Handling cross-cutting properties in automatic inference of lexical classes: A case study of Chintang

Olga Zamaraeva[*]
University of Washington
Department of Linguistics
olzama@uw.edu

Kristen Howell[*]
University of Washington
Department of Linguistics
kphowell@uw.edu

Emily M. Bender
University of Washington
Department of Linguistics
ebender@uw.edu

Abstract

In the context of the ongoing AGGREGATION project concerned with inferring grammars from interlinear glossed text, we explore the integration of morphological patterns extracted from IGT data with inferred syntactic properties in the context of creating implemented linguistic grammars. We present a case study of Chintang, in which we put emphasis on evaluating the accuracy of these predictions by using them to generate a grammar and parse running text. Our coverage over the corpus is low because the lexicon produced by our system only includes intransitive and transitive verbs and nouns, but it outperforms an expert-built, oracle grammar of similar scope.

1 Introduction

Machine-readable grammars are useful for linguistic hypothesis testing via parsing and treebanking (selecting the best parse for each sentence) because they represent internally coherent models and can be explored automatically (Bender et al., 2011, 2012a; Fokkens, 2014; Müller, 1999, 2016). Multilingual grammar engineering frameworks such as CoreGram (Müller, 2015) and the LinGO Grammar Matrix (Bender et al., 2002, 2010) facilitate the development of machine-readable grammars by providing shared analyses, but still require significant human effort to select the appropriate analyses for a given language. To partially automate this process, the AGGREGATION project takes advantage of the stored analyses in the Grammar Matrix and the linguistic information encoded in Interlinear Glossed Text (IGT). While at this stage the project efforts are mostly experimental in nature and focus on evaluating grammars obtained in this way, there already have been successful collaborations with documentary linguists which in at least one case led to insights into the language's morphological patterns (Zamaraeva et al., 2017).

The IGT data format, widely used by linguists, is well suited to inference tasks because it features detailed morpheme-by-morpheme annotation and translations to high-resource languages (Xia and Lewis, 2007; Bender et al., 2013). However, the inference processes required are heterogeneous. On the morphological side, an inference system identifies position and inflectional classes. On the syntactic side, an inference system uses syntactic generalizations to identify broad characteristics and defines lexical classes according to syntactic properties. The challenge we address here is in integrating the two. In this paper, we integrate a system that identifies morphological patterns in IGT data with one that predicts syntactic properties to define lexical classes that encode both morphological and syntactic features. We evaluate by replicating the case study of Bender et al. (2014), in which they automatically produced separate grammar fragments based on morphotactic and syntactic information and evaluated their system on a corpus of Chintang. We compare their results to our integrated system which includes both morphotactic and syntactic properties.[1]

2 Chintang

Chintang (ISO639-3: ctn) is a Sino-Tibetan language spoken by ∼5000 people in Nepal (Bickel et al., 2010; Schikowski et al., 2015). Here we summarize the characteristics of the language that are directly relevant to this case study.

The relative order of the verb and its core arguments (hereafter 'word order') in Chintang is described as free by Schikowski et al. (2015), in that all verb and argument permutations are valid

[*] The first two authors made equal contribution.

[1] Our code and sample data are available here: https://git.ling.washington.edu/agg/repro/computel3-ctn.

Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages: Vol. 1 Papers, pages 28–38,
Honolulu, Hawai'i, USA, February 26–27, 2019.

Tool	Task
SIL Toolbox	Export original IGT data to plain text
Xigt	Convert data into a robust data model with associated processing package
INTENT	Enrich the data: add phrase structure, POS tags to translation line and project to source language line
Inference code	**Create grammar specification: Case system, case frames of verbs**
MOM	**Create grammar specification: Inflection and position classes**
Grammar Matrix	Create grammar on the basis of specifications
LKB	Run the grammar on held-out sentences
[incr tsdb()]	Treebank: Inspect the parses for correctness

Table 1: AGGREGATION pipeline, **bold** indicates this paper's contribution

in the language, with the felicity of these combinations being governed primarily by information structure. Although Schikowski et al. (2015) note that no detailed analysis has been carried out regarding what other factors condition word order, they say that SV, APV and AGTV are the most frequent orders that they observe.[2] Dropping of core arguments is also common in the language (Schikowski et al., 2015).

The case system follows an ergative-absolutive pattern, with some exceptions (Stoll and Bickel, 2012; Schikowski et al., 2015). In ergative-absolutive languages, the subject of an intransitive verb has the same case marking as the most patient-like argument of a transitive verb, typically referred to as absolutive. The most agent-like argument of a transitive verb has a distinct case marking, usually called ergative. In Chintang, ergative case is marked with an overt marker while absolutive is zero-marked. A number of exceptions to the ergative-absolutive pattern arise due to valence changing operations (such as reflexive and benefactive). Other exceptions include variable ergative marking on first and second person pronouns and an overt absolutive marker on the pronoun *sa-* 'who'.[3]

Chintang's flexible word order, scarcity of overt case marking, and frequent argument dropping introduce challenges to grammar inference. First, the variety of phrase structure rules required to accommodate free word order in addition to argument optionally introduces potential for ambiguity to any implemented grammar. In some cases, this ambiguity is legitimate (in that multiple parses map to multiple semantic readings), but in other cases it may be an indication of under-constrained rules. Second, the lack of overt absolutive case marking in Chintang, together with common pronoun dropping, results in relatively few overt case morphemes in the corpus for a syntactic inference script to use.

3 AGGREGATION

The AGGREGATION project[4] is dedicated to automating the creation of grammars from IGT.[5] Table 1 presents all the tools involved in the pipeline, with information on which task each performs. We elaborate on the pieces of the pipeline below and encourage the reader to refer back to this table as needed to track what each component is.

As part of the AGGREGATION project, Bender et al. (2014) present the first end-to-end study of grammar inference from IGT by extracting broad syntactic properties (namely, word order, case system and case frame) and morphological patterns and testing the coverage of the resulting grammars on held out data. They used the methodology of Bender et al. (2013) for syntactic inference and Wax (2014) for morphological inference. However, they left integrating the two and creating grammars which benefit from both types of information to future work.

Like Bender et al. (2014), we take advantage of the Grammar Matrix customization system (Bender et al., 2002, 2010), which creates precision grammars that emphasize syntactically accurate coverage and attempt to minimize ambiguity. It uses stored analysis for particular phenomena to

[2] S = subject, P = patient, G = goal, T = theme, A = agent

[3] For a much more detailed account of the case frame for various verb types, see Schikowski et al. 2015.

[4] http://depts.washington.edu/uwcl/aggregation/

[5] We are aware of one project with similar goals: TypeGram (see e.g. Hellan and Beermann, 2014), couched in HPSG as well. Our pipeline places fewer expectations on the IGT annotation, inferring phrase structure, part of speech, and valence automatically.

create a grammar based on a user's specification of linguistic properties. These specifications are recorded in a 'choices file'. We follow the methodology of Bender et al. (2014) of formatting the output of our inference systems as a choices file, so that it can be directly input to the Grammar Matrix for grammar customization.

Unlike Bender et al. (2014), we take advantage of the Xigt data model (Goodman et al., 2015). This extensible and flexible format encodes the information in IGT data in such a way that relations between bits of information, such as the connection between a morpheme and its gloss, can be easily identified. Data encoded with Xigt is compatible with the INTENT system for enriching IGT by parsing the English translation and projecting that information onto the source language text (Georgi, 2016). Where Bender et al. (2014) use the methodology of Xia and Lewis (2007), in this work we use INTENT. We also use updated, Xigt-compatible versions of morphological and lexical class inference (Zamaraeva, 2016; Zamaraeva et al., 2017) and case system inference (Howell et al., 2017).

4 Methodology

Our goal is to maximize the information that we can learn about a language both morphologically and syntactically in order to produce grammars that parse strings with minimal ambiguity. We present a methodology that analyzes morphological and syntactic information independently and then creates lexical classes that share the information from both analyses.

4.1 Morphotactic inference with MOM

MOM is a system which infers morphotactic graphs from IGT. Developed originally by Wax (2014) to infer position classes, it was updated to work with the Xigt data model by Zamaraeva (2016) and to infer inflectional classes by Zamaraeva et al. (2017). Below we summarize the main inference algorithm shared across these different versions. As with most work using MOM and the Grammar Matrix, we target the morpheme-segmented line, and assume that the grammars produced will eventually need to be paired with a morphophonological analyzer to map to surface spelling or pronunciation.

MOM starts by reading in IGT that has been enriched with information about each morpheme,

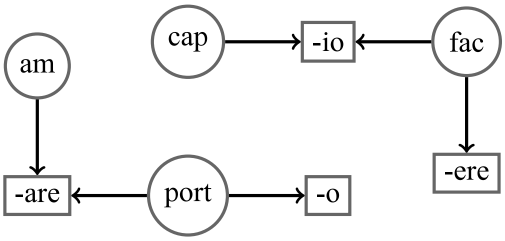

Figure 1: Sample graph MOM will initially build on the example training data consisting of Latin verbs *am-are* ('to love'), *port-are* ('to carry'), *port-o* ('I carry'), *cap-io* ('I take'), *fac-io* ('I do'), and *fac-ere* ('to do').

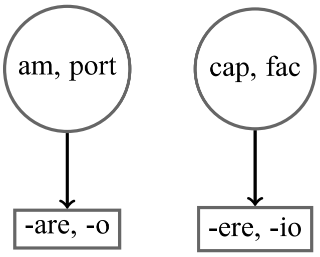

Figure 2: Sample MOM output, after compressing the graph in Figure 1 with a 50% overlap value.

including a part of speech (POS) tag of the word it belongs to as well as whether it is an affix or a stem. Then MOM iterates over the items with the relevant POS tag and builds a graph where nodes are stems and affixes while edges are input relationships. For example, if MOM sees the Latin verbs *am-are* ('to love'), *port-are* ('to carry'), *port-o* ('I carry'), *cap-io* ('I take'), *fac-io* ('I do'), and *fac-ere* ('to do') in the training data, it will create nodes and edges as depicted in Figure 1. Finally, after the original graph has been constructed, MOM will recursively merge nodes which have edge overlap above the threshold provided by the user (e.g. 50%, see Figure 2), in order to discover position classes for affixes and inflectional classes for stems.

4.2 Case and Valence Inference

Whereas MOM creates lexical classes based on morphotactics, the inference system described in this section creates lexical classes for verbs based on their valence in two steps: We first infer the overall case system of the language and second infer whether each verb in the training data is transitive or intransitive. We then create lexical classes that specify the argument structure and case requirements on those arguments.

We begin by predicting the overall case system, using the methodology developed by Bender et al. (2013) and updated by Howell et al. (2017). We reimplement Bender et al.'s GRAM method, which counts the case grams in the corpus and uses a heuristic to assign nominative-accusative, ergative-absolutive, split-ergative or none to the language based on the relative frequencies of the case grams. We apply one change to this system: because the split-ergative prediction does not give any information regarding the nature of the split (e.g. whether it is split according to features of nouns or verbs), we map the split-ergative (which it predicts for Chintang) to ergative-absolutive.

To determine the transitivity of verbs in the corpus, we take advantage of the English translation in the IGT. The dataset, which has been partially enriched by INTENT (Georgi, 2016),[6] includes parse trees and POS tags for the English translation. Furthermore, the Chintang words in the dataset are annotated for part of speech by the authors of the corpus. To infer transitivity using this information, we first do a string comparison between the gloss of the Chintang verb and each word in the language line to identify the English verb that it corresponds to. If no match is found, for example the verb is glossed with *have*, but the English translation contains *get* instead, the verb is skipped. However, if a match is found, we traverse the English parse tree to check if the V is sister to an NP. If so, it is classified as transitive, otherwise it is classified as intransitive. We exclude passive sentences from consideration; however, under this algorithm, verbs that take a verbal complement are classified as intransitive. We leave further fine-tuning in this respect to future work. Finally, once we have the case system and the transitivity for each verb, we assign the verb's case frame according to a heuristic specific to the case system. In the case of ergative-absolutive, we specify ergative case on the subject of transitive verbs and absolutive on the subject of intransitive verbs and object of transitive verbs.

4.3 Extensions to Morphological Analysis

We modified the MOM system by extending the data structure that it expects as input with fields for transitivity and case frame and by making it check these fields every time it considers which lexical class to assign to an item (we describe this process in more detail in §4.4). We added functionality so that MOM infers case lexical rules for nouns. Next, we added functions to collapse all homonym stems in each class into one stem with a disjunctive predication (e.g. stem *bekti*, predication *_youngest.son-or-youngest.male.sibling_n_rel*).[7] Furthermore, we improved the way MOM is dealing with stems that occur bare in the data. Previously, MOM put all bare stems into a lexical class which could not merge with inflecting classes later, even if the same stem occurred with affixes in other portions of the data. Our fixing this problem led to fewer lexical classes in the grammar, which means better coverage potential. Finally, we made necessary additions to MOM's output format so that all relevant information for each lexical class, including case features and transitivity values, would be encoded into a valid choices file that can be customized by the Grammar Matrix.

4.4 Integrating Inference Systems

We integrated the systems described in §4.2 and §4.3 such that lexical types are defined according to the output of both systems. We expect our grammars produced in this fashion to outperform either of the kinds of grammars produced by Bender et al. (2014), because our integrated verb classes contain both morphological and valence constraints, rather than leaving one of these categories underspecified.

We run the syntactic inference system before MOM, so that transitivity and case frame are included in MOM's input. While MOM builds lexical classes based on morphology, it also checks transitivity and case frame for compatibility before adding an item to a lexical class. Verbs for which there is no case frame prediction are classified in a 'dummy' category, rather than being thrown out in order to maximize the morphological patterns that can be detected by MOM. That is, we want to include these verbs in the graph during the morphological inference process.[8] However, without constraints on their valence, their inclusion in the final grammar would result in spurious ambiguity. Therefore, we allow these verbs in the grammar, but change their orthography to contain

[6] For this particular dataset, INTENT was not able to produce bilingual alignments. This means that we were not able to take advantage of word to word mappings.

[7] This follows the conventions of Minimal Recursion Semantics, see Copestake et al. 2005 and Flickinger et al. 2014.

[8] As will be mentioned in §5.2.4, we also allow MOM to merge together verbs with no valence information into lexical classes with valence constraints, based on morphological patterns.

Grammar spec	Origin	description
ORACLE	Bender et al. 2012b	Expert-constructed
BASELINE	Bender et al. 2014	Full-form lexicon, free word order (WO), no case
MOM-DEFAULT-NONE	Bender et al. 2014	MOM-inferred lexicon, free WO, no case
FF-AUTO-GRAM	Bender et al. 2014	Full-form lexicon, V-final WO, erg-abs case system
INTEGRATED	This paper	MOM-inferred lexicon, case frames, free WO, erg-abs

Table 2: Grammars used in evaluation

a non-Chintang string, so that they are effectively excluded from the working grammar unless they were merged into a verb class with valid valence constraints.

5 Case Study

So far we have described a methodology that extracts both syntactic and morphological information from IGT data, and outputs this information in a format compatible with the Grammar Matrix customization system. Now we present a case study of this system on a Chintang dataset, described in §5.1. We describe our comparison grammars as well as the output of our system (§5.2) and grammar development and parsing process (§5.3) before presenting the results in §6 and the discussion in §7.

5.1 Dataset

For our experiment, we use the same subset of the Chintang Language Resource Program (CLRP)[9] corpus as Bender et al. (2014), which contains 10862 instances of Interlinnear Glossed Text. These IGT comprise narratives and other recorded speech that were transcribed, translated and glossed by the CLRP. Example (1) illustrates the thorough glossing of IGT in this corpus. However, it is noteworthy, especially for the purposes of inference, that syntactic characteristics that do not correspond with an overt morpheme (such as SG and ABS) are not glossed in this data. We use the same train (8863 IGT), dev (1069 IGT) and test (930 IGT) splits as Bender et al. (2014).

(1) unisaŋa khatte
u-nisa-ŋa khatt-e
3sPOSS-younger.brother-ERG.A take-IND.PST
mo kosiʔ moba
mo kosi-iʔ mo-pe
DEM.DOWN river-LOC DEM.DOWN-LOC
'The younger brother took it to the river.' [ctn] (Bickel et al., 2013)

```
section=word-order
word-order=free
has-dets=no
...
section=case
case-marking=erg-abs
erg-abs-erg-case-name=erg
erg-abs-abs-case-name=abs
  case1_name=loc
  case2_name=dat...
...
verb1029_valence=intrans
verb1029_stem2_pred=_cry_v_rel
verb1029_stem2_orth=ratt
...
verb-pc6_inputs=verb-pc1, verb1029
verb-pc6_lrt1_lri1_orth=-a
verb-pc6_lrt1_lri1_inflecting=yes
```

Figure 3: Excerpts from a choices file

We use a version of the training portion of the data that has been converted to the Xigt data model. The converter skipped 169 IGT, resulting in a slightly smaller training set than that used by Bender et al. (2014). This dataset was enriched with English parses and part of speech tags using INTENT (Georgi, 2016).

5.2 Grammar Specification

The output of the morphotactic and syntactic inference scripts described earlier in this section is encoded in a 'choices file', illustrated in Figure 3, which is the suitable input to the Grammar Matrix customization system (Bender et al., 2010). In this subsection, we describe the choices files we developed for the baseline, oracle and our inference system, as well as the comparison choices from the 2014 experiment.[10]

5.2.1 Oracle Choices File

Our first point of comparison is a manually constructed 'oracle' choices file, from Bender et al.

[9] https://www.uzh.ch/clrp/

[10] For this comparison, we regenerated the BASELINE and the ORACLE grammars from their choices files. For MOM-DEFAULT-NONE and FF-AUTO-GRAM we worked from the same parsing results analyzed in Bender et al. 2014.

2012b. This choices file was developed by importing CLRP's Toolbox lexicon and defining the rest of the specifications by hand, based on linguistic analysis. As a result, the grammar produced by this choices file is expected to have very high precision, with moderate recall. It specifies the word order as verb-final (corresponding to the most frequent word orders observed by Schikowski et al. (2015)) and the case system as ergative-absolutive, and it defines both subject and object dropping as possible for all verbs. It also includes hand-specified lexical rules based on the analysis in Schikowski 2012.

Because the Grammar Matrix only included simple transitive and intransitive verbs at the time this choices file was developed, only those two classes were defined. Their case frames were specified by hand such that the intransitive verb class has an absolutive subject and the transitive verb class has an ergative subject and an absolutive object. Finally, because the Grammar Matrix did not support adjectives, adverbs, and many other lexical categories, the resulting grammar is not expected to parse any sentence containing those lexical categories. Conveniently, the limitations of the grammar defined by Bender et al. (2012b) make it a good comparison to the grammar we are able to produce using only the inference described in this paper.[11]

5.2.2 Baseline Choices File

As a second point of comparison, we use a choices file that is designed to create a working grammar with a sufficient lexicon that is otherwise naïve with respect to the particular grammatical characteristics of Chintang. We take this choices file from Bender et al. (2014), but make some modifications. The lexicon was extracted according to the methodology of Xia and Lewis (2007), defining bare bones classes for nouns, verbs and determiners. Because the inference system we use in the this paper does not consider determiners, we removed them from the baseline choices file. Finally, the baseline predicts free word order, no case system and argument dropping for subjects and objects of all verbs because these choices are expected to result in the highest coverage (though low precision) for any language.[12]

5.2.3 Bender et al. (2014) choices files

Our final comparison is to the results for the choices files created by Bender et al. (2014) that are most comparable to our present work. We look at their MOM-DEFAULT-NONE and FF-AUTO-GRAM grammars which represent the two types of inference that we have integrated.

The MOM-DEFAULT-NONE choices file uses a lexicon produced by MOM and the default choices for word order (free), case system (none), and argument optionality (all arguments are optional). The FF-AUTO-GRAM uses a lexicon of full word forms, as in the baseline but with case frames for each verb as observed in the data. The word order (V-final) and case system (ergative-absolutive) were inferred using syntactic inference.

5.2.4 Integrated Inference Output

We produced the inferred choices file by extracting the lexicon, case system and morphological rules, as described in §4.1, §4.2 and §4.3. We ran the inference system on training data, debugging and selecting an overlap value by parsing the dev data with the resulting grammars. Our system predicted ergative-absolutive case and a robust lexicon and set of morphological rules. We used the default word order (free) and default argument optionality choices (any verb can drop subject or object) from Bender et al. 2014. We also added the necessary choices for the 'dummy verb' category described in §4.3.

Choosing the overlap value We ran MOM with 10 overlap values from 0.1 to 1.0. Then we parsed sentences from the development set to identify the grammar that optimizes both coverage and ambiguity. For these data, that showed the optimum overlap value to be 0.2.[13] We inspected the differences in coverage and verified that they fall into three categories. First, some lexical entries which originally lacked a valence frame (the inference script was not able to assign one) successfully merged into lexical classes which did have a valid valence frame.[14] Second, some entries happened to merge into only an intransitive class in one grammar but only into a transitive class in another. Finally, upon merging into lexical classes,

[11] We do still have to add some default choices to our inferred grammar, as described in §5.2.4.

[12] It is incidental that these word order and argument dropping predictions are arguably valid analyses for this language.

[13] The 0.1 grammar had marginally lower ambiguity (2.76 fewer parses on average), but it parsed two fewer sentences.

[14] We assume that if a verb without a known valence frame has similar morphotactic possibilities as a verb with a known valence frame, it has the same valence frame.

lexical entries gained access to morphological patterns which were unseen in the training data. This happens with both noun and verb lexical entries. For example, not all grammars that we produced were able to successfully produce a parse tree for (2), because they did not merge the stem *sil* with a class that is compatible with the prefix *u-*.

(2) u- sil -u -set -kV
3SA- bite.and.pull.out -3P -DESTR.TR -3P
-ce
-IND.NPST -3NSP
'They snatch and kill them' [ctn] (Bickel et al., 2013)

5.3 Grammar customization and parsing

After producing the choices files, we used the Grammar Matrix customization system[15] to produce customized grammars. We loaded these into the LKB parsing software (Copestake, 2002) and used [incr tsdb()] (Flickinger and Oepen, 1998) to create stored profiles of syntactic and semantic parses. We treebanked these parses using [incr tsdb()] to identify correct parses, or parses that produce the desired semantic representation for the sentence. In particular, we checked to make sure that the predicate-argument structure matched what was indicated in the gloss, but did not require information such as negation, person and number or tense and aspect (all morphologically marked in Chintang), as our system doesn't yet attempt to extract these.

6 Results

To put the results of parsing the strings from Chintang in context, we first describe the produced grammars in terms of their size. Table 3 reports the size of the lexicons and the number of affixes of each grammar. The ORACLE grammar's lexicon includes the imported Toolbox lexicon from CLRP, so it includes many more stems than the others. The BASELINE and FF-AUTO-GRAM lexicons include full form lexical entries,[16] while the grammar produced by our INTEGRATED system has lexicons that include stems extracted from the training data. MOM-DEFAULT-NONE only did morphological analysis on verbs; for nouns it includes full-form entries. The ORACLE grammar has a number of morphological rules for nouns and verbs that were hand-crafted, while MOM-DEFAULT-NONE and INTEGRATED's lexical rules were extracted from the training data.

The results are reported in Table 4. 'Lexical coverage' is the number of sentences for which the grammar could produce an analysis (via full form lexical entry or morphological rules) for every word form. These numbers are quite small because there are no lexical entries for categories other than nouns and verbs. 'Parsed' shows the number of sentences in test data that received some spanning syntactic analysis, and 'correct' the number of items for which there was a correct parse, according to manual inspection and treebanking. Finally we report the number of readings, which shows the degree of ambiguity in the grammar. Our INTEGRATED system had the highest coverage as well as the highest correct coverage, but also had the most ambiguity.

7 Discussion

We expect INTEGRATED to have broader coverage than FF-AUTO-GRAM and BASELINE because it includes morphological rules allowing it to generalize to unseen entries. We also expect INTEGRATED to have higher precision (a higher proportion of correct parses) than MOM-DEFAULT-NONE because unlike MOM-DEFAULT-NONE, it integrates case frames which can rule out spurious analyses. We also expect it to have higher coverage than MOM-DEFAULT-NONE because it includes inferred morphological rules for nouns (in addition to verbs). Though the absolute numbers are small, these predictions are borne out by the data in Table 4.

To get a better sense of the differences between the systems, we performed an error analysis. We looked at all the parsed items (not just the treebanked ones) to get a broader view into the behavior of the grammars. This section provides the results of our analysis as well as some exploration into the higher ambiguity found by INTEGRATED.

7.1 Error Analysis

As expected (and as with the other grammars), the parsing errors for INTEGRATED are due to either lexical or syntactic failures. For 825 items, the parser did not succeed in **lexical analysis**. In principle, this includes both lack of stem or affix forms and failures due to the grammar's inability to construct the morphological pattern, even though all morphemes are found in the grammar. We examined a

[15] svn://lemur.ling.washington.edu/shared/matrix/trunk at revision 41969

[16] Note that the FF-AUTO-GRAM grammar only included verbs for which a case frame could be predicted.

Choices file	# verb entries	# noun entries	# verb affixes	# noun affixes
ORACLE	899	4750	233	36
BASELINE	3005	1719	0	0
FF-AUTO-GRAM	739	1724	0	0
MOM-DEFAULT-NONE	1177	1719	262	0
INTEGRATED	911	1755	220	76

Table 3: Amount of lexical information in each choices file

choices file	lexical coverage (%)	parsed (%)	correct (%)	readings
ORACLE	116 (12.5)	20 (2.2)	10 (1.1)	1.35
BASELINE *	38 (0.4)	15 (1.6)	8 (0.9)	27.67
FF-AUTO-GRAM	18 (1.9)	4 (0.4)	2 (0.2)	5.00
MOM-DEFAULT-NONE	39 (4.2)	16 (1.7)	3 (0.3)	10.81
INTEGRATED	105 (11.3)	32 (3.4)	15 (1.6)	91.56

* We report slightly different results for lexical coverage and average readings for the baseline than Bender et al. (2014) because we removed determiners from the choices file.

Table 4: Results on 930 held-out sentences

sample of 50 such items and only found instances of missing stems and affixes, and no failed combinatorics. The remaining 73 errors are accounted for on the **syntactic level**. These break into three categories: (1) both a V/VP and an NP could be formed, but the NP had a case marker incompatible with the verb's case frame (e.g. locative; 6 items of this kind total);[17] (2) a sentence did not contain a word which could be analyzed as a verb by the grammar (only as a noun, e.g. a sentence fragment; 23 total); (3) finally, the sentence was complex, i.e., contained more than one verb. This third category was the most common (44 total), as the grammar does not include subordination or coordination rules.

We also compared our results on the held-out data to the baseline and the oracle grammars. While INTEGRATED outperforms both BASELINE and ORACLE, BASELINE and ORACLE parse some sentences that INTEGRATED does not.

Integrated vs. oracle Comparison between ORACLE and INTEGRATED yields 130 different results. Of these, most are due to differences in lexical analysis. In particular, there are 55 items which fail due to lexical analysis in INTEGRATED but fail due to syntactic analysis in ORACLE. In all of these cases, our grammar lacked a lexical entry that the oracle grammar had; this is expected as the oracle lexicon is based on a different source. There are 38 items for which ORACLE cannot provide lexical analysis and INTEGRATED can but fails at the syntactic stage. Of these, most are missing stems and affixes in the oracle grammar but for one item, ORACLE is actually lacking the required affix ordering that INTEGRATED picks up from the training data. In addition, there are 11 cases where ORACLE fails at lexical analysis and INTEGRATED succeeds at both lexical and syntactic analysis. In two of those eleven cases, INTEGRATED outperforms ORACLE due to the robustness of the morphological rules, not the lexicon. In contrast, there are 3 items at which INTEGRATED fails lexically and ORACLE gives a parse, all due to lexicon differences. In 6 cases, ORACLE fails at the syntactic stage where INTEGRATED succeeds. Of these, 1 was rejected in treebanking,[18] two items are true wins due to morphotactic inference for nouns; two are because ORACLE only has a noun entry for something which INTEGRATED picked up as a verb, and finally one is parsed by INTEGRATED because it admits head-initial word orders while ORACLE insists on V-final. In contrast, ORACLE can parse one item for which INTEGRATED can perform lexical analysis but fails to parse. The sentence is *cor cor* and ORACLE has both a verb and a noun entry for the word *cor* while INTEGRATED does not.

Integrated vs. Baseline The difference between INTEGRATED and BASELINE is mainly due to lexical coverage. The BASELINE grammar featuring

[17] What is missing here is a grammar rule that handles e.g. locative NPs functioning as modifiers.

[18] If none of the parses have a structure and a semantic representation that are meaningful with respect to the translation, all parses for the item are rejected.

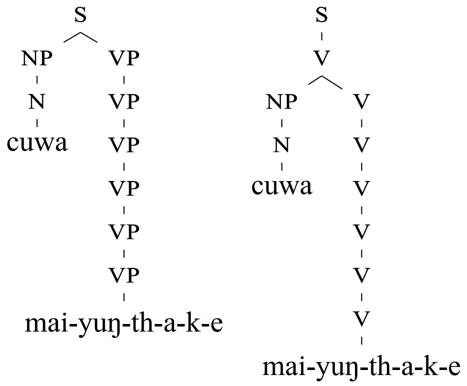

Figure 4: Analyses like the one above (for (3)) use homonymous intransitive (left) and transitive verb entry with a dropped argument (right). Together with the homonymous verb position classes, this produces ambiguity.

full-form lexical entries can lexically analyze 6 items that INTEGRATED cannot, while INTEGRATED analyzes 22 items that BASELINE cannot. Neither wins at syntactic analysis.[19]

7.2 Ambiguity

INTEGRATED produces noticeably more trees per sentence than either the oracle grammar or the baseline grammar, on average (see 'readings' column in Table 4). The baseline's low ambiguity is not striking, because it only parses a few syntactically and morphologically simple sentences. Still, on some items the baseline grammar produces hundreds of trees. The oracle grammar clearly has less ambiguity. The main reason for more ambiguity in INTEGRATED is simple combinatorics. We infer noun and verb inflectional and position classes, and often end up with homophonous affixes as well as homophonous transitive and intransitive entries for verbs. For example, the verb *yuŋ*, meaning 'sit', 'be' or 'squat', is associated with both a transitive and an intransitive entry in INTEGRATED. Figure 4 illustrates two of the trees that this grammar finds for (3).

(3) cuwa mai-yuŋ-th-a-k-e
water NEG-be.there-NEG-PST-IPFV-IND.PST
'Was there water?"

These homophonous lexical resources combine with the argument optionality rules, word order flexibility and the actual complexity of Chintang morphotactics to create a large search space for the parser.

7.3 Future work

The default setting in MOM is to produce morphological rules which are optional. Furthermore, MOM does not yet infer non-inflecting lexical rules. This means that uninflected forms are passed through to the syntax without being associated with the morphosyntactic or morphosemantic information that the zero in the paradigm actually reflects. In future work, we will explore how to automatically posit such zero-marked rules, including how to make sure that their position classes are required, so that the grammar can properly differentiate 'zero' and 'uninflected'.

We plan to extend our syntactic inference algorithm to account for verbs with alternate or 'quirky' case frames. Another avenue that our error analysis shows as particularly promising is to handle complex clauses, as there are tools to model coordination (Drellishak and Bender, 2005) and subordination (Howell and Zamaraeva, 2018; Zamaraeva et al., 2019) in the Grammar Matrix framework.

8 Conclusion

We have demonstrated the value of integrating morphological and syntactic inference for automatic grammar development. Although inferring these properties is most easily handled separately, we show that combining information about morphotactic and inflectional patterns with syntactic properties such as transitivity improves coverage. While this study looked at case and transitivity, the benefits of creating lexical classes that encode syntactic information alongside morphological information should generalize. This methodology can be extended to other linguistic phenomena on the morphosyntactic interface, such as agreement, and the coverage of grammars can be further extended by expanding the lexical classes and clause types that can be inferred from the syntax. In the future, we would like to perform further, in-depth studies in collaboration with documentary linguists, for example to see if our system can help refine the analysis of morphological classes in the lexicon of the language in question and whether a grammar fragment automatically produced this way can be easily extended to broader coverage.

[19] A few of these differences are due to the fact that BASELINE includes pronouns and other word categories (such as interjections) as 'nouns' in the lexicon. We built INTEGRATED assuming only things marked as nouns go in. In future work, we will include pronouns (but not interjections).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. BCS-1561833. 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.

We would like to thank Alex Burrell for providing code from a previous project that we built on in order to infer transitivity.

We thank previous AGGREGATION research assistants for converting the corpus into Xigt and enriching it with INTENT.

References

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, 2002.

Emily M Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation*, 8:1–50. ISSN 1570-7075.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2011. Grammar engineering and linguistic hypothesis testing: Computational support for complexity in syntactic analysis. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage and Processing*, pages 5–29. CSLI Publications, Stanford, CA.

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, Robert Schikowski, and Balthasar Bickel. 2012b. Deriving a lexicon for a precision grammar from language documentation resources: A case study of Chintang. *Proceedings of COLING 2012*, pages 247–262.

Emily M Bender, Michael Wayne Goodman, Joshua Crowgey, and Fei Xia. August 2013. Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 74–83, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-2710.

Emily M Bender, Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. June 2014. Learning grammar specifications from IGT: A case study of Chintang. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 43–53, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-2206.

Balthasar Bickel, Manoj Rai, Netra Paudyal, Goma Banjade, Toya Nath Bhatta, Martin Gaenszle, Elena Lieven, Iccha Purna Rai, Novel K Rai, and Sabine Stoll. 2010. The syntax of three-argument verbs in Chintang and Belhare (Southeastern Kiranti). *Studies in ditransitive constructions: a comparative handbook*, pages 382–408.

Balthasar Bickel, Martin Gaenszle, Novel Kishore Rai, Vishnu Singh Rai, Elena Lieven, Sabine Stoll, G. Banjade, T. N. Bhatta, N Paudyal, J Pettigrew, and M Rai, I. P.and Rai. 2013. Tale of a poor guy. URL https://corpus1.mpi.nl/qfs1/media-archive/dobes_data/ChintangPuma/Chintang/Narratives/Annotations/phengniba_tale.tbt. Accessed: 22 October 2018.

Ann Copestake. 2002. *Implementing typed feature structure grammars*. CSLI publications, Stanford.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Scott Drellishak and Emily Bender. 2005. A coordination module for a crosslinguistic grammar resource. In *International Conference on Head-Driven Phrase Structure Grammar*, volume 12, pages 108–128. URL http://web.stanford.edu/group/cslipublications/cslipublications/HPSG/2005/drellishak-bender.pdf.

Dan Flickinger, Emily M Bender, and Stephan Oepen. 2014. ERG semantic documentation. URL http://www.delph-in.net/esd. Accessed on 2018-10-22.

Daniel P. Flickinger and Stephan Oepen. 1998. Towards systematic grammar profiling. Test suite technology ten years after. In *Beiträge zur 6. Fachtagung der Sektion Computerlinguistik der DGfS*, Heidelberg, 1998.

Antske Sibelle Fokkens. 2014. *Enhancing Empirical Research for Linguistically Motivated Precision Grammars*. PhD thesis, Department of Computational Linguistics, Universität des Saarlandes.

Ryan Georgi. 2016. *From Aari to Zulu: Massively Multilingual Creation of Language Tools Using Interlinear Glossed Text*. PhD thesis, University of Washington.

Michael Wayne Goodman, Joshua Crowgey, Fei Xia, and Emily M Bender. 2015. Xigt: Extensible interlinear gloss text for natural language processing. *Language Resources and Evaluation*, 49 (2): 455–485.

Lars Hellan and Dorothee Beermann. 2014. Inducing grammars from IGT. In Mariani J. Vetulani Z., editor, *Human Language Technology Challenges for Computer Science and Linguistics.*, volume 8287 of *LTC 2011. Lecture Notes in Computer Science*. Springer.

Kristen Howell and Olga Zamaraeva. 2018. Clausal modifiers in the Grammar Matrix. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2939–2952.

Kristen Howell, Emily M Bender, Michael Lockwood, Fei Xia, and Olga Zamaraeva. 2017. Inferring case systems from IGT: Impacts and detection of variable glossing practices. *ComputEL-2*, pages 67–75.

Stefan Müller. 1999. *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche*. Number 394 in Linguistische Arbeiten. Max Niemeyer Verlag, Tübingen.

Stefan Müller. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling*, 3(1): 21–86. URL https://hpsg.hu-berlin.de/~stefan/Pub/coregram.html.

Stefan Müller. 2016. *Grammatical theory: From transformational grammar to constraint-based approaches*. Language Science Press.

Robert Schikowski. Chintang morphology. Unpublished ms, University of Zürich, 2012.

Robert Schikowski, NP Paudyal, and Balthasar Bickel. 2015. Flexible valency in Chintang. *Valency Classes: a Comparative Handbook*.

Sabine Stoll and Balthasar Bickel. 2012. How to measure frequency? Different ways of counting ergatives in Chintang (Tibeto-Burman, Nepal) and their implications. In Frank Seifart, Geoffrey Haig, Nikolaus P. Himmelmann, Dagmar Jung, Anna Margetts, and Paul Trilsbeek, editors, *Potentials of Language Documentation: Methods, Analyses, and Utilization*, pages 83–89. University of Hawai'i Press.

David Wax. 2014. Automated grammar engineering for verbal morphology. Master's thesis, University of Washington.

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, 2007.

Olga Zamaraeva. 2016. Inferring morphotactics from interlinear glossed text: combining clustering and precision grammars. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 141–150.

Olga Zamaraeva, František Kratochvíl, Emily M Bender, Fei Xia, and Kristen Howell. 2017. Computational support for finding word classes: A case study of Abui. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 130–140.

Olga Zamaraeva, Kristen Howell, and Emily M. Bender. 2019. Modeling clausal complementation for a grammar engineering resource. In *Proceedings of the Society for Computation in Linguistics*, volume 2, page Article 6. URL https://scholarworks.umass.edu/scil/vol2/iss1/6/.