A Typologically-Motivated Approach to Automatic Morphotactic Inference

Isaac Schifferer
University of Washington
isaacjs2@uw.edu

1 Introduction

One of the perennial challenges in the domain of automatic grammar inference is to produce grammars that not only make correct predictions about language data, but also whose structures are intelligible to linguists. In this regard, HPSG grammar fragments inferred using the AGGREGATION system (Bender et al., 2013, 2014) often fall short, particularly when it comes to describing languages' morphotactic systems. This work explores whether and how concepts from morphological typology can inform approaches to morphotactic inference in order to improve the quality of resulting grammars. To that end, I propose a set of changes to the AGGREGATION system's morphotactic inference process that leverage grammatical feature information to (1) steer the inference algorithm towards positing canonical morphotactic systems and (2) anticipate instances of common typological patterns related to flexivity and cumulative exponence. I evaluate these changes using datasets representing six morphologically diverse languages, which are additionally each from a different language family and geographical location. The results show that this approach can improve the quality of the resulting morphotactic analyses without impacting parsing performance.

2 Background

2.1 The LinGO Grammar Matrix

The LinGO Grammar Matrix (the Matrix; Bender et al., 2002, 2010; Zamaraeva et al., 2022) is a system that allows users to interactively develop machine-readable HPSG (Pollard and Sag, 1994) grammars. Tools like the LKB (Copestake, 2002) can be used to compile these grammars which in turn can be used to parse or generate sentences. The Matrix's core grammar and collection of modular libraries contain a wealth of generalized implementations of various aspects of grammar that enable the rapid development of precision grammars. The Matrix's implementation of morphology (O'Hara, 2008; Goodman, 2013) is primarily concerned with morphotactics, which has to do with the ordering of and co-occurrence restrictions on morphemes. This is largely due to the fact that the Matrix assumes a morphophonological analyzer that "standardizes" the data by doing things like mapping any kind of non-concatenative morphology (e.g. infixes) to a strictly concatenative representation (Bender et al., 2010; Bender and Good, 2005). In a Matrix grammar, the lexicon is modeled as a set of lexical classes (LCs) and lexical rules are modeled as a set of position classes (PCs). Each LC is composed of one or more stems with corresponding predicate values. Each PC has a set of inputs and is composed of a set of lexical rule types (LRTs). Each input is either an LC or another PC. An LRT contains one or more affixes and has some number of feature-value pairs associated with it, e.g. first person and plural number. All together, the morphotactic system of any given part of speech can be conceptualized as a directed, acyclic graph with the LCs and PCs as the nodes and the inputs of each PC forming edges from the input LCs/PCs to the PC. In this "input graph," words are formed by starting at some LC node and moving through the graph, picking up a stem in the LC node and an affix at each subsequent PC node, stopping at any point that satisfies all of the constraints (i.e. obligatory PCs and co-occurrence restrictions). The set of stem and affix combinations that can be created in this way constitute the set of valid words in the grammar.

2.2 AGGREGATION

The AGGREGATION project (Bender et al., 2013, 2014) centers on approaches for automatically inferring Matrix grammars based on the analysis of information stored in interlinear glossed text (IGT) data. To that end, Howell (2020) established an end-to-end inference pipeline that ingests the IGT data, uses it to infer a grammar, compiles that grammar, and evaluates its performance by attempting parse a set of held-out test sentences. A key component of the pipeline is the MOM morphotactic inference module (Wax, 2014; Zamaraeva, 2016; Zamaraeva et al., 2017, 2019), which infers the lexicon and morphotactics for nouns and verbs.[1] At a high level, the inference process in MOM consists of initializing an LC for each stem and a PC for each affix in the data and iteratively merging the pair of objects[2] with the highest overlap score until no pair has an overlap score above a certain threshold. The overlap score is defined as the proportion of the objects' collective inputs (outputs for LCs) that are shared. For example, if *verb-pc1* has inputs {*verb1*, *verb2*} and *verb-pc2* has inputs {*verb1*, *verb-pc3*}, their overlap is $\frac{|\{verb1\}|}{|\{verb1,verb2,verb\text{-}pc3\}|} = 0.333$.

2.3 Morphology

The proposed changes in this work draw on several ideas from morphological typology, the first being the concept of *canonical morphotactics*. Bonami and Crysmann

[1] Inference occurs separately for nouns and verbs, but the algorithm is the same for both parts of speech.

[2] PCs only merge with PCs and LCs only merge with LCs.

(2013) describe the canonical ideal of morphotactics as a situation where all morphs are organized into a sequence of position classes, where a position class is a collection of "morphs expressing different values for the same feature cluster in a single linear position, strictly ordered with respect to positions serving for the realisation of other features." (pg. 28). In other words, in a perfectly canonical morphotactic system, the realization of all words (in a given part of speech) can be described as a fixed sequence of affixations where at each step the affix chosen expresses a unique feature value for a specific feature that is not expressed in any other position. Additionally, no surface form appears in more than one position class.

I also consider two parameters of inflectional morphology as discussed in Bickel and Nichols, 2007, namely flexivity and exponence. Importantly, Bickel and Nichols argue that the values of these parameters cannot be broadly applied to languages, due to the fact that most (if not all) inflectional systems have some variation with respect to each parameter. Instead, they focus their discussion on individual *formatives*. A formative is *flexive* if it displays lexical allomorphy, meaning it has several possible surface forms and its realization is determined by the lexical item it is attached to (pg. 184). *Exponence* deals with how many grammatical features a formative expresses (pg. 188). With respect to this parameter, I am most interested in the sets of features that have been documented as being realized cumulatively across languages (though the following is not a comprehensive list). It is common for different combinations of person, number, and gender to be expressed cumulatively (Kibort and Corbett, 2008). Within that group, person and number are most commonly combined (Bickel and Nichols, 2007, pg. 228; Plank, 1999, pg. 292). Relatedly, number and case are another pair of features that have been observed to be cumulated, especially in Indo-European languages (Bickel and Nichols, 2007, pg. 188; Bickel and Nichols, 2013; Plank, 1999, pg. 302). Tense, aspect, and mood, or subsets thereof, are regularly realized together, in part because it is often difficult to make a clear distinction between the three semantically (Kibort, 2008; Östen Dahl and Velupillai, 2013).

3 Methodology

As discussed above, MOM groups similar affixes into position classes (PCs) by iteratively merging PCs, and this process is dictated by pairwise PC overlap scores. In this section, I describe a new implementation of the overlap score function, defined in Algorithm 1, that utilizes the grammatical feature information expressed by affixes to (when possible) model the morphological patterns introduced in the previous section. [3]

First, in order to reorient the inference process towards positing canonical position classes, the overlap function is primarily based on feature overlap as opposed to input overlap. This value is calculated in line 16 of the algorithm, where $pc.feats$ is the set of all features (not feature values) expressed by LRTs in a PC. Because the pair of PCs with the highest overlap score at each step is merged and that score only needs to be above the specified threshold (the default is 0.2), this approach does not strictly enforce canonical PCs, but rather it prioritizes "more canonical" merges. This helps to account for the reality that languages often subvert the canonical ideal in one way or another. [4]

To attempt to account for instances of flexivity, I retain the input overlap component from the original overlap function and use it as a prerequisite for having an overall overlap score greater than 0 (line 17). The intuition behind this is that if there are a pair of affixes (or two sets of affixes) that express the same feature(s) but there is no stem in the data that uses both affixes (or draws from both sets of affixes) on separate occasions, then there may be a legitimate co-occurrence restriction. Additionally, because this input overlap prerequisite does not explicitly target LC inputs, it could potentially help to prevent the grammar from being overly permissive in non-lexical cases of allomorphy as well.

The basic definition of feature overlap already accounts for cumulative exponence. However, I also added logic to further prioritize the merging of PCs whose affixes express multiple features. The subroutine BOOST (lines 1-6) is called several times in the main overlap function (lines 11-15) and takes as arguments the feature sets from the two PCs and an additional set of features.[5] If both PC feature sets share at least one feature with the extra feature set and one of the PC feature sets shares at least two, then all of the features from the extra feature set will be added to the PC feature sets, inflating the feature overlap for the pair of PCs and effectively prioritizing their merging. Only one PC is required to have more than one feature from a feature group to account for cases where some feature value is not explicitly marked and correspondingly not glossed, e.g. number and case are expressed cumulatively but singular number is not glossed in the data because it is analyzed as the default.

In addition to the changes that were motivated by specific concepts from the literature, I made a handful of practical considerations. Inside of the overlap function, if one or both PCs have no affixes that express features, the overlap score reverts to being based on shared inputs

[3] In this work, I only modify the PC overlap function. The (up to now) parallel LC overlap function remains based on output overlap because MOM assumes that stems do not express any features (any features that do show up on a stem are analyzed as being realized via zero-inflection).

[4] Additionally, this approach does not enforce the 'single feature per PC' aspect of canonical morphotactics or deprioritize affixes expressing more than one feature because, assuming the input data is accurate, any instances of affixes with multiple features are legitimate.

[5] In the algorithm definition, 'P', 'N', and 'G' stand for 'person', 'number', and 'gender', respectively, and 'T', 'A', and 'M' stand for 'tense', 'aspect', and 'mood', respectively.

(lines 9-10). There are several non-linguistic reasons why an affix might not have any corresponding features at this point in the process (e.g. missing or unrecognized glosses in the dataset), so without this cutout, grammars often end up with many single-affix PCs.[6] Apart from the changes to the overlap function, I also increased the default number of merging rounds in the pipeline's configuration. For most datasets, this has the effect of increasing the amount of merging overall, therefore decreasing the total number of LCs and PCs. Finally, I implemented a simple system for directly mapping dataset glosses onto MOM's internal feature values via an additional configuration file. This helps to decrease the amount of feature information lost from a dataset in MOM's internal representation of the data.

Algorithm 1 Feature-based overlap function

```
1:  function BOOST(feats1, feats2, feat_set)
2:      if |feats1 ∩ feat_set| ≥ 1 ∧ |feats2 ∩ feat_set| ≥ 1 then
3:          if |feats1 ∩ feat_set| + |feats2 ∩ feat_set| > 2 then
4:              feats1 ← feats1 ∪ feat_set
5:              feats2 ← feats2 ∪ feat_set
6:      return feats1, feats2
7:  function OVERLAP(pc1, pc2)
8:      input_overlap ← |pc1.inputs ∩ pc2.inputs|/|pc1.inputs ∪ pc2.inputs|
9:      if |pc1.feats| = 0 ∨ |pc2.feats| = 0 then
10:         return input_overlap
11:     if pc1.pos = 'verb' then
12:         pc1.feats, pc2.feats ← BOOST(pc1.feats, pc2.feats, {'P', 'N', 'G'})
13:         pc1.feats, pc2.feats ← BOOST(pc1.feats, pc2.feats, {'T', 'A', 'M'})
14:     else
15:         pc1.feats, pc2.feats ← BOOST(pc1.feats, pc2.feats, {'N', 'case'})
16:     feat_overlap ← |pc1.feats ∩ pc2.feats|/|pc1.feats ∪ pc2.feats|
17:     if input_overlap > 0 then
18:         return feat_overlap
19:     else
20:         return 0
```

4 Evaluation Methodology

I evaluated my changes by running the end-to-end AGGREGATION pipeline with six different datasets. The datasets were chosen from those available to maximize morphological, genealogical, and geographical diversity. I used a few metrics (originally calculated by Liu (2025)) to approximate morphological diversity, namely the average number of morphemes per word, the average number of unique word forms per stem, and the average number of surface forms per feature value. The six languages represented by the datasets are Wambaya [wmb] (Nordlinger, 1998), a Mirndi language of northern Australia, Lezgi [lez] (Clifton et al., 2024), a Nakh-Daghestanian language of the North Caucasus, Manipuri [mni] (Chelliah, 2019), a Sino-Tibetan language of northeast India, Wakhi [wbl] (Obrtelová, 2019), an Indo-European language of northeast Afghanistan, South Efate [erk] (Thieberger, 2006), an Austronesian language of Vanuatu, and Hiaki [yaq] (Harley, 2019), an Uzo-Aztecan language of northwest Mexico. The Wambaya, Lezgi, and Manipuri datasets were used during the development process, while the Wakhi, South Efate, and Hiaki datasets were held out as test datasets. For each dataset, I employed 10-fold cross-validation and report the average of each evaluation metric.

I analyzed the inferred grammars using a collection of grammar performance and complexity metrics, as well as by examining the noun and verb input graphs of one grammar for each dataset. To quantify grammar performance, I looked at *lexical coverage* (the proportion of sentences for which a grammar is able to produce an analysis for each word), *parse coverage* (the proportion of sentences for which a grammar yields at least one complete analysis), *ambiguity* (the average number of parses per sentence for the sentences with at least one parse), and *maximum results* (the number of parses for the single sentence in the evaluation set with the most parses). To quantify grammar complexity, I looked at the *number of LCs* and the *number of PCs* for both nouns and verbs. I manually inspected the input graphs using the MOM visualization tool[7] (Lepp et al., 2019), which generates interactive visualizations of a grammar's noun and verb input graphs.

5 Results

The grammar performance metrics for the baseline (first line) and final (second line) runs are presented in Table 1. Overall, there was not much change in the amount of coverage between the baseline and final values, but there were some improvements in the ambiguity metrics, namely for the Lezgi, Manipuri, and Hiaki datasets.

The grammar complexity metrics for the baseline and final runs are presented in Table 2. In most cases, the average number of LCs and PCs stayed the same or decreased by some amount. The most significant decreases came in the number of LCs, namely for Lezgi nouns and verbs, Manipuri verbs, and Hiaki verbs. Based on additional experiments run with subsets of the changes, these drops are mainly the result of increasing the number of merging rounds. While grammars with fewer LCs/PCs

[6] It is also important to note that this lack or loss of information almost always stems from the fact that the datasets used for inference were not created specifically for this task and it is challenging to perfectly convert them into the necessary format, and not from "bad data".

[7] http://uakari2.ling.washington.edu/aggregation/mom-gui/

are not inherently better, these results are desired because in general, MOM greatly under-combines LCs and PCs. For example, the inferred Chintang grammars in Zamaraeva's (2016) case study had at the fewest 48 PCs, while the hypothesized number of PCs from the literature was only 13.

As a result of the feature-based overlap function, the PCs of the final grammars were much more organized by feature than those of the baseline grammars. In most cases, the baseline grammars had multiple PCs whose affixes had non- or partially-overlapping feature sets, while the final grammars had no or very few such PCs. Consequently, the final noun and verb input graphs were often easier to interpret and appeared to represent more plausible hypotheses of the languages' morphotactic systems (though further investigation would be required to verify this with respect to each individual language). For example, in the final Wakhi verb input graph, shown in Figure 1[8], there is only one PC (*verb-pc1*) that contains affixes with non-overlapping features (its affixes have tense, person and number, or all three) and there is a reasonable set of connections between PCs. However, there were still several input graphs that remained large, unorganized, and difficult to interpret. In these input graphs, most of the PCs had many inputs and outputs, and while the affixes within each PC generally shared the same feature(s) (unlike in the corresponding baselines) the graphs still contained many long paths that pass through multiple PCs with the same feature(s). For example, in the final Lezgi noun input graph, shown in Figure 2, there are several PCs whose affixes express number (top left cluster) and several that express case (top right cluster). These outcomes were mainly due to the large number of featureless affixes in each dataset. Essentially, my implementation of the overlap function does not sufficiently prioritize instances of actual feature overlap over instances where the input overlap fallback is used, resulting in too many lower quality (in my estimation) merges early on in the process that prevent later, more desired merges.[9]

The results and patterns seen in the evaluation of the development datasets carried over to the test datasets. This is expected because the changes I made were motivated by the typological literature and/or known characteristics of the data used for grammar inference broadly, and they generally did not consider specific patterns observed in the development datasets, for better or for worse.

6 Conclusion

The results, including the decrease of ambiguity in some cases, the overall decrease in the amount of LCs and PCs, and most importantly the improvements to the quality of the input graphs, show that these changes were successful in reducing the complexity of inferred grammars without negatively impacting grammars' parsing abilities. In turn, assuming that the inflectional systems being modeled are generally canonical, the resulting grammars are indeed closer to something a linguist would create (of course, the individual grammars need to be compared to the established knowledge on each language in order to fully verify this). This suggests that typologically-informed approaches to inference can in fact result in higher quality grammars, even when the overall methodology is heuristic in nature, as is the case with MOM's overlap-based approach to morphotactic inference. Future work can improve this process by investigating the quality of output grammars with respect to linguist-produced analyses of specific languages, accounting for additional non-canonical phenomena, and continuing to close the gap between the feature information in IGT glosses and the information in MOM's internal representation of the data.

Dataset	Lexical Coverage	Coverage	Ambiguity	Max Results
Wambaya	8.19 %	1.83 %	26.33	356
Wambaya	8.44 %	1.83 %	26.33	356
Lezgi	11.61 %	9.36 %	3,315.30	28,110
Lezgi	11.11 %	9.11 %	2,087.67	23,968
Manipuri	6.99 %	6.17 %	3,340.79	32,592
Manipuri	7.05 %	6.23 %	1,697.07	23,711
Wakhi	25.77 %	13.47 %	18.21	586
Wakhi	25.18 %	14.79 %	18.68	586
South Efate	14.87 %	7.01 %	5,628.68	20,013
South Efate	14.89 %	6.93 %	5,928.53	21,786
Hiaki	16.47 %	9.71 %	131.47	11,904
Hiaki	16.55 %	9.35 %	30.33	1,125

Table 1: Baseline and Final Grammar Performance Metrics

Dataset	N LCs	N PCs	V LCs	V PCs
Wambaya	24.1	20.3	19.3	24.1
Wambaya	23.8	23.1	19.3	24.1
Lezgi	61.2	27.2	41.1	27.0
Lezgi	53.5	26.2	29.1	25.1
Manipuri	20.8	23.7	33.2	32.9
Manipuri	17.7	23.0	25.5	34.1
Wakhi	12.0	6.9	8.6	9.1
Wakhi	11.9	6.9	6.6	9.7
South Efate	34.5	11.9	13.8	11.1
South Efate	20.7	9.0	11.8	9.8
Hiaki	23.7	11.7	34.0	24.7
Hiaki	19.8	11.5	19.5	20.0

Table 2: Baseline and Final Grammar Complexity Metrics

[8] In the graph, the purple/circle nodes are LCs, the green/triangle nodes are suffix PCs, and the blue/triangle node is a prefix PC. Node size indicates the relative number of stems/affixes in the LC/PC.

[9] Since input graphs cannot contain cycles, one merge can prevent another by causing one PC to become the input of another PC that it otherwise has a high overlap with, for example.

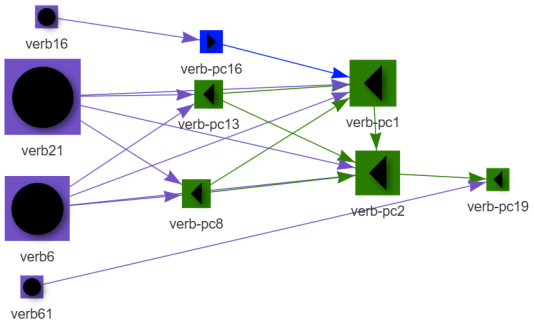

Figure 1: Final Wakhi verb input graph

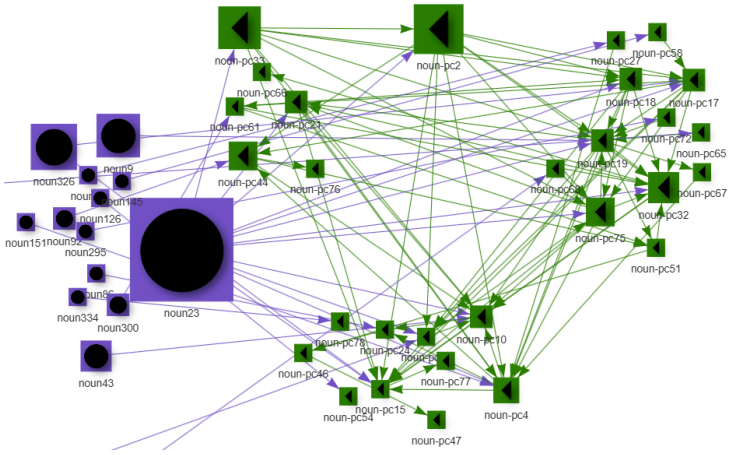

Figure 2: Final Lezgi noun input graph, visually sorted by feature

References

Emily M. Bender, Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. 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. Association for Computational Linguistics.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar Customization. 8(1):23–72.

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 *COLING-02: Grammar Engineering and Evaluation*.

Emily M. Bender and Jeff Good. 2005. Implementation for Discovery: A Bipartite Lexicon to Support Morphological and Syntactic Analysis. *Papers from the Regional Meeting of the Chicago Linguistic Society*, 41(2):1–16.

Emily M. Bender, Michael Wayne Goodman, Joshua Crowgey, and Fei Xia. 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. Association for Computational Linguistics.

Balthasar Bickel and Johanna Nichols. 2007. Inflectional Morphology. In Timothy Shopen, editor, *Language Typology and Syntactic Description: Volume 3: Grammatical Categories and the Lexicon*, 2 edition, volume 3, pages 169–240. Cambridge University Press.

Balthasar Bickel and Johanna Nichols. 2013. Exponence of Selected Inflectional Formatives (v2020.4). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Olivier Bonami and Berthold Crysmann. 2013. Morphotactics in an information-based model of realisational morphology. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, pages 27–47.

Shobhana Lakshmi Chelliah. 2019. Meithei texts. Manipur Digital Resources in UNT Digital Library. University of North Texas Libraries. (Accessed August 2019).

John M. Clifton, Charles M. Donet, and Jessica Smith. 2024. Lezgi Texts from Azerbaijan. Endangered Languages Archive. Handle: `http://hdl.handle.net/2196/fbbf9e7c-2074-4ef1-88e1-6804b5b2ec92` [Accessed: January 2026].

A. Copestake. 2002. *Implementing Typed Feature Structure Grammars*. Stanford: CSLI.

Michael Wayne Goodman. 2013. Generation of Machine-Readable Morphological Rules from Human-Readable Input.

Heidi Harley. 2019. Hiaki text corpus. University of Arizona. Unpublished FieldWorks (FLEx) project. (Accessed August 2019).

Kristen Howell. 2020. Inferring Grammars from Interlinear Glossed Text: Extracting Typological and Lexical Properties for the Automatic Generation of HPSG Grammars.

Anna Kibort. 2008. Grammatical Features Inventory: Tense. University of Surrey.

Anna Kibort and Greville G. Corbett. 2008. Grammatical Features Inventory: Gender. University of Surrey.

Haley Lepp, Olga Zamaraeva, and Emily M. Bender. 2019. Visualizing Inferred Morphotactic Systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 127–131, Minneapolis, Minnesota. Association for Computational Linguistics.

Tongxi Liu. 2025. The Interplay of Dataset Characteristics in Automated Grammar Generation: A Study with the AGGREGATION System. Master's thesis, University of Washington.

Rachel Nordlinger. 1998. *A Grammar of Wambaya, Northern Australia*. Pacific linguistics, Series C, 140. Pacific Linguistics.

Jaroslava Obrtelová. 2019. *From Oral to Written: A Text-linguistic Study of Wakhi Narratives*. Ph.D. thesis, Uppsala.

Kelly O'Hara. 2008. A Morphotactic Infrastructure for a Grammar Customization System.

Frans Plank. 1999. Split morphology: How agglutination and flexion mix. 3(3):279–340.

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

Nicholas Thieberger. 2006. *A Grammar of South Efate*. University of Hawai'i Press.

David Wax. 2014. Automated Grammar Engineering for Verbal Morphology. Master's thesis, University of Washington.

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, Berlin, Germany. Association for Computational Linguistics.

Olga Zamaraeva, Chris Curtis, Guy Emerson, Antske Fokkens, Michael Goodman, Kristen Howell, T. J. Trimble, and Emily M. Bender. 2022. 20 years of the Grammar Matrix: cross-linguistic hypothesis testing of increasingly complex interactions. 10(1):49–137.

Olga Zamaraeva, Kristen Howell, and Emily M. Bender. 2019. Handling cross-cutting properties in automatic inference of lexical classes: A case study of Chintang. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 28–38, Honolulu. Association for Computational Linguistics.

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, Honolulu. Association for Computational Linguistics.

Östen Dahl and Viveka Velupillai. 2013. Tense and Aspect (v2020.4). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.