ODIN: A Model for Adapting and Enriching Legacy Infrastructure

William D. Lewis
Department of Linguistics
University of Washington/CSU Fresno
wlewis2@u.washington.edu

Abstract

The Online Database of Interlinear Text (ODIN)[1] *is a database of interlinear text "snippets", harvested mostly from scholarly documents posted to the Web. Although large amounts of language data are posted to the Web as part of scholarly discourse, making the existing "e-Linguistic infrastructure" surprisingly rich, most linguistic data available on the Web exists in legacy formats, is highly display-centric, and is often difficult to locate or interoperate over. ODIN seeks to leverage this existing infrastructure into a rich, searchable, and interoperable resource by converting readily available semi-structured data to content-centric, searchable formats. To do this, ODIN mines scholarly papers and webpages for instances of linguistic data, focusing mostly on interlinear texts, extracts them, identifies source languages, and makes the instances available to search. Through ODIN's standard search feature, users can locate data by language name or Ethnologue code, and display lists of data by document for languages of interest. The newer Advanced Search feature allows users to locate instances by grammatical markup that is used (*e.g.*, NOM, ACC, ERG, PST, 3SG), and by linguistic constructions (*e.g.*, passives, conditionals, possessives, raising constructions, etc.). The latter are made possible through additional enrichment of discovered data using automated statistical taggers and parsers.*

1 The ODIN Vision

The Online Database of Interlinear Text (ODIN) is a database of interlinear text "snippets", harvested mostly from scholarly documents posted to the Web. ODIN was developed as part of the greater effort within the GOLD Community of Practice [10][2] and the Electronic Metastructure for Endangered Languages Data efforts[3], whose goals are to promote best practice standards and software, specifically those that facilitate interoperation over disparate sets of linguistic data. ODIN's genesis came from the realization that despite the fact that significant amounts of language data are being posted and maintained on the Web, there is no uniform search strategy for discovering these data, and most that can be discovered cannot be easily manipulated or used. The *e-Linguistics infrastructure* may be expansive and rich, yet "discovering" language data on the Web often depends on haphazard, low-precision string-based search strategies (using tools such as Google[4] or Yahoo[5]), or even on decidedly low-tech discoveries made by word-of-mouth.[6] In our pursuit of a better way to locate and use language data within the existing infrastructure, we came to realize certain norms in the presentation of data could be tapped for automated discovery and manipulation. One of the more typical semi-structured formats that linguists use is Interlinear Glossed Text, or **IGT**. We conceived of ODIN as a means to locate instances of IGT on the Web by language name and code, such that the linguist doing a search could be reasonably confident that the resources discovered do in fact contain language data of interest. As we built ODIN, we realized that IGT leant itself other types of search, including search over the linguistic annotations used within it. We also realized that because IGT natively contains language data for two languages—a source language and a translation—the application of automated taggers and parsers to the translation, and its alignment with the source, could lead to other types of discovery, moving beyond merely finding data, but actively manipulating and enriching it as part of the search mechanism. ODIN can be seen as a prototype for the linguistic search tools of the future, providing the facility to search across thousands of instances of language data in hundreds of languages, unified into a common format and normalized to a common vocabulary. We see ODIN as a baby-step in the direction of tools that can provide even more utility, going beyond

[1] http://www.csufresno.edu/odin
[2] http://www.linguistics-ontology.org
[3] http://emeld.org
[4] http://www.google.com
[5] http://www.yahoo.com
[6] OLAC, the Open Language's Archive Community, is a major step in the direction of a unified means for language resource discovery. Despite having only 35 archives, resources can be discovered for thousands of the world's languages.

merely finding data, but interacting with it, enriching it, and even providing automated analyses and on-the-fly data comparisons.

2 Background

2.1 Interlinear Glossed Text

Interlinear Glossed Text (IGT) is a common method for encoding and displaying linguistic data, and is often used in scholarly papers to present language data relevant to a particular analysis. It is most commonly presented in a three-line form, a sample of which is shown in (1) below. The first line gives data for the language in question, either phonetically encoded or transcribed in the language's native orthography. The line is broken down into words and morphemes, where words are usually delimited by spaces, and morphemes by dashes ("-"), although other characters can be used to delimit morphemes, such as "+" and "=" (the latter two are most often used for delimiting clitics). The second line contains a morpheme-by-morpheme or word-by-word gloss for the data in the first line, or a mixture of the two. The second-line delimiters are generally the same as those in the first, and morphemes and words usually align between the two lines. Where a given word or morpheme can be glossed by more than one term, periods (".") or colons (":"), and sometimes dashes or spaces, separate the additional glosses (for instance, in (1) *kataab* is glossed as *book*, *3s* ('third person' and 'singular number'), *f* ('feminine gender'), and *NOM* ('nominative case')). Finally, the third line contains a free-translation of the first line.[7]

(1) fawad- ne sumbal- n kataab dittii
Fawad- ERG Sumbal- DAT book.3s.f.NOM give-PERF.f
'Fawad gave the book back to Sumbal.'[1]

Glosses in the second line take two forms: those representing grammatical information (usually formal or semantic features), which we label with the term **gram** (in the spirit of [4], and described in more detail in [13]), and those that contain unconstrained translations of words and morphemes, which we will label by the generic term **gloss**. Grams are often put in upper case to differentiate them from glosses.

2.2 Why IGT?

Although other forms of interlinear text exist, ODIN is built primarily around IGT, that is, interlinear text that is commonly found in scholarly linguistic discourse. IGT is a semi-structured data type, and as such, is amenable to some degree of automation. There are consistencies in its use across the discipline—the standard 3- or 4-line format, phonetically or orthographically encoded language data, a morpheme-by-morpheme or word-by-word gloss, a free translation, some markup terminology regularities—which provide a rich underlying structure that make the data format more readily accessible to automated manipulation than other less structured data types. Of even greater importance is its ubiquity: it is used throughout the discipline to present language data, and can be found in tens of thousands of Web-accessible linguistic documents.

On one hand, the IGT data format is a presentation format, and as such, is geared for human consumption, not for automation. Two problems are presented for automation: First, structural associations between elements are not clear, in that the alignment between lines is not explicit, nor is the relation between elements on the same line necessarily evident. Second, the semantics used for describing grammatical concepts, the grams, is not consistent across instances, and can vary by language studied, theoretical tradition, and even by researcher.

On the other hand, its consistent structure and format make it conducive to mining and enrichment. Finding IGT involves locating salient linguistic documents and then searching these documents for text that "looks like" IGT. Once discovered, instances of IGT can be "unpacked", that is, the relations between elements across and within the lines of IGT can be made explicit and stored in such a way to facilitate search. In addition, term ambiguities can be resolved through manual and automated means. Finally, the content of IGT can be enriched through the use of statistical taggers and parsers applied against the near universal English translation found within IGT instances.[8] These methods for discovery and enrichment are covered in detail in the next section.

3 ODIN

3.1 Crawling for and Recognizing IGT

Locating IGT on the Web is done in two steps. First, documents that might contain IGT are located. This is done by crawling the Web using linguistically-related search terms and collecting those documents that most likely contain IGT. Second, IGT within these documents is recognized and extracted. IGT recognition and extraction involves scanning the text of a document for patterns that resemble IGT, applying relevant heuristics to improve precision (*e.g.*, language identification, accurate alignment, etc.), and then extracting and storing the recognized instances into the ODIN database.

[7] See the Leipzig Glossing Rules (http://www.eva.mpg.de/lingua/files/morpheme.html) for specific examples of "best-practice" with respect to IGT. The EMELD site (http://www.emeld.org) also contains specific instructions on encoding IGT and other linguistic data types, and discusses best-practice with respect to archiving, interoperation, etc.

[8] Surprisingly, an English translation is often provided with examples even when the source document is in another language.

3.1.1 Crawling for IGT

The major difficulty with locating documents that contain IGT is reducing the size of the search space. We decided very early in the development of ODIN that unconstrained Web crawling was too time and resource intensive a process to be feasible due to the Web's massive size. Focused crawls, *ala* Mercator [17], which optimize crawling performance by quickly eliminating irrelevant pages, presented a good method for reducing the search space, but still required large, resource intensive crawls. We discovered that highly focused **meta-crawls** were far more fruitful. Meta-crawling essentially involves throwing queries against an existing search engine, such as Google or Yahoo, and crawling only the pages returned by those queries.

We found the most successful queries to be those that use strings contained within IGT itself. Since the markup vocabulary for IGT often contains grams (*e.g.*, NOM, ACC, ERG, etc.), the most successful strategy involves using the highest frequency grams as search terms. In addition, we found precision increases when we include two or more grams per query. Thus, for example, although ERG alone returns a large number of linguistic documents, ERG combined with ABS (or any other high frequency gram) returns a far less noisy and far more relevant set of documents.

Other queries we use include: language names and codes (drawn from the Ethnologue database [11][9]), linguists' names and the languages they work on (drawn from the LinguistList linguist database), linguistically relevant terms (drawn from the SIL linguistic glossary), and queries that look for particular language data, such as words or morphemes found in IGT.

Given a set of documents returned from a query, we then search through them for signs of IGT. Integrated with the crawlers is an IGT recognizer, whose mandate is a limited one: if a targeted document appears to contain just one instance of IGT, that document is harvested for inclusion in a subsequent more thorough off-line recognition process.

3.1.2 Recognizing IGT

The current method for recognizing IGT uses regular expression "templates" to look for text that resembles IGT. A regex template, or rule, might look something like that shown in (2). The rule in (2) would "discover" an instance of IGT like that shown in (1).

(2)

```
\t*(\()\d*\).*\n
\t*.*\n
\t*\'.*\n
```

[9]We are in the process of converting to ISO 639-3 language codes. ISO 639-3 language codes are a merger of Ethnologue and ISO 639-2 language codes.

The IGT recognizer uses a chart parsing algorithm [7] to track the rules currently being processed. When a regex rule is completed successfully, the relevant instance of IGT is extracted and added to the database as a potential instance of IGT. The chart is then cleared and processing continues.

The IGT recognizer as it is currently implemented suffers from fairly low precision (0.61) and recall (0.52). Since we feel that precision is far more important than recall, the assumption being that linguists would be far less forgiving of incorrectly identified IGT than of missing instances, we have concentrated most of our efforts on improving precision at the cost of recall.

A few heuristics have been developed to improve precision. The first involves a rather "unforgiving" alignment algorithm that filters out IGT instances where the first and second lines fail to align. Precision rises to 0.88 using this method, but recall drops rather precipitously to 0.18.

Additional language identification heuristics are used to further improve precision. The first looks for language names in the surrounding text. Since the database of language names contained in the Ethnologue is used [11], each language name that is found can be mapped to a unique three-letter language code. The second involves building statistical language models [5] over the first line of IGT, the line encoding the language data of interest, which, when compared against an inventory of language models built over already collected IGT, returns one or more ranked language codes. The language codes that are returned are then compared against the language code returned by the first heuristic, and if a match is found, the IGT instance is automatically added to the database. Ultimately, precision for IGT recognition rises to 0.98 using these heuristics, although recall remains very low, at 0.13.

3.1.3 Manual and Automated Review

A large percentage of the IGT collected by ODIN has been manually reviewed. Of the 33,713 instances of IGT in the ODIN database at the time of this writing (06/17/2006), 18,193 have been hand verified to be both IGT and to be in the languages in question (there are 701 languages currently in ODIN). The percentage of hand-vetted examples, 54%, will reduce over time since the automated methods operate at a faster pace than any supervised strategies. Of the 6,540 records added to the database in the two-month period ending the 17th of June 2006, 18% were hand verified, and the remaining 82% were added without verification. Hand verification will continue to be part of the process since current automated methods, especially those that use statistical methods, require existing data to build statistical models over. Once data for a new language have been added to the database, however, models can easily be trained on these data, meaning that future IGT instances discovered for that language can be added without supervision.

One of six verification levels is associated with each data

point collected by ODIN. First, those data that have been collected that are IGT but have had no language code associated with them are stored at the level of "Very Low", and are not accessible to search; these data are stored such that they can be manually reviewed at some point in the future. Second, those data that are collected but have not been verified either manually or automatically, other than through a simple language name check (where the language name is retrieved from the surrounding text), are stored at the verification level of "Low". Third, those that are verified through automated means, mainly using the Cavnar & Trenkle algorithm verified against a recovered language name, are recorded as "Auto". Those manually verified are stored either as "High" (fourth) or "Highest" (fifth), where the former marks data that contains some corruption, which is likely introduced through the extraction process, and where the latter is completely clean and matches the source perfectly; both High and Highest are verified against the source document to be in the language specified. Finally, an additional verification level is provided for data that are manually supplied by a data provider. Since these data are likely to be very clean and manually verified by an expert on the language, it is encoded at the "Very Highest" level. A list of the verification levels and the amount of data discovered for each is contained in Table 1. (Note: Each verification level corresponds to a number on a 0–100 scale, with Very Low, Low, Auto, High, Highest, Very Highest distributed across this scale. A numerical scale accommodates additional verification levels should needs change in the future.)

Level	# Instances
Very Low	3038
Low	1985
Auto	10513
High	5630
Highest	12511
Very Highest	334

Table 1. Amount of Data Collected at Each Verification Level as of 2006/08/11

3.2 Extracting and Curating IGT

Extracting IGT is fairly straightforward once an instance of IGT has been recognized. Once extracted, each instance is stored in the ODIN database in such a way as to facilitate search. Currently, IGT is stored in the database in its "raw" presentation format, that is, as lines of text as discovered in the source document. Further, the lines are parsed and aligned, and the aligned elements between the language and gloss line are also stored as separate entries (see Section 3.3.1), as are the part-of-speech tagged translation lines. The redundancy means that the database itself is not fully normalized, but the inefficiencies were dictated by speed concerns required by search and display. All instances that are stored in the database are indexed and searchable by language code, facilitating relatively rapid language code and name search.

3.3 Enriching the Data

The primary motivation behind ODIN is to locate instances of IGT and expose these instances to search. The initial search strategy behind ODIN was to comply with the Open Languages Archives Community (OLAC) model[19], that is, allow search by language name or code, either through OLAC's search interfaces[10] or through a locally provided facility. IGT, however, has remarkably rich content that opens possibilities for other types of search. To facilitate these other types of search, we have enriched IGT content through various means. First of all, aligning morphemes and words and their glosses (located on the the canonical first and second lines, respectively) allows systematic queries across these data. Using very limited part-of-speech (POS) tagging of the second line, it is possible to make guesses as to the status of various kinds of morphemes, such as whether morphemes are roots, affixes or clitics. Further, the annotation that is consistently used in the gloss line provides another avenue for search, and if normalized to a common vocabulary, allows search across disparate vocabularies. Finally, because the translation line is almost universally English, robust POS taggers and parsers for English can be brought to bear on IGT. Queries across the enriched English translations can give us clues as to the structures that might exist in the source language data.

3.3.1 Aligning IGT Instances

The first step in enriching the IGT involves aligning elements between the language and gloss lines. The explicit alignment between these lines associates each morpheme or word with its respective gloss. Although the methods used for alignment are somewhat adaptive depending on the delimiters used—linguists are not completely consistent in the use of word and morpheme delimiters in IGT—a number of instances fail to align correctly, due mostly to missing delimiters or inconsistent uses across the lines of data; this means that a number of instances can only be partially aligned. Of the data currently stored in ODIN, 74% of the IGT instances have been fully aligned.

Although the alignment algorithms have proven most useful for explicitly representing the relationship between elements for purposes of search, they have also proven useful for converting IGT to an XML format. XML is useful since it explicitly encodes relationships between elements,

[10] OLAC can be search via an interface at LinguistList (http://www.linguistlist.org) or the Linguistics Data Consortium (http://www.language-archives.org/tools/search)

is an archivable and interoperable data format, and provides the means for subsequently rendering data in a variety of output formats (*e.g.*using XSLT). We have developed methods for rendering IGT stored in ODIN in the Hughes, Bird and Bow (HBB) [12] which can be applied to any data currently housed in ODIN that fully aligns.

3.3.2 Term Disambiguation

As noted earlier, glosses in IGT take two forms: those representing grammatical information, the "grams", such as *ERG*, *NOM*, *PERF* in (1), or glosses, such as *book* or *give*. Significant manual effort has gone into disambiguating the grams that are used in IGT, done as part of the initial funding used to develop ODIN. Eighty-four terms have been determined to have near universal interpretations, with several hundred more commonly used for specific languages or language families. A gram lexicon has been built in ODIN, which contains a mapping of these terms to a common semantic, namely, relevant concepts contained in GOLD, the General Ontology of Linguistic Description[11]. These term mappings can be rendered as XML-encoded *terminology sets*, or *termsets*, which are themselves returned as results for certain kinds of queries cast against ODIN.

It is naive to assume that all instances of a term map to the same concept. As noted earlier, terms' meanings can vary by language or language family, researcher or theoretical tradition. For instance, the term NOM is used nearly universally to mean NominativeCase, with over 99.5% instances of NOM having this mapping. However, some research traditions dictate the use of NOM for Nominalizer rather than NominativeCase. To help in discovering these outliers, we have applied statistical clustering techniques to look at distribution of terms across the database, where outliers are not mapped to a GOLD concept without manual intervention. These methods prevent some false hits in search.

3.3.3 Tagging and Parsing the English Translation

IGT is generally used within a larger rhetorical context, and it is often annotated to be compatible with that context. For instance, a paper on negation may contain instances of IGT where the gloss-line annotation shares this focus, *e.g.*, where NEG and other relevant annotation are used or highlighted. Other annotation lacking relevance to the context are backgrounded or ignored altogether. However, it is possible to discover other phenomena in an IGT instance through additional, automated enrichment: Because most IGT instances that have been discovered by ODIN have English translation lines (this is true for over 98% of the instances currently housed in ODIN), we have found it fruitful to apply robust statistical techniques for tagging and parsing against the English translation, and use these additional structures to support other kinds of query. For instance, it is possible to discover a variety of linguistically salient constructions, such as passives, conditionals, counterfactuals, raising constructions, sententially negated constructions, etc. (see Section 3.4.4), in a tagged and parsed English gloss that would not otherwise be discoverable. The English translations for all examples in ODIN have been tagged using the Ratnaparkhi MaxEnt tagger [18]. A subset of the examples have also been parsed using the Collins Parser [6]. The error rate for the Collins parser has been high enough—approximately 26% of the instances are misparsed—that we have not parsed all instances of IGT, and are currently exploring methods for improving parsing methods, including implementing multiple parsers over the data.

3.4 Search

The focus of ODIN is and has always been search: how can linguists find the data that they are interested in and how can the data be encoded in such a way as to accommodate the variety of queries that a linguist might ask. As mentioned earlier, we initially limited search to queries by language name and code, which maintained compatibility with OLAC. More recently we have extended the search facility by adding what we have labeled *Advanced Search*, which provides three additional search query types: search by language family, search by concept/gram, and search by linguistic constructions. Each of the search facilities provided by ODIN are discussed in detail in the following sections.

3.4.1 Language Name/Code Search

The initial search facility provided by ODIN was designed to maintain compatibility with OLAC. OLAC provides for interoperation over language repositories through well-defined metadata based on the Dublin Core metadata set [3]. Registered repositories are harvested regularly by OLAC, and made available to search through the OLAC portals at LinguistList and the LDC. Using these portals, users can search for language repositories using language names and codes, where results are presented as a list of relevant repositories and URLs. ODIN regularly generates OLAC metadata as new data are discovered. Results to OLAC queries are supplied via a PHP script that lists all relevant documents for supplied language codes. A sample output screen is shown in Figure 1.

Note that the results are ordered by Verification level, with the documents containing instances at the Highest verification level listed first (the verification levels are detailed in Section 3.1.3). Users have the option to view the documents from which IGT were harvested, or can view the instances themselves (which, when displayed, are fully cited),

[11] GOLD was conceived of early in the EMELD efforts. See [8] and [9] for more detailed background.

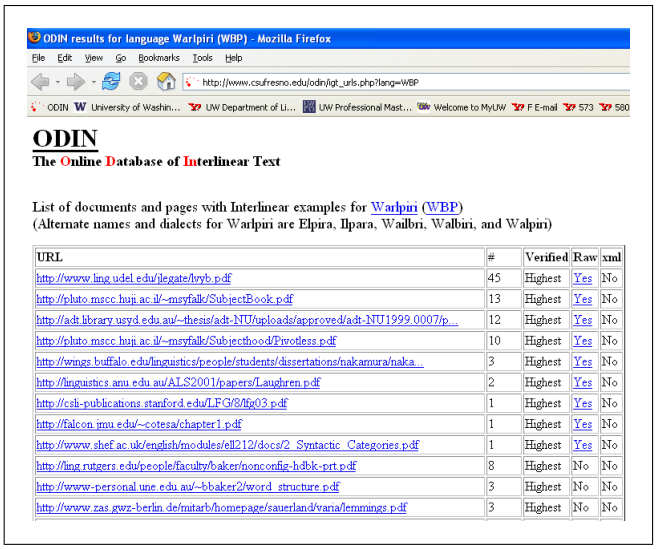

Figure 1. Query Results for Warlpiri

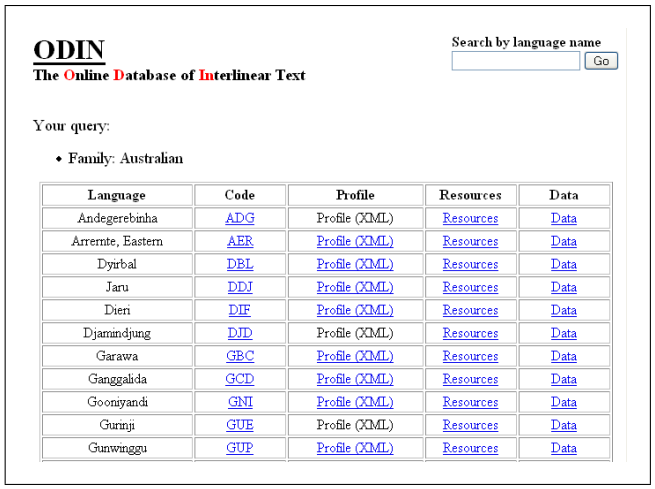

Figure 2. Query Results for Australian Languages

where the latter can be viewed in either the raw, as-extracted format or in XML (where available).

3.4.2 Language Family Search

An extension to language search is search by language family. We use the language families as defined in Ethnologue, and provide the facility for users to select a family name from a pull-down list. Results are returned as a list of languages within that family, but only for those languages for which data exist in ODIN. An example output is shown in Figure 2. The user has the option of displaying *Resources* for each language, which are effectively the documents the IGT was extracted from, or he or she can view the instances of IGT themselves (again, with individual citation records provided by document). The user can also display an XML encoded **Language Profile**, which contain information that has been automatically compiled from harvested IGT. Profiles contain information about grammatical categories discovered in IGT, such as cases, aspects, tenses, etc.[12]

The Language Family search can also be combined with the Concept/Gram and Construction Searches discussed in the next two sections. The combination allows users to constrain the results to specific language families, thus reducing the number of documents and IGT instances returned for any particular query.

3.4.3 Concept/Gram Search

The Concept/Gram search allows users to search through IGT for instances that contain morphemes or words that encode particular grammatical concepts, such NominativeCase, PerfectiveAspect, SubjunctiveModality, etc. The user specifies the desired concepts in terms of GOLD, and the instances of data that map to these concepts are searched for. Grams such as PST and PAST are normalized to a common semantic, namely PastTense, so the query for PastTense will return both of these forms, as well as 1SGPast, HodiernalPast, and others. The user can also request morphemes encoded as affixes (prefixes, suffixes) or clitics (proclitics, enclitics), which, although not normally explicitly encoded in IGT, can be deduced from their relationships to root morphemes. Thus a query might look something like "PastTense encoded as a Suffix, Singular encoded as an Enclitic", which will return all instances of IGT where both conditions hold.

The Concept search in ODIN is not as sophisticated as that discussed in [20], where the full power of GOLD was brought to bear on an RDF-encoded database. Because of speed concerns, ODIN has limited the search facility to a two-tiered, GOLD compatible hierarchy: users can search for a specific concept, such as ErgativeCase, or a parent concept, such as Case. In [20], the full GOLD hierarchy could be queried, allowing users to abstract away from specific categories and relationships, using the full inference capability of RDF. Although users may wish such power in a linguistic query, implementing such queries are not tractable in real-time given current technology.

3.4.4 Construction Search

The Construction Search is the most powerful and most innovative of the query facilities currently provided by ODIN. Rather than limiting search to just the content and markup natively contained within IGT, Construction Search

[12] It is not our intention to discuss language profiles in detail here. The reader is referred to [10] for more thorough coverage of terminology sets and language profiles.

searches over "enriched" content, where the additional content and structure is added to the English translations through the use of statistical taggers and parsers. Given enriched English content, a search for Relative Clauses, for instance, looks for the tell-tale structural clues of its presence: a word tagged as NN or NNP followed by an appropriate relativizer. Likewise, a search for Passives looks for forms of the verb *to be* followed by the past participle of the verb, tagged by VBN. Currently, 15 construction queries have been implemented, with some 40 additional queries being evaluated and built.

It is crucial to recognize that construction queries rely on the English translation, *not* on the source language data. Thus, they can only be seen as guesses: A passive discovered in the English translation does not mean that a passive will necessarily exist in the source language data. However, linguists are often very careful in how they craft translations in IGT such that the translations closely mirror the intended meaning and the syntactic structure of the source. Thus, discovering a passive in the English translation could realistically be associated with a passive-like structure in the source language data: passives, topicalized structures, scrambled structures, or other related phenomena, often characterized by "movement" away from some canonical order.

3.5 Fair Use

Crucial to the ODIN effort has been the fair use of the data that are collected. We fully recognize that any instance of data that is collected by ODIN is proprietary and the property either of the linguist who crafted the example or the native community where the example might have originated. In line with linguistic custom and Section 107 of Copyright Law[13], all instances of IGT that are returned as results of a query are fully cited as to their source, and the source document is provided via link (no copies of source documents are maintained on the ODIN site). Although search across instances for which no citation information is provided by ODIN, only links to source documents will be provided; no data will be displayed.

4 Future Directions

In this section, we discuss future directions for ODIN. Two projects that are currently underway are discussed, the first involving improvements to recall of the IGT recognition process, and the second involving projections of structure from the English translation to the source language data.

[13] See [16] and [14] for a thorough review of copyright law as it pertains to linguistics data.

4.0.1 Recognition

The IGT recognizer, as currently designed, is fairly brittle in that it does not allow much flexibility in the definitions it uses, leading to lowered recall. Even slight variations in the format of IGT can lead to recognition failure, requiring the subsequent manual addition of new rules. Although our regex templates have fairly broad coverage, capturing the structure of most IGT instances, many instances are still skipped, some because they contain only minor variations of existing structures. With the next generation IGT recognizer currently being designed, the rules, along with language identification and other IGT indicators, will be treated as *features* used by machine learning algorithms. A machine learning classifier, such as one based on the Maximum Entropy [2] or Support Vector [21] algorithms, can be trained on these features, and, during the training process, will converge on the features that are most salient to discovering IGT in text. These methods, we feel, will improve recall overall, without significant reductions in precision.

4.0.2 Projecting Structure

As discussed in Section 3.4.4, it is important to recognize that search over the English translation can only discover constructions contained in the English translation; it does not mean that the same or similar constructions will be found in the source language data with which the translation is associated. We assume that a relationship exists, since the linguist will likely bias the translation to be as close to the original language data as possible. There is no guarantee, however, that every construction found in the English gloss will also be found in the source.

Drawing inspiration from Yarowsky and Ngai 2001 [22], efforts are currently underway [15] to project tags and parsed structures from the English translations onto the associated source language data. Because each instance of IGT contains an intervening gloss line, the English translation can be aligned with the gloss line, which in turn can easily be aligned with the language data. Where possible, the tags and parses applied to the English translation can then be projected onto the source language data. These projected structures can then be searched, where queries can be cast against the source itself rather than against the English translation. Current experiments show successful projections on the order of 60–90%, depending on the quality of markup and the typological characteristics of the source language (and how much it diverges typologically from from English).

5 Conclusion

The ODIN database is presented here as a model for leveraging existing infrastructure to facilitate sophisticated search, and as a model for interoperation over legacy data

and legacy formats. It stands as an example of how we might proceed in developing the tools and methodologies to grow a richer e-Linguistics, especially by transforming and enriching existing Web-accessible resources. ODIN itself is a resource that is searchable at the level of data, going beyond other systems both in its breadth, the number of languages served, and depth, the granularity of search that is provided. Future work looks to apply sophisticated statistical NLP technology to harvested language data, which will result in even more searchable content. The resulting infrastructure can then be further exploited for the purposes beyond search, such as developing language-specific tools and resources (*e.g.*statistical taggers and parsers), most especially for languages for which very few digital resources exist. The creation of sophisticated tools can then facilitate the uptake of additional data and resources, expanding the scope of the ever evolving e-Linguistics infrastructure.

6 Acknowledgements

A special thanks goes to Terry Langendoen and Scott Farrar for their support of ODIN from the beginning. I gratefully acknowledge the support of the EMELD PIs and associates, especially Gary Simons, Helen Aristar-Dry, and Anthony Aristar. Finally, I acknowledge the NSF-funded Data-Driven Linguistic Ontology Development project (BCS-0411348) which supported the author through the formative stages of ODIN's development.

References

[1] R. N. Akhtar. Affix -s(uu) constructions in Punjabi. *Essex Graduate Student Papers in Language and Linguistics*, 1, 1997.

[2] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), March 1996.

[3] S. Bird and G. Simons. Extending Dublin Core metadata to support the description and discovery of language resources. *Computers and the Humanities*, 17(4):375–388, 2003.

[4] J. L. Bybee and O. Dahl. The creation of tense and aspect systems in the languages of the world. *Studies in Language*, 13(1):51–103, 1989.

[5] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, April 1994.

[6] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, 1999.

[7] J. Earley. An Efficient Context-Free Parsing Algorithm. *Communications of the ACM*, 13(2):94, 1970.

[8] S. Farrar. New ways of thinking about lexical resources: a proposal for the Semantic Web. Presented as the Workshop on Lexical Resources, February 2003.

[9] S. Farrar and D. T. Langendoen. A linguistic ontology for the Semantic Web. *GLOT International*, 7(3):97–100, 2003.

[10] S. Farrar and W. D. Lewis. The GOLD Community of Practice: An infrastructure for linguistic data on the Web. In *Proceedings of the EMELD 2005 Workshop on Digital Language Documentation: Linguistic Ontologies and Data Categories for Language Resources*, Cambridge, Massachusetts, 2005.

[11] R. G. Gordon, editor. *Ethnologue: Languages of the World*. Fifteenth edition, 2005.

[12] B. Hughes, S. Bird, and C. Bow. Interlinear text facilities. In *E-MELD 2003*, Michigan State University, 2003. [http://emeld.org/workshop/2003/baden-demo.html]. See also [http://www.cs.mu.oz.au/research/lt/emeld/interlinear].

[13] W. D. Lewis. Mining and migrating interlinear glossed text. Technical report, Workshop on Digitizing and Annotating Texts and Field Recordings, LSA Institute, July 2003. Available at http://emeld.org/workshop/2003/papers03.html.

[14] W. D. Lewis, S. Farrar, and D. T. Langendoen. Linguistics in the internet age: Tools and fair use. In *Proceedings of EMELD 2006 Workshop on Digital Language Documentation: Tools and Standards: The State of the Art*, Lansing, Michigan, 2006.

[15] W. D. Lewis, F. Xia, and D. Jinguji. Projecting structure onto lesser studied languages. Available at http://faculty.washington.edu/wlewis2/papers/papers.html, in prep.

[16] M. Liberman. Legal, ethical, and policy issues concerning the recording and publication of primary language materials. In S. Bird and G. Simons, editors, *Proceedings of the workshop on web-based language documentation and description*, 2000. http://www.ldc.upenn.edu/exploration/expl2000/ (2006-May-17).

[17] M. Najork and A. Heydon. High-performance web crawling. In J. M. Abello, P. M. Pardalos, and M. G. C. Resende, editors, *Handbook of massive data sets*. Kluwer Academic Press, Dordrecht/Boston/London, 2002.

[18] A. Ratnaparkhi. *Maximum Entry Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998.

[19] G. Simons and S. Bird. The open language archives community: An infrastructure for distributed archiving of language resources. *Literary and Linguistic Computing*, 18:117–128, 2003.

[20] G. F. Simons, W. D. Lewis, S. O. Farrar, D. T. Langendoen, B. Fitzsimons, and H. Gonzalez. The semantics of markup: Mapping legacy markup schemas to a common semantics. In *Proceedings of the 4th workshop on NLP and XML (NLPXML-2004)*, pages 25 – 32, Barcelona, Spain, 2004b. held in cooperation with ACL-04.

[21] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[22] D. Yarowksy and G. Ngai. Inducing Multilingual POS taggers and NP Bracketers via robust projection across aligned corpora. In *Proceedings of NAACL-2001*, pages 377–404, 2001.