A Typologically-Motivated Approach to Automatic Morphotactic Inference

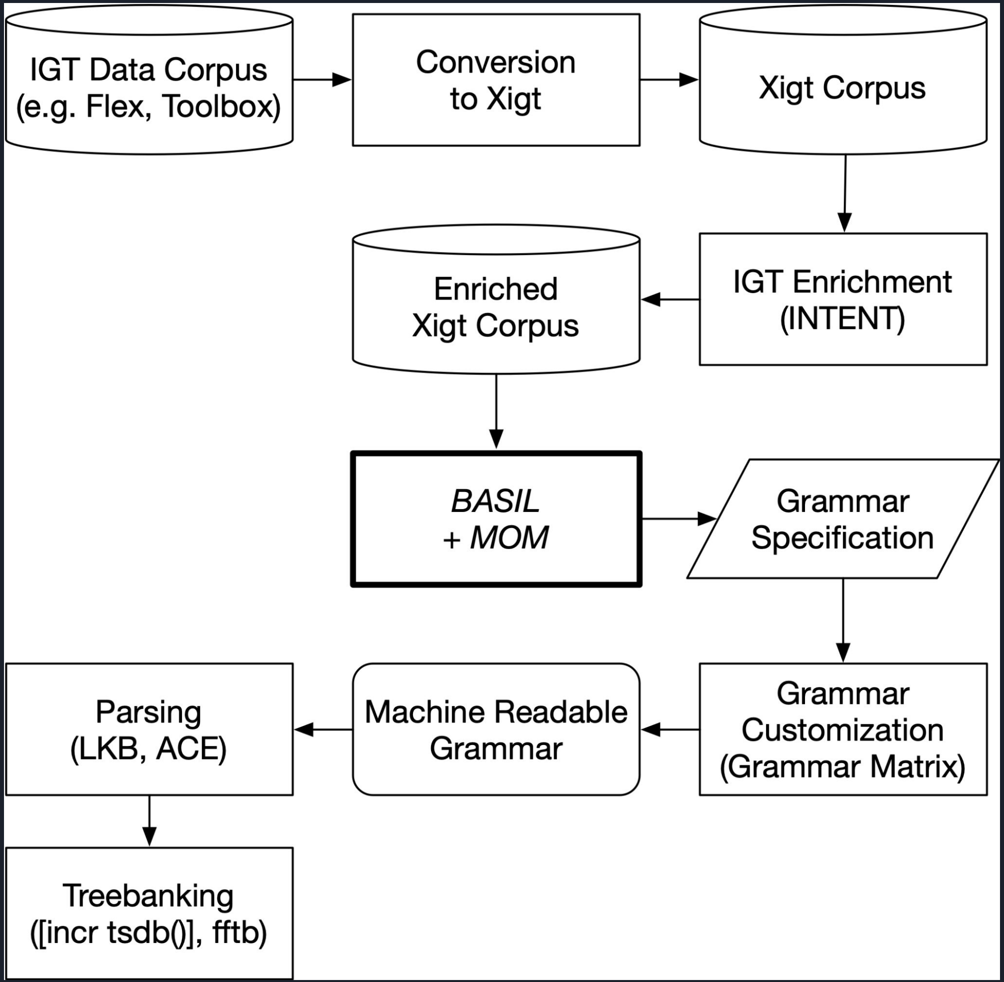
Isaac Schifferer
University of Washington

The Grammar Matrix

- Tool for creating implemented grammars
- Core grammar + modular libraries for individual phenomena (e.g. valence-changing morphology, noun incorporation)
- HPSG + MRS
- Interactive web app → choices file

AGGREGATION

- Automatic Generation of Grammars for Endangered Languages from Glosses and Typological Information
- Rule-based grammar inference system
- (Language Documentation →) Interlinear Glossed Text (IGT) → choices file
- IGT is a rich source of information, but only because of the hard work of linguists and community members


IGT Data Corpus
(e.g. Flex, Toolbox)
Conversion
to Xigt
Xigt Corpus
IGT Enrichment
(INTENT)
Enriched
Xigt Corpus
BASIL
+ MOM
Grammar
Specification
Grammar
Customization
(Grammar Matrix)
Machine Readable
Grammar
Parsing
(LKB, ACE)
Treebanking
([incr tsdb()], fftb)

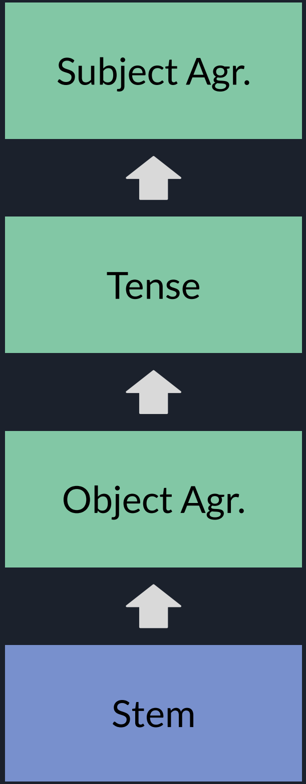
Howell, 2020

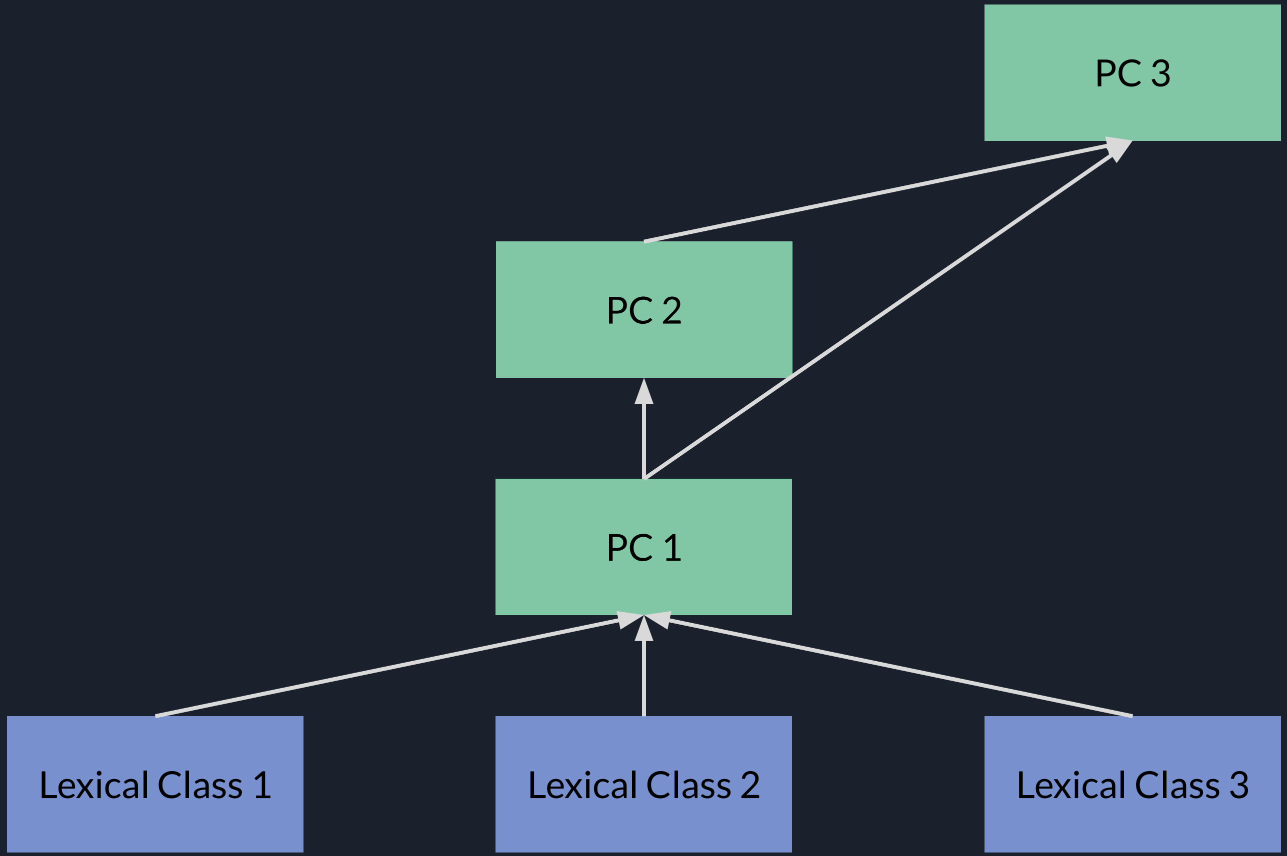
Morphotactics in the Matrix

Swahili:

tu-	li-	ku-	lipa
1PL-	PST-	2SG-	pay
[Subj. Agr.]	[Tense]	[Obj. Agr.]	Stem

Adapted from Stump (1993).




PC 3
PC 2
PC 1
Lexical Class 1
Lexical Class 2
Lexical Class 3

```
verb1_name=verb1
  verb1_feat1_name=case
  verb1_feat1_value=acc
  verb1_feat1_head=obj
verb1_valence=trans
  verb1_stem1_orth=aliyulu
  verb1_stem1_pred=_find_v_rel
  verb1_stem2_orth=didima
  verb1_stem2_pred=_tell_v_rel
```

```
verb-pc1_name=verb-pc1
verb-pc1_order=suffix
verb-pc1_inputs=verb1, verb-pc6, verb15
  verb-pc1_lrt1_name=verb-pc1_lrt1
    verb-pc1_lrt1_lri1_inflecting=yes
    verb-pc1_lrt1_lri1_orth=-j
  verb-pc1_lrt2_name=verb-pc1_lrt2
    verb-pc1_lrt2_feat1_name=tense
    verb-pc1_lrt2_feat1_value=nfut
    verb-pc1_lrt2_feat1_head=verb
    verb-pc1_lrt2_lri1_inflecting=yes
    verb-pc1_lrt2_lri1_orth=-lumi
```

LC and PC of a Wambaya grammar inferred from data from Nordlinger, 1998

MOM

- Infers the morphotactics of nouns and verbs
- Input: Glossed and morpheme-segmented nouns/verbs
- Output: Hypothesized set of lexical classes (LCs) and position classes (PCs)
- Assumes a fixed order of affixation (all prefixes, then all suffixes)

Basic process:

- For each word in the dataset, initialize an LC for the stem and a PC for each affix and add the appropriate inputs
- <u>Iteratively merge the pair of PCs with the highest amount of input overlap</u>
- Stop when no pair of PCs has an overlap value greater than the configured threshold

Input Overlap

Algorithm 1 Input-based overlap function

function OVERLAP($pc1$, $pc2$)

$\quad input\_overlap \leftarrow |pc1.inputs \cap pc2.inputs| / |pc1.inputs \cup pc2.inputs|$

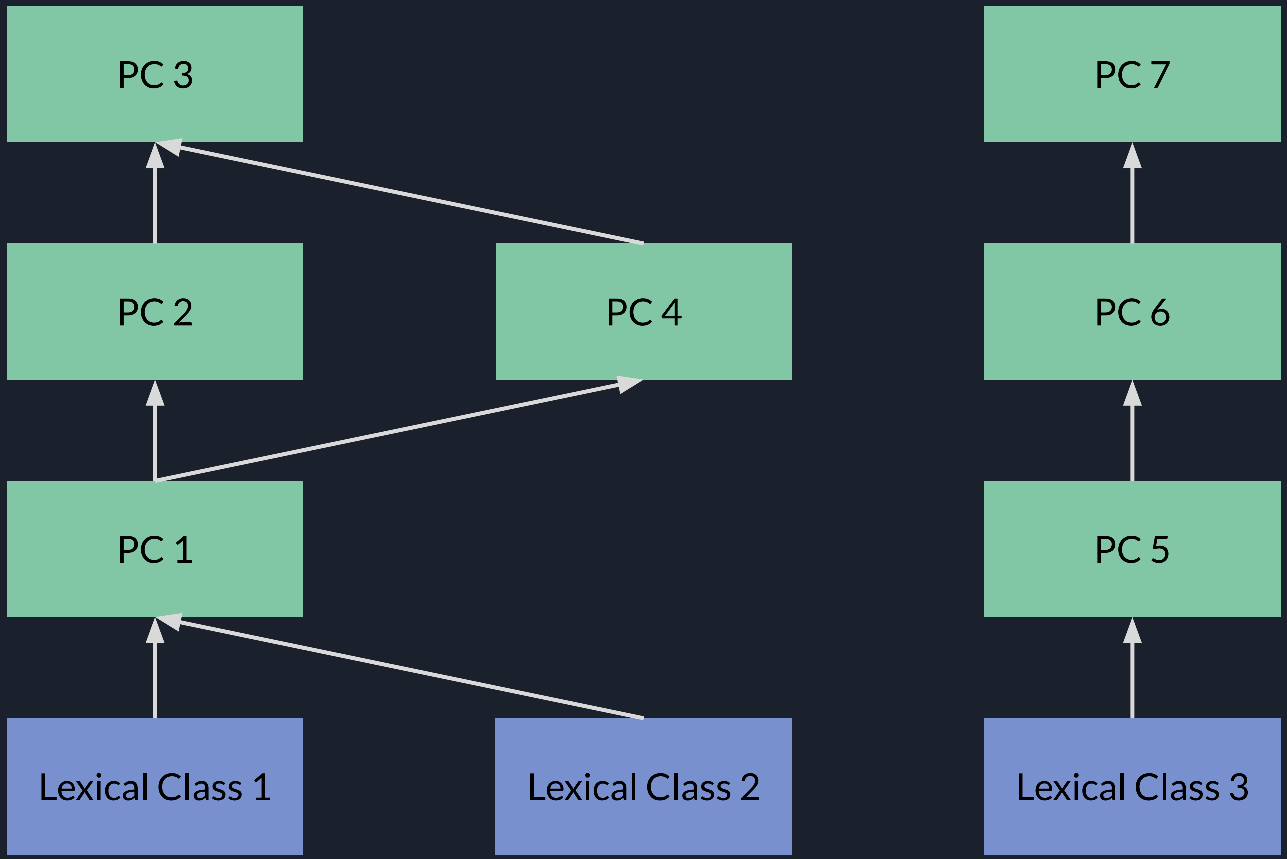
$\quad$ **return** $input\_overlap$

Example:

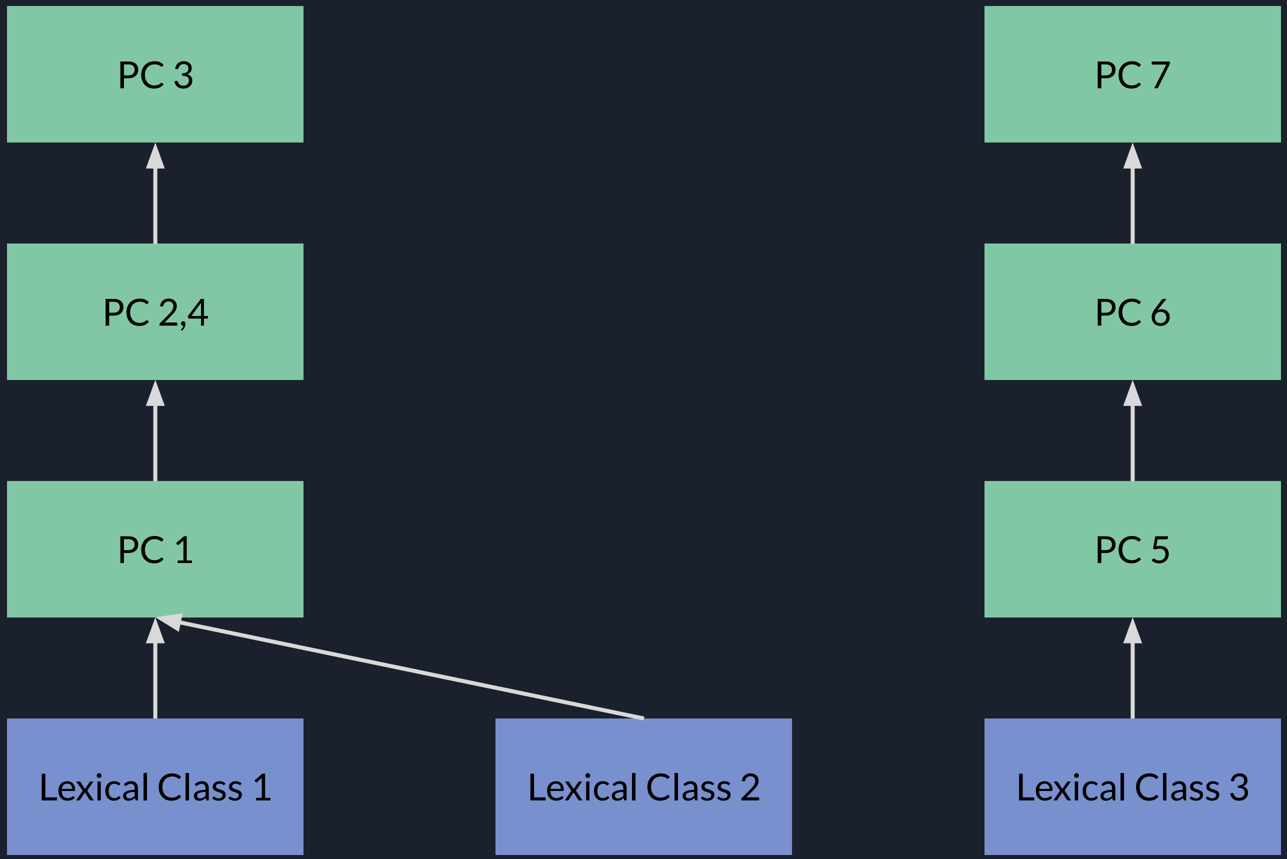
verb-pc1 inputs: {verb1, verb2, verb-pc2}

verb-pc3 inputs: {verb1, verb2, verb-pc4}

Overlap: |{verb1, verb2}| / |{verb1, verb2, verb-pc2, verb-pc4}| = 0.5



PC 3
PC 7
PC 2
PC 4
PC 6
PC 1
PC 5
Lexical Class 1
Lexical Class 2
Lexical Class 3


PC 3
PC 7
PC 2,4
PC 6
PC 1
PC 5
Lexical Class 1
Lexical Class 2
Lexical Class 3

Research Question

- How can concepts from morphological typology inform approaches to automatic morphotactic inference?
  - How can the grammatical feature information in IGT gloss lines be utilized in the inference process?
- Primary hypothesis: morphotactic inference can be improved by using grammatical feature information to (1) model canonical morphotactics and (2) anticipate instances of common typological patterns
  - Canonical PC: 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” (Bonami and Crysmann, 2013, pg. 28)
  - Typological parameters: flexivity, cumulative/fused exponence

Feature Overlap

$$feat\_overlap \leftarrow |pc1.feats \cap pc2.feats| / |pc1.feats \cup pc2.feats|$$

Example:

verb-pc1 features: {case}

verb-pc2 features: {case, number}

Overlap: |{case}| / |{case, number}| = 0.5

Methodology: new overlap function

- Feature overlap
- Input overlap prerequisite
- Feature cluster “boosting”

Practical considerations

- Input overlap fallback
- Configuration change: “unlimited” merging rounds (previously 1)

Algorithm 3 Feature-based overlap function with feature-specific considerations

```
function BOOST_OVERLAP(feats1, feats2, feat_set)
    if |feats1 ∩ feat_set| ≥ 1 ∧ |feats2 ∩ feat_set| ≥ 1 then
        if |feats1 ∩ feat_set| + |feats2 ∩ feat_set| ≥ 3 then
            feats1 ← feats1 ∪ feat_set
            feats2 ← feats2 ∪ feat_set
    return feats1, feats2
function OVERLAP(pc1, pc2)
    input_overlap ← |pc1.inputs ∩ pc2.inputs|/|pc1.inputs ∪ pc2.inputs|
    if |pc1.feats| = 0 ∨ |pc2.feats| = 0 then                    ▷ input fallback
        return input_overlap
    if pc1.pos_tag = ‘verb’ then                                 ▷ boosted feature groupings
        pc1.feats, pc2.feats ← BOOST_OVERLAP(pc1.feats, pc2.feats, {‘P’, ‘N’, ‘G’})
        pc1.feats, pc2.feats ← BOOST_OVERLAP(pc1.feats, pc2.feats, {‘T’, ‘A’, ‘M’})
    else
        pc1.feats, pc2.feats ← BOOST_OVERLAP(pc1.feats, pc2.feats, {‘N’, ‘case’})
    feat_overlap ← |pc1.feats ∩ pc2.feats|/|pc1.feats ∪ pc2.feats|
    if input_overlap > 0 then                                    ▷ input overlap prerequisite
        return feat_overlap
    else
        return 0
```

Evaluation Methodology

- Data-driven development: 3 development datasets, 3 test datasets
- Evaluation: 10-fold cross-validation
- Metrics
  - Grammar performance: lexical and sentence coverage, amount of ambiguity
  - Grammar complexity: # LCs, # PCs
- Manual inspection
- Results are compared to the baseline inferred grammars
  - Comparison to gold standard grammars or other existing analyses of specific languages is left to future work

Language [iso]	Number of IGT	Number of Words	Morphemes per Word	Type Stems Ratio	Allomorph Ratio
Lezgi [lez]	1,531	18,724	2.17	3.05	1.48
Manipuri [mni]	1,785	15,887	1.51	1.80	1.39
Wambaya [wmb]	797	2,933	1.54	1.86	1.35
Wakhi [wbl]	770	3,219	1.79	1.86	1.17
South Efate [erk]	2,240	24,774	1.4	2.55	1.38
Hiaki [yaq]	10,971	77,894	1.19	1.44	3.69

Table 3.1: Dataset characteristics of development and test datasets (Liu, 2025)

Results: Grammar Performance

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 5.13: Baseline and Final Grammar Performance Metrics by Dataset

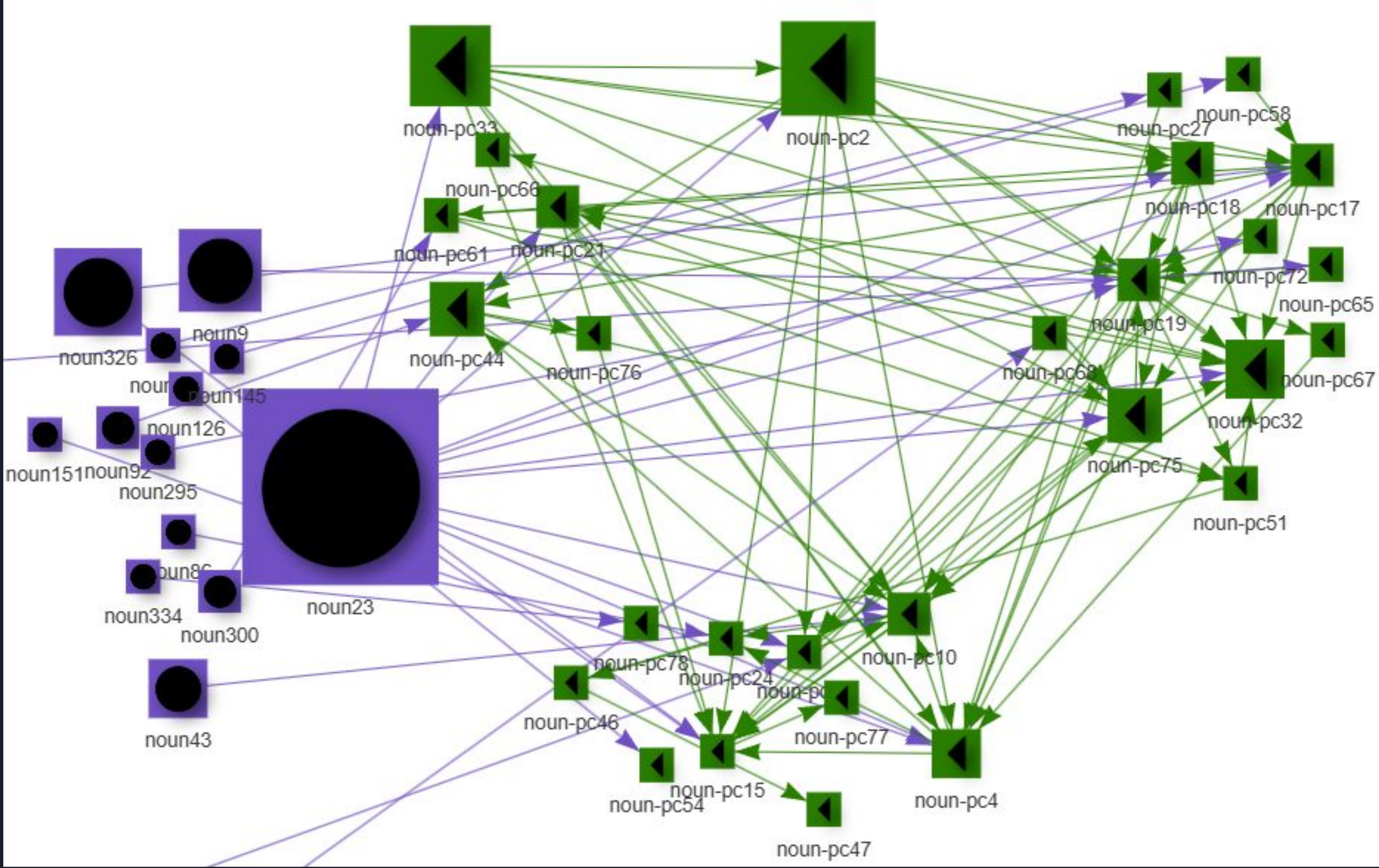
Results: Grammar Complexity

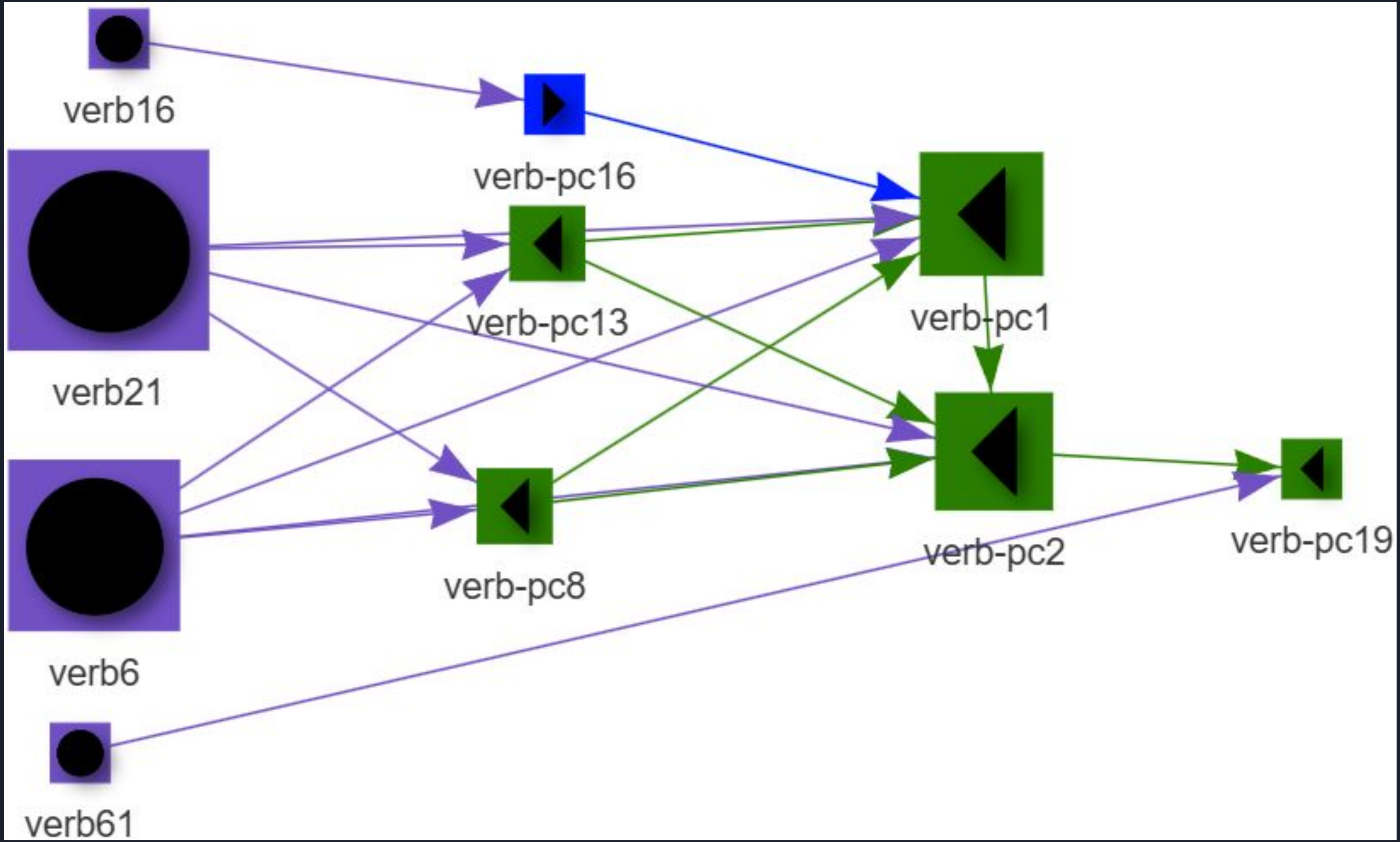
Dataset	N LCs	N PCs	V LCs	V PCs
Wambaya	24.1	20.3	19.3	24.1
Wambaya	23.8	23.1	19.3	24.1
Lezgi	61.2	27.2	41.1	27.0
Lezgi	53.5	26.2	29.1	25.1
Manipuri	20.8	23.7	33.2	32.9
Manipuri	17.7	23.0	25.5	34.1
Wakhi	12.0	6.9	8.6	9.1
Wakhi	11.9	6.9	6.6	9.7
South Efate	34.5	11.9	13.8	11.1
South Efate	20.7	9.0	11.8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

Table 5.14: Baseline and Final Grammar Complexity Metrics by Dataset

Discussion

- Changes had similar effects for the development and test datasets
- Coverage was not affected but the amount of ambiguity decreased in some cases
- Number of LCs/PCs stayed the same or decreased (with a few exceptions)
  - The most significant instances of additional merging were due to the increased number of merging rounds
- PCs were much more canonical on average
  - i.e. the affixes within a PC almost always expressed the same set of features
- Some inputs graphs remained relatively unchanged from the baseline
  - Featureless affixes are likely the culprit
- Feature cluster boosting had little effect
- Overall, the typologically-informed approach was successful, even though the inference approach remained heuristic in nature


noun-pc33
noun-pc2
noun-pc58
noun-pc27
noun-pc66
noun-pc18
noun-pc17
noun-pc61
noun-pc21
noun-pc72
noun-pc65
noun9
noun-pc19
noun326
noun-pc44
noun-pc76
noun-pc68
noun-pc67
noun145
noun126
noun-pc32
noun151
noun92
noun295
noun-pc75
noun-pc51
noun334
noun300
noun23
noun-pc10
noun-pc78
noun-pc24
noun43
noun-pc46
noun-pc77
noun-pc15
noun-pc54
noun-pc4
noun-pc47


verb16
verb-pc16
verb21
verb-pc13
verb-pc1
verb6
verb-pc8
verb-pc2
verb-pc19
verb61

Future work

- Case study with a specific language
- Account for additional non-canonical phenomena
- Prioritize feature overlap over input overlap to a greater extent
- More accurate tracking of the features expressed by affixes in a PC, specifically for those that can be subject- or object-marking (e.g. agreement features)

Selected References

Bonami, O., & Crysmann, B. (2013). Morphotactics in an information-based model of realisational morphology. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, 27–47. https://doi.org/10.21248/hpsg.2013.2

Chelliah, S. L. (2019). Meithei texts. [Manipur Digital Resources in UNT Digital Library. University of North Texas Libraries. (Accessed August 2019)].

Clifton, J. M., Donet, C. M., & Smith, J. (2024). Lezgi Texts from Azerbaijan [Endangered Languages Archive. Handle: http://hdl.handle.net/2196/fbbf9e7c-2074-4ef1-88e1- 6804b5b2ec92 [Accessed: January 2026]].

Goodman, M. W. (2013). Generation of Machine-Readable Morphological Rules from Human-Readable Input.

Harley, H. (2019). Hiaki text corpus. [University of Arizona. Unpublished FieldWorks (FLEx) project. (Accessed August 2019).].

Howell, K. (2020). Inferring Grammars from Interlinear Glossed Text: Extracting Typological and Lexical Properties for the Automatic Generation of HPSG Grammars. Retrieved April 16, 2025, from http://hdl.handle.net/1773/46080

Liu, T. (2025). *The Interplay of Dataset Characteristics in Automated Grammar Generation: A Study with the AGGREGATION System* [Master’s thesis, University of Washington]. Retrieved September 26, 2025, from https://hdl.handle.net/1773/53678

Nordlinger, R. (1998). *A Grammar of Wambaya, Northern Australia*. Pacific Linguistics.

Obrtelová, J. (2019). *From Oral to Written: A Text-linguistic Study of Wakhi Narratives* [Doctoral dissertation]. Uppsala University.

O’Hara, K. (2008). A Morphotactic Infrastructure for a Grammar Customization System.

Selected References, cont.

Thieberger, N. (2006). *A Grammar of South Efate*. University of Hawai'i Press. Retrieved January 13, 2026, from https://muse.jhu.edu/pub/5/monograph/book/8234

Wax, D. (2014). *Automated Grammar Engineering for Verbal Morphology* [Master's thesis, University of Washington].

Zamaraeva, O. (2016, August). Inferring Morphotactics from Interlinear Glossed Text: Combining Clustering and Precision Grammars. In M. Elsner & S. Kuebler (Eds.), *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology* (pp. 141–150). Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-2021

Zamaraeva, O., Howell, K., & Bender, E. M. (2019a, February). Handling cross-cutting properties in automatic inference of lexical classes: A case study of Chintang. In A. Arppe, J. Good, M. Hulden, J. Lachler, A. Palmer, L. Schwartz, & M. Silfverberg (Eds.), *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages* Volume 1 (Papers) (pp. 28–38). Association for Computational Linguistics. Retrieved April 30, 2025, from https://aclanthology.org/W19-6005/

Zamaraeva, O., Kratochvíl, F., Bender, E. M., Xia, F., & Howell, K. (2017, March). Computational Support for Finding Word Classes: A Case Study of Abui. In A. Arppe, J. Good, M. Hulden, J. Lachler, A. Palmer, & L. Schwartz (Eds.), *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages* (pp. 130–140). Association for Computational Linguistics. https : //doi.org/10.18653/v1/W17-0118