Towards Creating Precision Grammars from Interlinear Glossed Text: Inferring Large-Scale Typological Properties

Emily M. Bender Michael Wayne Goodman Joshua Crowgey Fei Xia
Department of Linguistics
University of Washington
Seattle WA 98195-4340
{ebender,goodmami,jcrowgey,fxia}@uw.edu

Abstract

We propose to bring together two kinds of linguistic resources—interlinear glossed text (IGT) and a language-independent precision grammar resource—to automatically create precision grammars in the context of language documentation. This paper takes the first steps in that direction by extracting major-constituent word order and case system properties from IGT for a diverse sample of languages.

1 Introduction

Hale et al. (1992) predicted that more than 90% of the world's approximately 7,000 languages will become extinct by the year 2100. This is a crisis not only for the field of linguistics—on track to lose the majority of its primary data—but also a crisis for the social sciences more broadly as languages are a key piece of cultural heritage. The field of linguistics has responded with increased efforts to document endangered languages. Language documentation not only captures key linguistic data (both primary data and analytical facts) but also supports language revitalization efforts. It must include both primary data collection (as in Abney and Bird's (2010) universal corpus) and analytical work elucidating the linguistic structures of each language. As such, the outputs of documentary linguistics are dictionaries, descriptive (prose) grammars as well as transcribed and translated texts (Woodbury, 2003).

Traditionally, these outputs were printed artifacts, but the field of documentary linguistics has increasingly realized the benefits of producing digital artifacts as well (Nordhoff and Poggeman, 2012). Bender et al. (2012a) argue that the documentary value of electronic descriptive grammars can be significantly enhanced by pairing them with implemented (machine-readable) precision grammars and grammar-derived treebanks. However, the creation of such precision grammars is time consuming, and the cost of developing them must be brought down if they are to be effectively integrated into language documentation projects.

In this work, we are interested in leveraging existing linguistic resources of two distinct types in order to facilitate the development of precision grammars for language documentation. The first type of linguistic resource is collections of interlinear glossed text (IGT), a typical format for displaying linguistic examples. A sample of IGT from Shona is shown in (1).

(1) Ndakanga ndakatenga muchero
ndi-aka-nga ndi-aka-teng-a mu-chero
SBJ.1SG-RP-AUX SBJ.1SG-RP-buy-FV CL3-fruit
'I had bought fruit.' [sna] (Toews, 2009:34)

The annotations in IGT result from deep linguistic analysis and represent much effort on the part of field linguists. These rich annotations include the segmentation of the source line into morphemes, the glossing of those individual morphemes, and the translation into a language of broader communication. The IGT format was developed to compactly display this information to other linguists. Here, we propose to repurpose such data in the automatic development of further resources.

The second resource we will be working with is the LinGO Grammar Matrix (Bender et al., 2002; 2010), an open source repository of implemented linguistic analyses. The Grammar Matrix pairs a core grammar, shared across all grammars it creates, with a series of libraries of analyses of cross-linguistically variable phenomena. Users access the system through a web-based questionnaire which elicits linguistic descriptions of languages and then outputs working HPSG (Pollard and Sag, 1994) grammar fragments compatible with DELPH-IN (www.delph-in.net) tools based on those descriptions. For present purposes, this system can be viewed as a function which maps simple descriptions of languages to preci-

Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 74–83,
Sofia, Bulgaria, August 8 2013.

sion grammar fragments. These fragments are relatively modest, yet they relate linguistic strings to semantic representations (and vice versa) and are ready to be built out to broad coverage.

Thus we ask whether the information encoded by documentary linguists in IGT can be leveraged to answer the Grammar Matrix's questionnaire and create a precision grammar fragment automatically. The information required by the Grammar Matrix questionnaire concerns five different aspects of linguistic systems: (i) constituent ordering (including the presence/absence of constituent types), (ii) morphosyntactic systems, (iii) morphosyntactic features, (iv) lexical types and their instances and (v) morphological rules. In this initial work, we target examples of types (i) and (ii): the major constituent word order and the general type of case system in a language. The Grammar Matrix and other related work are described in further in §2. In §3 we present our test data and experimental set-up. §§4–5 describe our methodology and results for the two tasks, respectively, with further discussion and outlook in §§6–7.

2 Background and Related Work

2.1 The Grammar Matrix

The Grammar Matrix produces precision grammars on the basis of description of languages that include both high-level typological information and more specific detail. Among the former are aspects (i)–(iii) listed in §1. The third of these (morphosyntactic features) concerns the type and range of grammaticized information that a language marks in its morphology and/or syntax. This includes person/number systems (e.g., is there an inclusive/exclusive distinction in non-singular first person forms?), the range of aspectual distinctions a language marks, and the range of cases (if any) in a language, inter alia. The answers to these questions in turn cause the system to provide relevant features that the user can reference in providing the more specific information elicited by the questionnaire ((iv) and (v) above), viz., the definition of both lexical types (e.g., first person dual exclusive pronouns) and morphological rules (e.g., nominative case marking on nouns).

The information input by the user to the Grammar Matrix questionnaire is stored in a file called a 'choices file'. The choices file is used both in the dynamic definition of the html pages (so that the features available for lexical definitions depend on earlier choices) and as the input to the customization script that actually produces the grammar fragments to spec. The customization system distinguishes between choices files which are complete and consistent (and can be used to create working grammar fragments) and those which do not yet have answers to required questions or give answers which are inconsistent according to the underlying grammatical theory. The ultimate goal of the present project is to be able to automatically create complete and consistent choices files on the basis of IGT, and in fact to create complete and consistent choices files which take maximal advantage of the analyses stored in the Grammar Matrix customization system, answering not only the minimal set of questions required but in fact all which are relevant and possible to answer based on the information in the IGT.

Creating such complete and consistent choices files is a long-term project, with different approches required for the different types of questions outlined in §1. Bender et al. (2012b) take some initial steps towards answering the questions which define lexical rules. We envision answering the questions regarding morphosyntactic features through an analysis of the grams that appear on the gloss line, with reference to the GOLD ontology (Farrar and Langendoen, 2003). The implementation of such systems in such a way that they are robust to potentially noisy data will undoubtedly be non-trivial. The contribution of this paper is the development of systems to handle one example each of the questions of types (i) and (ii), namely detecting major constituent word order and the underlying case system. For the first, we build directly on the work of Lewis and Xia (2008) (see §2.2). Our experiment can be viewed as an attempt to reproduce their results in the context of the specific view of word order possibilities developed in the Grammar Matrix. The second question (that of case systems) is in some ways more subtle, requiring not only analysis of IGT instances in isolation and aggregation of the results, but also identification of particular kinds of IGT instances and comparison across them.

2.2 RiPLes

The RiPLes project has two intertwined goals. The first goal is to create a framework that allows the rapid development of resources for resource-poor languages (RPLs), which is accomplished by

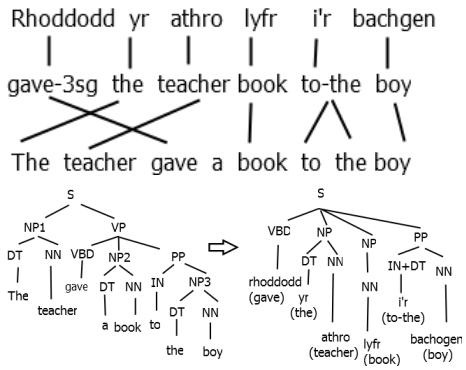

Figure 1: Welsh IGT with alignment and projected syntactic structure

bootstrapping NLP tools with initial seeds created by projecting syntactic information from resource-rich languages to RPLs through IGT. Projecting syntactic structures has two steps. First, the words in the language line and the translation line are aligned via the gloss line. Second, the translation line is parsed by a parser for the resource-rich language and the parse tree is then projected to the language line using word alignment and some heuristics as illustrated in Figure 1 (adapted from Xia and Lewis (2009)).[1] Previous work has applied these projected trees to enhance the performance of statistical parsers (Georgi et al., 2012). Though the projected trees are noisy, they contain enough information for those tasks.

The second goal of RiPLes is to use the automatically created resources to perform cross-lingual study on a large number of languages to discover linguistic knowledge. For instance, Lewis and Xia (2008) showed that IGT data enriched with the projected syntactic structure could be used to determine the word order property of a language with a high accuracy (see §4). Naseem et al. (2012) use this type of information (in their case, drawn from the WALS database (Haspelmath et al., 2008)) to improve multilingual dependency parsing. Here, we build on this aspect of RiPLes and begin to extend it towards the wider range of linguistic phenomena and more detailed classification within phenomena required by the Grammar Matrix questionnaire.

2.3 Other Related Work

Our work is also situated with respect to attempts to automatically characterize typological properties of languages, including Daumé III and Campbell's (2007) Bayesian approach to discovering typological implications and Georgi et al.'s (2010) work on predicting (unknown) typological properties by clustering languages based on known properties. Both projects use the typological database WALS (Haspelmath et al., 2008), which has information about 192 different typological properties and about 2,678 different languages (though the matrix is very sparse). This approach is complementary to ours, and it remains an interesting question whether our results could be improved by bringing in information about other typological properties of the language (either extracted from the IGT or looked up in a typological database).

Another strand of related work concerns the collection and curation of IGT, including the ODIN project (Lewis, 2006; Xia and Lewis, 2008), which harvests IGT from linguistics publications available over the web and TypeCraft (Beermann and Mihaylov, 2009), which facilitates the collaborative development of IGT annotations. TerraLing/SSWL[2] (Syntactic Structures of the World's Languages) has begun a database which combines both typological properties and IGT illustrating those properties, contributed by linguists.

Finally, Beerman and Hellan (2011) represents another approach to inducing grammars from IGT, by bringing the hand-built linguistic knowledge sources closer together: On the one hand, their cross-linguistic grammar resource (TypeGram) includes a mechanism for mapping from strings specifying verb valence and valence-altering lexical rules to sets of grammar constraints. On the other hand, their IGT authoring environment (TypeCraft) provides support for annotating examples with those strings. The approach advocated here attempts to bridge the gap between IGT and grammar specification algorithmically, instead.

3 Development and Test Data

Our long-term goal is to produce working grammar fragments from IGT produced in documentary linguistics projects. However, in order to evaluate the performance of approaches to answering the high-level questions in the Grammar Matrix questionnaire, we need both IGT and gold-standard answers for a reasonably-sized sample of languages. We have constructed development and test data for this purpose on the basis of work done

[1] The details of the algorithm and experimental results were reported in (Xia and Lewis, 2007).

[2] http://sswl.railsplayground.net/, accessed 4/25/13

Sets of languages	DEV1 (n=10)	DEV2 (n=10)	TEST (n=11)
Range of testsuite sizes	16–359	11–229	48–216
Median testsuite size	91	87	76
Language families	Indo-European (4), Niger-Congo (2), Afro-Asiatic, Japanese, Nadahup, Sino-Tibetan	Indo-European (3), Dravidian (2), Algic, Creole, Niger-Congo, Quechuan, Salishan	Indo-European (2), Afro-Asiatic, Austro-Asiatic, Austronesian, Arauan, Carib, Karvelian, N. Caucasian, Tai-Kadai, Isolate

Table 1: Language families and testsuites sizes (in number of grammatical examples)

by students in a class that uses the Grammar Matrix (Bender, 2007). In this class, students work with descriptive resources for languages they are typically not familiar with to create testsuites (curated collections of grammatical and ungrammatical examples) and Grammar Matrix choices files. Later on in the class, the students extend the grammar fragments output by the customization system to handle a broader fragment of the language. Accordingly, the testsuites cover phenomena which go beyond the customization system.

Testsuites for grammars, especially in their early stages of development, require examples that are simple (isolating the phenomena illustrated by the examples to the extent possible), built out of a small vocabulary, and include both grammatical and ungrammatical examples (Lehmann et al., 1996). The examples included in descriptive resources often don't fit these requirements exactly. As a result, the data we are working with include examples invented by the students on the basis of the descriptive statements in their resources.[3]

In total, we have testsuites and associated choices files for 31 languages, spanning 17 language families (plus one creole and one language isolate). The most well-represented family is Indo-European, with nine languages. We used 20 languages, in two dev sets, for algorithm development (including manual error analysis), and saved 11 languages as a held-out test set to verify the generalizability of our approach. Table 1 lists the language families and the range of testsuite sizes for each of these sets of languages.

4 Inferring Word Order

Lewis and Xia (2008) show how IGT from ODIN (Lewis, 2006) can be used to determine, with high accuracy, the word order properties of a language. They identify 14 typological parameters related to word order for which WALS (Haspelmath et al., 2008) or other typological resources provide information. The parameter most closely relevant to the present work is Order of Words in a Sentence (Dryer, 2011). For this parameter, Lewis and Xia tested their method in 97 languages and found that their system had 99% accuracy provided the IGT collections had at least 40 instances per language.

The Grammar Matrix's word order questions differ somewhat from the typological classification that Lewis and Xia (2008) were using. Answering the Grammar Matrix questionnaire amounts to more than making a descriptive statement about a language. The Grammar Matrix customization system translates collections of such descriptive statements into working grammar fragments. In the case of word order, this most directly effects the number and nature of phrase structure rules included in the output grammar, but can also interact with other aspects of the grammar (e.g., the treatment of argument optionality). More broadly, specifying the word order system of a grammar determines both grammaticality (accepting some strings, ruling out others) and, for the fixed word orders at least, aspects of the mapping of syntactic to semantic arguments.

Lewis and Xia (2008), like Dryer (2011), gave the six fixed orders of S, O and V plus "no dominant order". In contrast, the Grammar Matrix distinguishes Free (pragmatically constrained), V-final, V-initial, and V2 orders, in addition to the six fixed orders. It is important to note that the relationship between the word order type of a language and the actual orders attested in sentences can be somewhat indirect. For a fixed word order language, we would expect the order declared as its type to be the most common in running text, but not the only type available. English, for example, is an SVO language, but several constructions allow for other orders, including subject-auxiliary inversion, so-called topicalization, and others:

(2) Did Kim leave?

(3) The book, Kim forgot.

In a language with more word order flexibility in general, there may still be a preferred word order

[3] Such examples are flagged in the testsuites' meta-data.

which is the most common due to pragmatic or other constraints. Users of the Grammar Matrix are advised to choose one of the fixed word orders if the deviations from that order can generally be accounted for by specific syntactic constructions, and a freer word order otherwise.

The relationship between the correct word order choice for the Grammar Matrix customization system and the distribution of actual token word orders in our development and test data is affected by another factor, related to Lewis and Xia's 'IGT bias' which we dub 'testsuite bias'. The collections of IGT we are using were constructed as testsuites for grammar engineering projects and thus comprise examples selected or constructed to illustrate specific grammatical properties in a testing regime where one example is enough to represent each sentence type of interest. Therefore, they do not represent a natural distribution of word order types. For example, the testsuite authors may show the full range of possible word orders in the word order section of the testsuite and then default to one particular choice for other portions (those illustrating e.g., case systems or negation).

4.1 Methodology

Our first stpes mirror the RiPLes approach, parsing parse the English translation of each sentence and projecting the parsed structure onto the source language line. Functional tags, such as SBJ and OBJ, are added to the NP nodes on the English side based on our knowledge of English word order and then carried over to the source language side during the projection of parse trees. The trees are then searched for any of ten patterns: SOV, SVO, OSV, OVS, VSO, VOS, SV, VS, OV, and VO. The six ternary patterns match when both verbal arguments are present in the same clause. The four binary patterns are for intransitive sentences or those with dropped arguments. These ten patterns make up the *observed word orders*.

Given our relatively limited data set (each language is one data point), we present an initial approach to determining *underlying word order* based on heuristics informed by general linguistic knowledge. We compare the distribution of observed word orders to distributions we expect to see for canonical examples of underlying word orders. We accomplish this by first deconstructing the ternary observed-word-orders into binary patterns (the four above plus SO and OS). This gives us three axes: one for the tendency to exhibit VS or SV order, another for VO or OV order, and another for OS or SO order. By counting the observed word orders in the IGT examples, we can place the language in this three-dimensional space. Figure 4.1 depicts this space with the positions of canonical word orders.[4] The canonical word order positions are those found under homogeneous observations. For example, the canonical position for SOV order is when 100% of the sentences exhibit SO, OV, and SV orders; and the canonical position for Free word order is when each observed order occurs with equal frequency to its opposite order (on the same axis; e.g. VO and OV). We select the underlying word order by finding which canonical word order position has the shortest Euclidean distance to the observed word order position.

When a language is selected as Free word order, we employ a secondary heuristic to decide if it is actually V2 word order. The V2 order cannot be easily recognized only with the binary word orders, so it is not given a unique point in the three-dimensional space. Rather, we try to recognize it by comparing the ternary orders. A Free-order language is reclassified as V2 if SVO and OVS occur more frequently than SOV and OSV.[5]

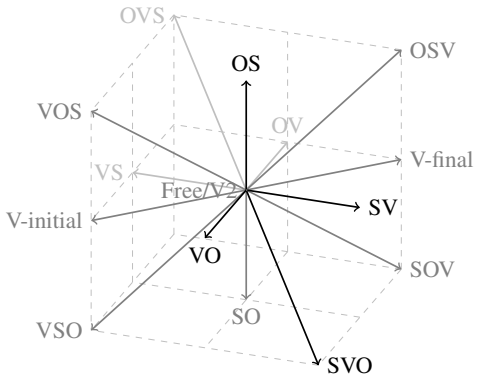

Figure 2: Three axes of basic word order and the positions of canonical word orders.

4.2 Results

Table 2 shows the results we obtained for our dev and test sets. For comparison, we use a most-

[4]Of the eight vertices of this cube, six represent canonical word orders the other two impossible combinations: The vertex for (SV, VO, OS) (e.g.) has S both before and after O.

[5]The VOS and VSO patterns are excluded from this comparison, since they can go either way—there may be unaligned constituents (i.e. not a S, O, or V) before the verb which are ignored by our system.

frequent-type baseline, selecting SOV for all languages, based on Dryer's (2011) survey. We get high accuracy for DEV1, low accuracy for DEV2, and moderate accuracy for TEST, but all are significantly higher than the baseline.

Dataset	Inferred WO	Baseline
DEV1	0.900	0.200
DEV2	0.500	0.100
TEST	0.727	0.091

Table 2: Accuracy of word-order inference

Hand analysis of the errors in the dev sets show that some languages fall victim to the testsuite bias, such as Russian, Quechua, and Tamil. All of these languages have Free word order, but our system infers SVO for Russian and SOV for Quechua and Tamil, because the authors of the testsuites used one order significantly more than the others. Similarly, the Free word order language Nishnaabemwin is inferred as V2 because there are more SVO and OVS patterns given than others. We also see errors due to misalignment from RiPLes' syntactic projection. The VSO language Welsh is inferred as SVO because the near-ubiquitous sentence-initial auxiliary doesn't align to the main verb of the English translation.

5 Inferring Case Systems

Case refers to linguistic phenomena in which the form of a noun phrase (NP) varies depending on the function of the NP in a sentence (Blake, 2001). The Grammar Matrix's case library (Drellishak, 2009) focuses on case marking of core arguments of verbs. Specifying a grammar for case involves both choosing the high-level case system to be modeled as well as associating verb types with case frames and defining the lexical items or lexical rules which mark the case on the NPs. Here, we focus on the high-level case system question as it is logically prior, and in some ways more interesting than the lexical details: Answering this question requires identifying case frames of verbs in particular examples and then comparing across those examples, as described below.

The high-level case system of a language concerns the alignment of case marking between transitive and intransitive clauses. The three elements in question are the subjects of intransitives (dubbed S), the subjects (or agent-like arguments) of transitives (dubbed A) and the objects (or patient-like arguments) of intransitives (O). Among languages which make use of case, the most common alignment type is a nominative-accusative system (Comrie 2011a,b). In this type, S takes the same kind of marking as A.[6] The Grammar Matrix case library provides nine options, including none, nominative-accusative, ergative-absolutive (S marked like O), tripartite (S, A and O all distinct) and several more intricate types. For example, in a language with one type of split case system the alignment is nominative-accusative in non-past tense clauses, but ergative-absolutive in past tense ones.

Case system	Case grams present: NOM ∨ ACC	Case grams present: ERG ∨ ABS
none		
nom-acc	✓	
erg-abs		✓
split-erg (conditioned on V)	✓	✓

Table 3: GRAM case system assignment rules

As with major constituent word order, the constraints implementing a case system in a grammar serve to model both grammaticality and the mapping between syntactic and semantic arguments. Here too, the distribution of tokens may be something other than a pure expression of the case alignment type. Sources of noise in the distribution include: argument optionality (e.g., transitives with one or more covert arguments), argument frames other than simple intransitives or transitives, and quirky case (verbs that use a non-standard case frame for their arguments, such as the German verb *helfen* which selects a dative argument, though the language's general system is nominative-accusative (Drellishak, 2009)).

5.1 Methodology

We explore two possible methodologies for inferring case systems, one relatively naïve and one more elaborate, and compare them to a most-frequent-type baseline. Method 1, called GRAM, considers only the gloss line of the IGT and assumes that it complies with the Leipzig Glossing Rules (Bickel et al., 2008). These rules not only prescribe formatting aspects of IGT but also provide a set of licensed 'grams', or tags for grammatical properties that appear in the gloss line. GRAM scans for the grams associated with case, and assigns case systems according to Table 3.

This methodology is simple to implement and

[6] English's residual case system is of this type.

expected to work well given Leipzig-compliant IGT. However, since it does not model the function of case, it is dependent on the IGT authors' choice of gram symbols, and may be confused by either alternative case names (e.g., SBJ and OBJ for nominative and accusative or LOC for ergative in languages where it is homophonous with the locative case) or by other grams which collide with the case name-space (such as NOM for nominalizer). It also only handles four of the nine case systems (albeit the most frequent ones).

Method 2, called SAO, is more theoretically motivated, builds on the RiPLes approach used in inferring word order, and is designed to be robust to idiosyncratic glossing conventions. In this methodology, we first identify the S, A and O arguments by projecting the information from the parse of the English translation (including the function tags) to the source sentence (and its glosses). We discard all items which do not appear to be simple transitive or intransitive clauses with all arguments overt, and then collect all grams for each argument type (from all words within in the NP, including head nouns as well as determiners and adjectives). While there are many grammatical features that can be marked on NPs (such as number, definiteness, honorifics, etc.), the only ones that should correlate strongly with grammatical function are case-marking grams. Furthermore, in any given NP, while case may be multiply marked, we only expect one type of case gram to appear. We thus assume that the most frequent gram for each argument type is a case marker (if there are any) and assign the case system according to the following rules, where S_g, O_g and A_g denote the most frequent grams associated with these argument positions, respectively:

- Nominative-accusative: S_g=A_g, $S_g \neq O_g$
- Ergative-absolutive: S_g=O_g, $S_g \neq A_g$
- No case: S_g=A_g=O_g, or $S_g \neq A_g \neq O_g$ and S_g, A_g, O_g also present on each of the other argument types
- Tripartite: $S_g \neq A_g \neq O_g$, and S_g, A_g, O_g (virtually) absent from the other argument types
- Split-S: $S_g \neq A_g \neq O_g$, and A_g and O_g are both present in the list for the S argument type

Here, we're using Split-S to stand in for both Split-S and Fluid-S. These are both systems where some S arguments are marked like A, and some like O. In Split-S, which is taken depends on the verb. In Fluid-S, it depends on the interpretation of the verb. These could be distinguished by looking for intransitive verbs that appear more than once in the data and checking whether their S arguments all have consistently A or O marking.

This system is agnostic as to the spelling of the case grams. By relying on more analysis of the IGT than GRAM, it also introduces new kinds of brittleness. Recognizing the difference between grams being present and (virtually) absent makes the system susceptible to noise.

5.2 Results

Table 4 shows the results for the inference of case-marking systems. Currently GRAM performs best, but both methods generally perform better than the baseline. The better performance of GRAM is expected, given the small size and generally Leipzig-compliant glossing of our data sets. In future work, we plan to incorporate data from ODIN, which is likely less consistently annotated but more voluminous, and we expect SAO to be more robust than GRAM to this kind of data.

Dataset	GRAM	SAO	Baseline
DEV1	0.900	0.700	0.400
DEV2	0.900	0.500	0.500
TEST	0.545	0.545	0.455

Table 4: Accuracy of case-marking inference

We find that GRAM is sometimes able to do well when RiPLes gives alignment errors. For example, Old Japanese is a NOM-ACC language, but the case-marking grams (associated to postpositions) are not aligned to the NP arguments, so SAO is not able to judge their distribution. On the other hand, SAO prevails when non-standard grams are used, such as the NOM-ACC language Hupdeh, which is annotated with SUBJ and OBJ grams. This complementarity suggests scope for system combination, which we leave to future work.

6 Discussion and Future Work

Our initial results are promising, but also show remaining room for improvement. Error analysis suggests two main directions to pursue:

Overcoming testsuite bias In both the word order and case system tasks, we see the effect of testsuite bias on our system results. The testsuites for freer word order languages can be artificially dominated by a particular word order that the testsuite author found convenient. Further, the restricted vocabulary used in testsuites, combined

with a general preference for animates as subjects, leads to stems and certain grams potentially being misidentified as case markers.

We believe that these aspects of testsuite bias are not typical of our true target input data, viz., the larger collections of IGT created by field projects. On the other hand, there may be other aspects of testsuites which are simplifying the problem and to which our current methods are overfitted. To address these issues, we intend to look to larger datasets in future work, both IGT collections from field projects and IGT from ODIN. For the field projects, we will need to construct choices files. For ODIN, we can search for data from the languages we already have choices files for.

As we move from testsuites to test corpora (e.g., narratives collected in documentary linguistics projects), we expect to find different distributions of word order types. Our current methodology for extracting word order is based on idealized locations in our word order space for each strict word order type. Working with naturally occurring corpora it should be possible to gain a more empirically based understanding of the relationship between underlying word order and sentence type distributions. It will be particularly interesting to see how stable these relationships are across languages with the same underlying word order type but from different language families and/or with differences in other typological characteristics.

Better handling of unaligned words The other main source of error is words that remain unaligned in the projected syntactic structure and thus only loosely incorporated into the syntax trees. This includes items like case marking adpositions in Japanese, which are unaligned because there is no corresponding word in English, and auxiliaries in Welsh, which are unaligned when the English translation doesn't happen to use an auxiliary. In the former case, our SAO method for case system extraction doesn't include the case grams in the set of grams for each NP. In the latter, the word order inference system is unable to pick up on the VSO order represented as Aux+S+[VP]. Simply fixing the attachment of the auxiliaries will not be enough in this case, as the word order inference algorithm will need to be extended to handle auxiliaries, but fixing the alignment is the first step. Alignment problems are also the main reason our initial attempts to extract information about the order of determiners and nouns haven't yet been able to beat the most-frequent-type baseline.

Better handling of these unaligned words is a non-trivial task, and will require bringing in sources of knowledge other than the structure of the English translation. The information we have to leverage in this regard comes mainly from the gloss line and from general linguistic/typological knowledge which can be added to the algorithm. That is, there are types of grams which are canonically associated with verbal projections and types of grams canonically associated with nominal projections. When these grams occur on unaligned elements, we can hypothesize that the elements are auxiliaries and case-marking adpositions respectively. Further typological considerations will motivate heuristics for modifying tree structures based on these classifications.

Other directions for future work include extending this methodology to other aspects of grammatical description, including additional high-level systems (e.g., argument optionality), discovering the range of morphosyntactic features active in a language, and describing and populating lexical types (e.g., common nouns with a particular gender). Once we are able to answer enough of the questionnaire that the customization system is able to output a grammar, interesting options for detailed evaluation will become available. In particular, we will be able to parse the IGT (including held-out examples) with the resulting grammar, and then compare the resulting semantic representations to those produced by parsing the English translations with tools that produce comparable semantic representations for English (using the English Resource Grammar (Flickinger, 2000)).

7 Conclusions and Future Work

In this paper we have presented an approach to combining two types of linguistic resources—IGT, as produced by documentary linguists and a cross-linguistic grammar resource supporting precision parsing and generation—to create language-specific resources which can help enrich language documentation and support language revitalization efforts. In addition to presenting the broad vision of the project, we have reported initial results in two case studies as a proof-of-concept. Though there is still a ways to go, we find these initial results a promising indication of the approach's ability to assist in the preservation of the key type of cultural heritage that is linguistic systems.

Acknowledgments

We are grateful to the students in Ling 567 at the University of Washington who created the test-suites and choices files used as development and test data in this work and to the three anonymous reviewers for helpful comments and discussion.

This material is based upon work supported by the National Science Foundation under Grant No. BCS-1160274. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Steven Abney and Steven Bird. 2010. The human language project: building a universal corpus of the world's languages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 88–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dorothee Beerman and Lars Hellan. 2011. Inducing grammar from IGT. In *Proceedings of the 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2011)*.

Dorothee Beermann and Pavel Mihaylov. 2009. TypeCraft: Linguistic data and knowledge sharing, open access and linguistic methodology. Paper presented at the Workshop on Small Tools in Cross-linguistic Research, University of Utrecht. The Netherlands.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation*, pages 1–50. 10.1007/s11168-010-9070-1.

Emily M. Bender, Sumukh Ghodke, Timothy Baldwin, and Rebecca Dridan. 2012a. From database to treebank: Enhancing hypertext grammars with grammar engineering and treebank search. In Sebastian Nordhoff and Karl-Ludwig G. Poggeman, editors, *Electronic Grammaticography*, pages 179–206. University of Hawaii Press, Honolulu.

Emily M. Bender, David Wax, and Michael Wayne Goodman. 2012b. From IGT to precision grammar: French verbal morphology. In *LSA Annual Meeting Extended Abstracts 2012*.

Emily M. Bender. 2007. Combining research and pedagogy in the development of a crosslinguistic grammar resource. In Tracy Holloway King and Emily M. Bender, editors, *Proceedings of the GEAF 2007 Workshop*, Stanford, CA. CSLI Publications.

Balthasar Bickel, Bernard Comrie, and Martin Haspelmath. 2008. The Leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. Max Planck Institute for Evolutionary Anthropology and Department of Linguistics, University of Leipzig.

Barry J. Blake. 2001. *Case*. Cambridge University Press, Cambridge, second edition.

Bernard Comrie. 2011a. Alignment of case marking of full noun phrases. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich.

Bernard Comrie. 2011b. Alignment of case marking of pronouns. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich.

Hal Daumé III and Lyle Campbell. 2007. A Bayesian model for discovering typological implications. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 65–72, Prague, Czech Republic, June. Association for Computational Linguistics.

Scott Drellishak. 2009. *Widespread But Not Universal: Improving the Typological Coverage of the Grammar Matrix*. Ph.D. thesis, University of Washington.

Matthew S. Dryer. 2011. Order of subject, object and verb. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich.

Scott Farrar and Terry Langendoen. 2003. A linguistic ontology for the semantic web. *Glot International*, 7:97–100.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):15 – 28.

Ryan Georgi, Fei Xia, and William Lewis. 2010. Comparing language similarity across genetic and typologically-based groupings. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 385–393, Beijing, China, August. Coling 2010 Organizing Committee.

Ryan Georgi, Fei Xia, and William Lewis. 2012. Improving dependency parsing with interlinear glossed text and syntactic projection. In *Proceedings of COLING 2012: Posters*, pages 371–380, Mumbai, India, December.

Ken Hale, Michael Krauss, Lucille J. Watahomigie, Akira Y. Yamamoto, Colette Craig, LaVerne Masayesva Jeanne, and Nora C. England. 1992. Endangered languages. *Language*, 68(1):pp. 1–42.

Martin Haspelmath, Matthew S. Dryer, David Gil, and Bernard Comrie, editors. 2008. *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich. http://wals.info.

Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Hervè Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. 1996. TSNLP: Test suites for natural language processing. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 711–716, Stroudsburg, PA, USA. Association for Computational Linguistics.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 685–690, Hyderabad, India.

William D. Lewis. 2006. ODIN: A model for adapting and enriching legacy infrastructure. In *Proceedings of the e-Humanities Workshop, Held in cooperation with e-Science*, Amsterdam.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637, Jeju Island, Korea, July. Association for Computational Linguistics.

Sebastian Nordhoff and Karl-Ludwig G. Poggeman, editors. 2012. *Electronic Grammaticography*. University of Hawaii Press, Honolulu.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Carmela Toews. 2009. The expression of tense and aspect in Shona. *Selected Proceedings of the 39th Annual Converence on African Linguistics*, pages 32–41.

Tony Woodbury. 2003. Defining documentary linguistics. *Language documentation and description*, 1(1):35.

Fei Xia and William D. 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 D. Lewis. 2008. Repurposing theoretical linguistic data for tool development and search. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 529–536, Hyderabad, India.

Fei Xia and William Lewis. 2009. Applying NLP technologies to the collection and enrichment of language data on the web to aid linguistic research. In *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, March. Association for Computational Linguistics.