowledge from IGT with reasonable accuracy and the accuracy increases as more data becomes available. The language profiles built this way could serve as a complement to manually crafted resources such as WALS.

4.6 Comparison with WALS

The task is similar to the goal of the WALS project. In fact, the morphological and syntactic features in WALS form the initial attribute set for our language profiles.[13] The main difference between WALS and our approach is that the information in WALS (including features, feature values, and data points) was gathered by a team of more than 40 linguists, many of them the leading authorities in the field. In contrast, the language profiles in our work are created automatically from opportunistically harvested and enriched linguistic data found on the Web (essentially the IGT in ODIN). Another difference is that our language profiles also include highly language-specific information (e.g., lists of language-specific syntactic constructions, such as *bei*- and *ba*- constructions in Mandarin), as discussed in harvested documents. The information is gathered by checking the construction names included in and surrounding IGT.

The benefits of our approach are twofold. First, we can build language profiles for hundreds of languages with little human effort and the language profiles can be updated whenever the ODIN database is expanded or enriched. Second, each entry in the language profile in ODIN is linked to the relevant IGT instances that are used to answer the question. For instance, a language profile not only lists the canonical word order of the language but also IGT instances from which this information is derived.

5 Extending the search facility

The last stage of the three-stage approach is to provide a search facility for linguists to search the original IGTs, the enriched IGTs and the automatically created language files. The current search interface for ODIN allows a variety of search options, including search by language name or code, language family, and by grams and their related concepts (e.g., Accusative case). Once data is discovered that fits a particular pattern that a user is interested in, he/she can either display the data (where sufficient citation information exists and where the data is not corrupted by the text-to-pdf conversion process) or locate documents from which the data is extracted. Additional search facilities allow users to search across linguistically salient structures ("constructions") and return results in the form of language data and language profiles.

The ODIN database also contains thousands of tree structures for hundreds of languages, each linked to the English tree structures from which they were derived. This can provide unprecedented options for cross-lingual query across "syntactic structures".[14]

[12] Some IGT instances are not sentences and therefore are not useful for answering this question. Further, those instances marked as ungrammatical (usually with an asterisk "*") are ignored for this and all typological questions.

[13] WALS uses the term *feature* to refer to a property such as canonical word order. Since *feature* in NLP has a very different meaning, in this paper we use the term *attribute* instead to avoid potential confusion.

[14] We fully recognize that the projected structures should be considered highly experimental, due to noise in the pro-

Table 3: Thirteen typlogical questions tested in the case study (ndo=no dominant order, nr=not relevant)

Label	WALS #	**Description**	**Possible Values**
Word Order			
WOrder	330	Order of Words in a sentence	SVO,SOV,VSO,VOS,OVS, OSV,ndo[11]
V+OBJ	342	Order of the Verb, Object and Oblique Object (e.g., PP)	VXO,VOX,OVX,OXV,XVO,XOV,ndo
DT+N	N/A	Order of Nouns and Determiners (*a*, *the*)	DT-N, N-DT, ndo, nr
Dem+N	358	Order of Nouns and Demonstrative Determiners	Dem-N, N-Dem, ndo, nr
JJ+N	354	Order of Adjectives and Nouns	JJ-N, N-JJ, ndo
PRP$+N	N/A	Order of possessive pronouns and nouns	PRP$-N, N-PRP$, ndo, nr
Poss+N	350	Order of Possessive NPs and nouns	NP-Poss, NP-Poss, ndo, nr
P+NP	346	Order of Adpositions and Nouns	P-NP, NP-P, ndo
Morpheme Order			
N+num	138	Order of Nouns and Number Inflections (Sing, Plur)	N-num, num-N, ndo
N+case	210	Order of Nouns and Case Inflections	N-case, case-N, ndo, nr
V+TA	282	Order of Verbs and Tense/Aspect Inflections	V-TA, TA-V, ndo, nr
Existence Tests			
Def	154	Do definite determiners exist?	Yes, No
Indef	158	Do indefinite determiners exist?	Yes, No

We plan to extend the current query facility in three steps to allow these structure-based queries. The first step is to do a user study and identify the types of queries that linguists would be interested in. We have already consulted with a number of syntacticians and other linguists, and have compiled a list of “constructions” that would be of the most interest, and plan to consult with more linguists to extend this list.[15] Some of the initial construction queries have already been implemented in ODIN as “prototypes” for testing purposes. The second step is to identify tools that would facilitate implementing these queries. One such tool is *tgrep2*,[16] which is widely used to search treebank style phrase structures. Since the tool is robust and widely used and supported, we plan to extend it to handle the rich data structures found in the enriched IGT data. The third step is to write a large set of queries in tgrep2 (or other query languages) that “pre-package” the most desirable queries into a form that can be easily executed as a Web service, and design a Web GUI that provides the most accessibility to these queries.

6 Conclusion

One of the major obstacles that linguists encounter is finding data relevant to their research. In this paper, we outline a three-stage procedure to alleviate the problem. First, language data embedded in existing linguistic scholarly discourse is collected and stored in the ODIN database. Second, the language data is automatically analyzed and enriched, and language profiles are created from the enriched data. Our case study shows that knowledge discovery (for the targeted attributes) works reasonably well with even a small amount of IGT data. Third, a search facility is provided that allows linguists to search the original data, the enriched data, and the language profiles by language name, language family, and construction names.

There are several directions for future research. We will improve and thoroughly evaluate the module that extracts various fields from IGT. We will also build more complete language profiles for a dozen or so languages for which we have sufficient IGT data and linguistic knowledge to adequately evaluate the results. Finally, we are exploring ways of extending the query facility (e.g., using *tgrep2*) to allow sophisticated search on the original and enriched IGT data, and plan to provide a GUI with pre-packaged queries which will be easy for linguists to use.

Acknowledgements This work has been supported, in part, by NSF grants BCS-0748919 and BCS-0720670 and a RRF grant from the University of Washington. We would also like to thank four anonymous reviewers for their valuable comments.

References

John Frederick Bailyn. 2001. Inversion, Dislocation and Optionality in Russian. In Gerhild Zybatow, editor, *Current Issues in Formal Slavic Linguistics*.

jection algorithms, and the resulting structures still need to be reviewed by the linguist throwing the query. However, our case study demonstrates the reasonably high accuracy of answering typological questions with even very limited supplies of data. This supports their utility in spite of noise and error.

[15] A similar study was discussed in (Soehn et al., 2008).

[16] http://tedlab.mit.edu/˜ dr/TGrep2/

Mark C. Baker and Osamuyimen Thompson Stewart. 1996. Unaccusativity and the adjective/verb distinction: Edo evidence. In *Proceedings of the Fifth Annual Conference on Document Analysis and Information Retrieval (SDAIR)*, Amherst, Mass.

Balthasar Bickel and Johanna Nichols. 2002. Autotypologizing databases and their use in fieldwork. In *Proceedings of the LREC Workshop on Resources and Tools in Field Linguistics*, Las Palmas, Spain, Jun.

Steven Bird and Gary Simons. 2003. Extending dublin core metadata to support the description and discovery of language resources. *Computers and the Humanities*, 17(4):375–388.

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.

William Croft. 2003. *Typology and Universals*. Cambridge University Press, Cambridge, England.

Alexis Dimitriadis, Menzo Windhouwer, Adam Saulwick, Rob Goedemans, and Tams Br. forthcoming. How to integrate databases without starting a typology war: the typological database system. In Simon Musgrave Martin Everaert and Alexis Dimitriadis, editors, *The Use of Databases in Cross-Linguistic Studies*. Mouton de Gruyter, Berlin.

Scott Farrar and D. Terence Langendoen. 2003. A linguistic ontology for the semantic web. *GLOT International*, 7(3):97–100.

Scott Farrar and William D. Lewis. 2006. The GOLD Community of Practice: An infrastructure for linguistic data on the Web. *Language Resources and Evaluation*. Available at http://faculty.washington.edu/wlewis2/papers/FarLew-06.pdf.

Scott Farrar. 2003. *An ontology for linguistics on the Semantic Web*. Ph.d., University of Arizona, May.

Raymond G. Gordon, editor. 2005. *Ethnologue: Languages of the World*. SIL International, Dallas, 15 edition.

T. Grenager, D. Klein, and D. Manning. 2005. Unsupervised learning of field segmentation models for information extraction. In *In Proc. ACL-05*.

Martin Haspelmath, Matthew Dryer David Gil, and Bernard Comrie, editors. 2005. *World Atlas of Language Structures*. Oxford University Press, Oxford.

William 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.

Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI)*, pages 913–918, Vancouver, Canada. AAAI Press.

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

Jan-Philipp Soehn, Heike Zinsmeister, and Georg Rehm. 2008. Requirements of a user-friendly, general-purpose corpus query interface. In *Proceedings of the LREC 2008 Workshop Sustainability of Language Resources and Tools for Natural Language Processing*, Marrakech, Morocco, May 31.

B. Wellner, A. McCallum, F. Peng, and M. Hay. 2004. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proc. of the 20th Conference on Uncertainty in AI (UAI 2004)*.

Fei Xia and William Lewis. 2007. Multilingual structural projection across interlinear text. In *Proc. of the Conference on Human Language Technologies (HLT/NAACL 2007)*, pages 452–459, Rochester, New York.

Fei Xia and William 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, April.