The Interplay of Dataset Characteristics in Automated Grammar Generation: A Study with the AGGREGATION System

Tongxi (Tom) Liu

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2025

Committee:
Emily M. Bender
Fei Xia

Program Authorized to Offer Degree:
Linguistics

University of Washington

Abstract

The Interplay of Dataset Characteristics in Automated Grammar Generation: A Study with the AGGREGATION System

Tongxi (Tom) Liu

Chair of the Supervisory Committee:
Emily M. Bender
Linguistics

In this thesis, I investigate how linguists can effectively prepare Interlinear Glossed Text (IGT) data for use with the AGGREGATION grammar inference system, particularly under constraints such as limited time, sparse annotations, and variable corpus quality. AGGREGATION aims to automate the creation of precision Head-driven Phrase Structure Grammar (HPSG) grammars from IGT, but its output quality depends heavily on input structure and annotation consistency. To explore this, I develop a modeling framework to evaluate how structural and annotation-based features (such as affix ambiguity, type-stems ratio, and POS tag source) affect grammar quality across 75,000 grammar runs on 25 datasets. I use both linear mixed-effects models and XGBoost to identify predictors of four key metrics: coverage, ambiguity, morphological complexity, and inference time. Results show that smaller, structurally coherent datasets often outperform larger, noisier ones. Manual POS tags improve coverage and generalization but increase ambiguity, while automatic tags result in cleaner grammars with lower parse success. A case study on Meitei highlights how annotation quality interacts with language-specific features. This work offers practical guidance for preparing IGT data for grammar generation and proposes future improvements to AGGREGATION, including support for structure-aware sampling and multi-version grammar comparison.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Emily M. Bender, whose mentorship has shaped every stage of my academic journey. Since I first took LING 566 as an undergraduate at the University of Washington in 2021, she introduced me to HPSG and grammar engineering and has continually helped me grow as a scholar and offered clear guidance and steady support. Throughout the CLMS program, her thoughtful feedback on my writing, her generous time and attention, and her example of how to engage with scientific work critically and responsibly have been invaluable to my development as a researcher.

I am also sincerely grateful to Professor Fei Xia, whose instruction gave me a solid foundation in natural language processing and whose guidance helped shape the experimental methodology of this thesis. Her teaching and support have greatly enhanced my understanding of both computational techniques and their linguistic relevance.

Lastly, I want to acknowledge the late Jiahui Huang, who served as my undergraduate mentor. At a time when I lacked clear direction, our in-depth conversations opened the door to linguistics and inspired me to pursue this path. His kindness, curiosity, and intellectual generosity continue to influence me deeply.

DEDICATION

To my family and my loved one, with love and gratitude. To those who guided me at the beginning of this journey.

路漫漫其修远兮，吾将上下而求索。

My way layeth remote and so far, far away;

I shall go up and down to make my long search aye.

——屈原《离骚》(translated by Sun Dayu)

Chapter 1

INTRODUCTION

Grammar engineering combines computational linguistics and linguistic theory to create human and machine-readable grammars that connect surface strings and their semantic representations. The AGGREGATION project (Bender et al., 2014; Howell and Bender, 2022a) makes efforts to generate grammars for low-resource and endangered languages using Interlinear Glossed Text (IGT) data. It seeks to partially automate the creation of precise Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994), which traditionally demands manual effort. The AGGREGATION project builds upon the foundational work of the LinGO Grammar Matrix (Bender et al., 2002, 2010; Zamaraeva et al., 2022), the RiPLeS project (Xia et al., 2016), and the INTENT toolkit (Georgi, 2016), integrating additional tools like the MOM lexical and morphological inference system (Wax, 2014) and the BASIL syntactic inference system (Howell, 2020) to streamline the grammar generation process.

In this thesis, I explore how linguists working under time and resource constraints can best prepare data for use with the AGGREGATION system. This inquiry acknowledges the practical challenges linguists face in collecting and preparing data, which is typically gathered for other research purposes but may later be repurposed for machine-readable grammar generation. It focuses on discerning the most effective data preparation strategies to maximize the AGGREGATION system's performance and output quality. Specifically, this research seeks to determine the optimal characteristics of input datasets, including size, the inclusion or exclusion of linguist provided Part-of-Speech (POS) tags, and other relevant features, that are most conducive to producing effective grammars, in terms of higher coverage and lower ambiguity, with the AGGREGATION system.

The remainder of this thesis is structured as follows. In Chapter 2, I introduce related

background by examining the AGGREGATION project, detailing its approach to using IGT data for automating grammar generation, and discussing the components and methodologies underpinning this effort, including the MOM and the BASIL inference systems. I further explain the theoretical foundation HPSG provided and the Grammar Matrix's role in facilitating grammar engineering.

In Chapter 3, I present my methodology in detail, which defines the independent and dependent variables, describes the selection and preprocessing of datasets, and outlines the procedures for grammar generation and evaluation. I also introduce the methodological approach for a case study on the Meitei language.

In Chapter 4, I analyze the technical implementation of the experimental pipeline by describing the enhancements made to the original AGGREGATION pipeline scripts, the development of the AGGFlow system, and the challenges encountered in adapting these tools to support my research objectives, which may provide an insight into the computational solutions that streamline the grammar generation process.

In Chapter 5, I report the results of 75,000 grammar generation and evaluation runs conducted across 25 datasets. These experiments reveal that grammar quality is not solely dependent on data size, but rather on the interaction between structural features, annotation quality, and morphological complexity. Notably, I find that features like Affix Ambiguity and Allomorph Ratio jointly shape both parsing coverage and ambiguity. Manual POS tags tend to support better generalization, though at the cost of increased structural noise. The Meitei case study further illustrates how annotation choices and dataset size affect grammar usability, emphasizing the importance of careful data curation. These findings contribute practical guidelines for preparing data in ways that improve grammar generation outcomes with AGGREGATION.

In Chapter 6, I reflect on the implications of these findings for data preparation and grammar engineering practices. I revisit the original research question to clarify what structural and annotation features most influence grammar quality, especially under practical constraints. I distinguish between proxy variables that describe overall corpus properties

and actionable structural predictors that linguists can directly influence. I also discuss the modeling limitations and interpretive boundaries of the current approach, and outline directions for future work, including typologically informed modeling and enhancements to the AGGREGATION system to support semi-automatic data selection and grammar evaluation.

Chapter 2

BACKGROUND

In the Background chapter of this thesis, I aim to explore the conceptual foundations of the AGGREGATION project and related background. This exploration begins with an overview of the AGGREGATION project's motivations, its innovative approach to using IGT data for grammar automation, and the theoretical and practical foundations that support this effort, as discussed in Section 2.1. Following this, Section 2.2 provides a detailed look at the data repository utilized in this study.

Subsequent sections pertain to the key components and methodologies integral to the AGGREGATION pipeline. Section 2.3 introduces the input format, including Interlinear Glossed Texts (IGT) and their transformation through Xigt and enrichment via INTENT. To show how morphological analysis and grammar generation work, the implementation of the Matrix-Odin Morphology (MOM) system is briefly introduced in Section 2.4. Further, Section 2.5 introduces BASIL, an extension of previous grammar inference efforts, inferring additional linguistic features to enhance grammar generation. The Head-driven Phrase Structure Grammar (HPSG) theoretical framework and its application within the AGGREGATION project are discussed in Section 2.6, emphasizing how it bridges syntactic structures with semantic interpretations. The following Section (2.7) describes the grammar specification file generated by the AGGREGATION inference system and loaded by the LinGO Grammar Matrix to produce a grammar. Then, aspects of grammar testing and parsing, facilitated by tools such as ACE and ART, are covered in Section 2.8. Lastly, the motivation of this research question is discussed in Section 2.9.

2.1 The AGGREGATION Project

The foundational goals of the AGGREGATION project, as articulated by Bender et al. (2012, 2013), center on the development of an automated system designed to generate precise Head-driven Phrase Structure Grammar (HPSG) grammars by integrating two types of linguistic resources: interlinear glossed text (IGT), which contains morphosyntactic annotations produced in the context of language documentation, and the LinGO Grammar Matrix (Bender et al., 2002, 2010; Zamaraeva et al., 2022), a crosslinguistic grammar customization system that outputs precision grammars based on typological specifications. The system is designed to infer linguistic properties directly from IGT data in order to populate the Grammar Matrix's questionnaire and generate corresponding grammar fragments.

Early stages of the project focused on identifying a subset of high-level typological features from IGT, such as constituent word order and case alignment systems. These features correspond to parameters that users of the Grammar Matrix customization system can specify as they use the system to create a grammar. The approach of the AGGREGATION project involves syntactic projection from the English translations in IGT to the source language line, as well as the analysis of glosses for morphosyntactic patterns. This method builds on prior work from the RiPLes project (Xia and Lewis, 2007; Lewis and Xia, 2008; Xia and Lewis, 2009; Xia et al., 2016) and is intended to reduce the manual effort typically involved in creating machine-readable grammars.

Developments in inference for the Lexicon & Morphology parts of the customization system questionnaire were advanced by the work of Wax (2014) and Zamaraeva (2016); Zamaraeva et al. (2019a), who pioneered methods for automatically generating comprehensive lexicons and morphotactics directly from IGTs. These methodologies are potentially helpful for understanding the morphological structures of under-documented languages. The study on the Chintang language by Zamaraeva et al. (2019a) illustrated the practical application of these computational techniques by demonstrating the system's capability to generate grammars that can parse running text with a degree of accuracy comparable to human-built

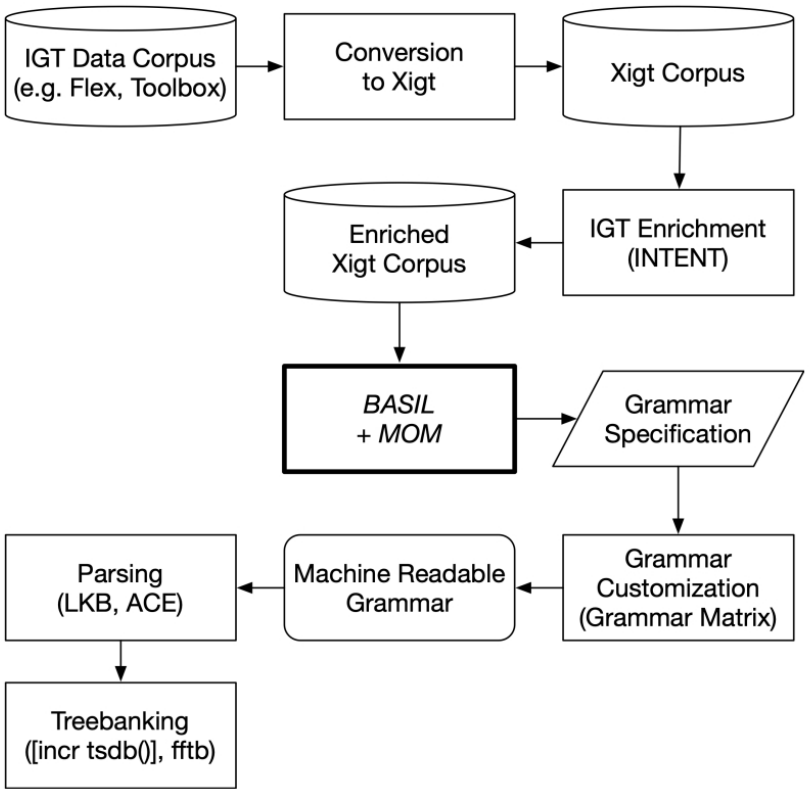


Figure 2.1: AGGREGATION Pipeline (Howell 2020, p. 16)

grammars. Subsequent enhancements and applications by Howell (2020), Conrad (2021), Howell and Bender (2022b), Dods (2022), and Lin (2023) have expanded the scope of grammatical features that the AGGREGATION system can infer, further refining the system's accuracy and utility.

2.1.1 AGGREGATION Pipeline

The AGGREGATION pipeline, as detailed in Figure 2.1 by Howell (2020), represents a multi-stage process that begins with the input of raw IGT data.

This data is first converted to the Xigt format with a structured XML representation suitable for further computational processing (Section 2.3.2). In a subsequent step, the

Xigt corpus undergoes enrichment through the INTENT system (Section 2.3.3). Upon this enriched data, a sequence of analytical processes is applied, including the inference of morphological rules (Section 2.4) and lexical items (Section 2.5), followed by the analysis of syntactic properties. The final step in this process is the generation of a grammar specification that, through the Grammar Matrix customization system (Bender et al., 2002, 2010; Zamaraeva et al., 2022) (Section 2.7), is transformed into a machine-readable HPSG (Pollard and Sag, 1994) grammar. The grammar can be used with ACE (Crysmann and Packard, 2012) to parse or generate sentences, and ART (Automated Regression Testing)[1] to perform batch processing and compare results against stored profiles (Section 2.8).

The AGGREGATION project and its pipeline are central to the research presented in this thesis. By examining the efficiency and effectiveness of the AGGREGATION system in generating grammars from IGT data, I seek to identify the optimal characteristics of input datasets that enhance the system's performance, thus providing potential insights for field linguists preparing data for grammar generation computational analysis. Furthermore, the findings could potentially contribute to the refinement of the exsiting system.

2.2 Data Overview

I utilize a carefully selected subset of datasets from the AGGREGATION data repository,[2] which encompasses enriched Xigt data (Section 2.3.3) for a diverse array of languages. The datasets represent a range of linguistic families, and the original data formats include Toolbox, Flex, Odin, Microsoft Word, LaTeX, XML, and ELAN. Table 2.1 shows an overview of the datasets employed in this study and information about the languages they are drawn from.

---

[1] https://sweaglesw.org/linguistics/libtsdb/art.html

[2] https://git.ling.washington.edu/agg/data
Permissions for general use of this repository have not been granted to the public. The datasets it contains originate from language documentation projects and should remain under the control of the linguists who produced them and, where applicable, the communities whose languages they represent. The AGGREGATION project is not structured as a long-term archive.

ISO	Language	Family	Data Format	Data Source
aqn	Northern Alta	Austronesian	Flex	García-Laguía (2022)
bbl	Tsova-Tush	Nakh-Daghestanian	Flex	Hauk (2019)
ctn	Chintang	Sino-Tibetan	Toolbox	Bickel et al. (2016, 2011) Stoll et al. (2016)
erk	South Efate	Austronesian	Toolbox	Thieberger (2006b,a)
esu	Yup'ik	Eskimo-Aleut	Odin	Strunk (2019)
ikx	Ik	Kuliak	LaTex	Schrock (2019)
kre	Panara	Nuclear-Macro-Je	LaTex	Bardagil Mas (2019)
lez	Lezgi	Nakh-Daghestanian	Flex	Donet (2019)
mni	Meitei	Sino-Tibetan	Flex	Chelliah (2011, 2019)
nuk	Nuuchahnulth	Wakashan	Flex	Inman (2019)
qux	Yaoyos Quechua	Quechuan	Toolbox	Shimelman (2019)
tdh	Thulung	Sino-Tibetan	Toolbox	Lahassois (2019)
tsz	P'urhepecha	Language isolate	Flex	Paz et al. (2022)
wbl	Wakhi	Indo-European	Flex	Kaufman et al. (2020)
wmb	Wambaya	Mirndi	Odin	Nordlinger (1998)
yaq	Haiki	Uzo-Aztecan	Flex	Harley (2019)
yak	Yakima Sahaptin	Sahaptian	LaTex	Hargus (2019)
ybh	Yakkha	Sino-Tibetan	Toolbox	Schackow (2022)

Table 2.1: Datasets Used in the Study

The datasets used in this study are drawn from those available in the AGGREGATION data repository. From this collection, I selected datasets that contain sufficiently detailed and well-formed IGT suitable for analysis. Datasets that were too small or could not be converted from their original formats to Xigt were excluded. The selected datasets span a range of language families and provide the necessary linguistic detail for examining grammar

inference and generation processes.

2.3 IGT, Xigt, INTENT, and Enriched Xigt

This section outlines the structure and function of Interlinear Glossed Text (IGT) as used in the AGGREGATION pipeline, and describes how it is represented by the Xigt format and further enriched using the INTENT system to support automated grammar generation.

2.3.1 IGT

Interlinear Glossed Text (IGT) represents a fundamental data type within linguistic research (Comrie et al., 2008), especially for the documentation and analysis of endangered languages. IGTs provide a morpheme-by-morpheme translation, connecting the source language with a language of broader communication, typically English. The structure of an IGT reflects the linguistic analysis conducted by the researcher, with insights into the language's structure encoded through the glossing process. Example (1) is an IGT in the Toolbox format from Meitei language (Chelliah, 1997), which features various tiers such as transcription (tx), gloss word (gw), morpheme phonetic (mph), morpheme gloss (mgl), part of speech (pos), morpheme English (meng), and English translation (eng).

(1) *An example IGT in the Toolbox format (Chelliah 1997, p. 105):*

\tx əygə catsi
\gw əy-kə cat-si
\mph əy -kə cat -si
\mgl I -ASS go -SUP
\pos PRO -Unsure VERB -Unsure
\meng I too let's go
\eng 'Let's go together!'

In IGT, *grams* refer to elements on the gloss line that provide glosses for grammatical morphemes in the source language. These grammatical morphemes or markers indicate specific grammatical functions such as tense, aspect, mood, case, gender, number, and other grammatical categories that are relevant to the language being studied. Unlike glosses using English lemmas (e.g., *I*, *go*), which are the base forms of words, *grams* (e.g., *-ASS*, *-SUP*) are not necessarily complete words in English or any other language; instead, they are abbreviations or symbols that represent grammatical concepts.

2.3.2 Xigt

Xigt (Extensible Interlinear Glossed Text; Goodman et al. (2015)), represents an extension to the traditional IGT format, specifically designed to accommodate a wider array of linguistic information. Unlike standard IGT formats, which primarily focus on providing a morpheme-by-morpheme gloss, Xigt is crafted to be inherently extensible to encode additional linguistic annotations and metadata for computational analysis and grammar engineering.

Example (2) is the translation of the Toolbox-format IGT (1) into Xigt. It has explicit alignments between tiers, a feature that enhances the traditional IGT format by making the relationships between elements clear and computationally accessible. These alignments link morphemes, glosses, and part-of-speech tags directly. This structured approach supports further computational tasks and ensures that the rich linguistic detail inherent in the original data is preserved and highlighted.

(2) *An example IGT in Xigt format:*

```
<igt id="igt1">
  <tier id="l" type="phrases">
    <item id="l1">əygə catsi</item>
  </tier>
  <tier id="p" type="phrases">
    <item id="p1">əy-kə cat-si</item>
  </tier>
  <tier id="w" type="words" segmentation="p">
```

```
  <item id="w1" segmentation="p1[0:4]">əy-kə</item>
  <item id="w2" segmentation="p1[5:9]">cat-si</item>
</tier>
<tier id="sw" type="words" segmentation="l">
  <item id="sw1" segmentation="l1[0:3]">əygə</item>
  <item id="sw2" segmentation="l1[5:9]">catsi</item>
</tier>
<tier id="m" type="morphemes" segmentation="w">
  <item id="m1.1" type="stem" segmentation="w1[0:2]">əy</item>
  <item id="m1.2" type="clitic" segmentation="w1[3:4]">-kə</item>
  <item id="m2.1" type="stem" segmentation="w2[0:2]">cat</item>
  <item id="m2.2" type="suffix" segmentation="w2[3:5]">-si</item>
</tier>
<tier id="g" type="glosses" alignment="m">
  <item id="g1.1" alignment="m1.1">I</item>
  <item id="g1.2" alignment="m1.2">ASS</item>
  <item id="g2.1" alignment="m2.1">go</item>
  <item id="g2.2" alignment="m2.2">SUP</item>
</tier>
<tier id="pos" type="pos" alignment="m">
  <item id="pos1.1" alignment="m1.1">PRO</item>
  <item id="pos1.2" alignment="m1.2">Unsure</item>
  <item id="pos2.1" alignment="m2.1">VERB</item>
  <item id="pos2.2" alignment="m2.2">Unsure</item>
</tier>
<tier id="msa" type="msa" alignment="m">
  <item id="msa1.1">n</item>
  <item id="msa1.2">&lt;Not Sure&gt;</item>
  <item id="msa2.1">&lt;Not Sure&gt;</item>
  <item id="msa2.2">Attaches to any category</item>
</tier>
<tier id="cf" type="cf" alignment="m">
  <item id="cf1.1">əy</item>
  <item id="cf1.2">-kə</item>
  <item id="cf2.1">cat</item>
  <item id="cf2.2">-si</item>
</tier>
<tier id="meng" type="morphemeEnglish" alignment="m">
  <item id="meng1.1" alignment="m1.1">I</item>
  <item id="meng1.2" alignment="m1.2">too</item>
  <item id="meng2.1" alignment="m2.1">let's</item>
  <item id="meng2.2" alignment="m2.2">go</item>
</tier>
<tier id="eng" type="translations" alignment="p">
  <item id="eng1" alignment="p1">Let's go together!</item>
```

```
  </tier>
</igt>
```

2.3.3 INTENT & Enriched Xigt

The Xigt format supports the inclusion of additional linguistic information beyond basic glossing, such as part-of-speech (POS) tags and syntactic dependencies. These types of annotations are utilized by the AGGREGATION system to support grammar inference and generation. However, this information is not typically included in the original IGT and must be added. The INTENT tool (Georgi, 2016) is used for this purpose, providing automatic enrichment of Xigt data by projecting POS tags and syntactic dependency relations based on the English gloss and translation lines.

Example (3) shows how Xigt data appear after enrichment through INTENT. The added layers of linguistic annotation exemplifies the enrichment process. In particular, INTENT adds syntactic dependency annotations and additional POS tags aligned to the morpheme tier. These enhancements are projected from the English translation. The `dependencies` tier captures syntactic relationships such as `aux` (auxiliary), `root` (main verb), and `advmod` (adverbial modifier), which are inferred through analysis of the English sentence structure using `SpaCy`[3] with the `en_core_web_lg` model.[4]

(3) *An example of enriched Xigt:*

```
<igt id="igt1">
  <tier id="l" type="phrases">
    <item id="l1">əygə catsi</item>
  </tier>
  <tier id="p" type="phrases">
    <item id="p1">əy-kə cat-si</item>
```

[3] https://spacy.io/

[4] https://github.com/explosion/spacy-models/releases/download/en_core_web_lg-2.0.0/en_core_web_lg-2.0.0.tar.gz

```
</tier>
<tier id="w" type="words" segmentation="p">
  <item id="w1" segmentation="p1[0:4]">əy-kə</item>
  <item id="w2" segmentation="p1[5:9]">cat-si</item>
</tier>
<tier id="m" type="morphemes" segmentation="w">
  <item id="m1.1" type="stem" segmentation="w1[0:2]">əy</item>
  <item id="m1.2" type="clitic" segmentation="w1[3:4]">-kə</item>
  <item id="m2.1" type="stem" segmentation="w2[0:2]">cat</item>
  <item id="m2.2" type="suffix" segmentation="w2[3:5]">-si</item>
</tier>
<tier id="g" type="glosses" alignment="m">
  <item id="g1.1" alignment="m1.1">I</item>
  <item id="g1.2" alignment="m1.2">ASS</item>
  <item id="g2.1" alignment="m2.1">go</item>
  <item id="g2.2" alignment="m2.2">SUP</item>
</tier>
<tier id="pos" type="pos" alignment="m">
  <item id="pos1.1" alignment="m1.1">PRO</item>
  <item id="pos1.2" alignment="m1.2">Unsure</item>
  <item id="pos2.1" alignment="m2.1">VERB</item>
  <item id="pos2.2" alignment="m2.2">Unsure</item>
</tier>
<tier id="msa" type="msa" alignment="m">
  <item id="msa1.1">n</item>
  <item id="msa1.2">Attaches to any category</item>
  <item id="msa2.1">n</item>
  <item id="msa2.2">Attaches to any category</item>
</tier>
<tier id="cf" type="cf" alignment="m">
  <item id="cf1.1">əy</item>
  <item id="cf1.2">-kə</item>
  <item id="cf2.1">cat</item>
  <item id="cf2.2">-si</item>
</tier>
<tier id="eng" type="translations" alignment="p">
  <item id="eng1">Let's go together!</item>
</tier>
<tier id="tw" type="words" segmentation="eng">
  <item id="tw1">Let's</item>
  <item id="tw2">go</item>
  <item id="tw3">together</item>
</tier>
<tier id="tw-pos" type="pos" alignment="tw"
data-creation-time="2024-02-07 13:13:17" data-method="spacy" data-
```

```
  provenance="INTENT2-2.0a6">
    <item id="tw-pos-1" alignment="tw1">VERB</item>
    <item id="tw-pos-2" alignment="tw2">VERB</item>
    <item id="tw-pos-3" alignment="tw3">ADV</item>
  </tier>
  <tier id="words_d" type="words" segmentation="eng">
    <item id="w_d1">Let's</item>
    <item id="w_d2">go</item>
    <item id="w_d3">together</item>
  </tier>
  <tier id="dependencies" type="dependencies" dep="tw" head="tw"
  data-creation-time="2024-02-07 13:13:17" data-method="spacy" data-
  provenance="INTENT2-2.0a6" dep="tw" head="tw">
    <item id="dep1" dep="tw1" head="tw2">aux</item>
    <item id="dep2" dep="tw2">root</item>
    <item id="dep3" dep="tw3" head="tw2">advmod</item>
  </tier>
</igt>
```

2.4 Matrix-Odin Morphology (MOM) Implementation

The Matrix-Odin Morphology (MOM) system (Wax, 2014; Zamaraeva et al., 2019b, 2017; Zamaraeva, 2016) is a core component of the AGGREGATION project, designed to automate the analysis of verbal and nominal morphology using IGT data and the customization capabilities of the Grammar Matrix. MOM was initially developed for data from the Online Database of Interlinear Glossed Text (ODIN) and now operates over structured Xigt representations.

MOM assumes that the alignment between language, gloss, and translation tiers has already been determined and encoded in the IGT. Its primary function is to analyze this aligned data across multiple IGT instances in order to identify sets of morphemes that occupy the same morphological slots. By detecting patterns in how morphemes co-occur and distribute across IGTs, MOM induces a position class-based morphological structure that can be mapped to the Grammar Matrix's customization framework.

Operating on IGT data in Xigt format, MOM accesses structured tiers such as the language line, gloss line, and translation line. It uses these tiers to group morphemes according

to shared distributional and functional properties, such as tense, aspect, and mood, and uses this information to infer both their position within a morphological template and their associated morphosyntactic values.

The results of this analysis are compiled into a `choices` file, as described in Section 2.7, which encodes morphotactic rules and feature associations for use by the Grammar Matrix customization system.

2.5 BASIL

BASIL (Howell, 2020) (Building Analyses from Syntactic Inference in Low-Resource Languages) extends and integrates previous grammar inference work within the AGGREGATION project into a single end-to-end system. Operating on enriched IGT data in Xigt format, BASIL infers a wide range of syntactic and morphological properties and encodes them in grammar specification files compatible with the LinGO Grammar Matrix. In addition to earlier components focused on morphological analysis (e.g., verb stems, affixes, and morphotactic patterns), BASIL introduces modules for inferring syntactic features such as argument optionality, coordination, and sentential negation, as well as additional lexical items including determiners, adpositions, auxiliaries, and coordinators. The system also models grammatical agreement and captures features such as person, number, and gender on both nouns and verbs, and represents tense, aspect, and mood, whether expressed morphologically or periphrastically.

2.6 Head-driven Phrase Structure Grammar (HPSG)

Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994) is a lexicalist, surface-oriented, and sign-based grammatical theory for analyzing natural languages. HPSG posits that linguistic knowledge is encoded in richly detailed lexical entries and that syntax and semantics are closely interlinked through *typed feature structures*. It is the fundamental skeleton that supports the AGGREGATION project, specifically through the Grammar Matrix

(Section 2.7), facilitating the automatic generation of human and machine readable grammars for low-resource languages.

2.6.1 Typed Feature Structures

Central to HPSG is the concept of typed feature structures, which are complex objects encapsulating grammatical information. Each feature within a structure has a unique identifier and is associated with a value, which itself can be a feature structure. This hierarchical organization allows for the representation of detailed linguistic information. Consider the pronoun *she*, which can be described by a feature structure indicating it is *third* person, *feminine* gender, in the *nominative* case, and *singular* number.

2.6.2 Minimal Recursion Semantics (MRS)

Minimal Recursion Semantics (MRS) (Copestake et al., 2005) is a framework for representing the sentential semantics in a highly structured and formalized manner. Developed to be consistent with the HPSG framework, MRS captures the meaning of sentences through a set of relations and links between them (via shared variables), allowing for the representation of predicate-argument structures, scope relations, and other semantic phenomena in a flat, non-hierarchical format. This semantic representation is used for representing the underlying meaning of sentences beyond their surface structure for detailed analysis and processing by computational linguistic tools. MRS representations are beneficial for tasks such as semantic parsing, machine translation, and natural language understanding. MRS is a major component of grammar engineering projects like the AGGREGATION system.

To illustrate HPSG's capability to capture both syntactic and semantic information, consider the sentence *She sings*. Example (4) shows the representation of the parse tree and Figure 2.2 is its corresponding MRS representation generated by the LKB parsing software using the mini English grammar provided by the Grammar Matrix customization system.[5]

[5]https://matrix.ling.washington.edu/customize/matrix.cgi?choices=web/sample-choices/mini-english

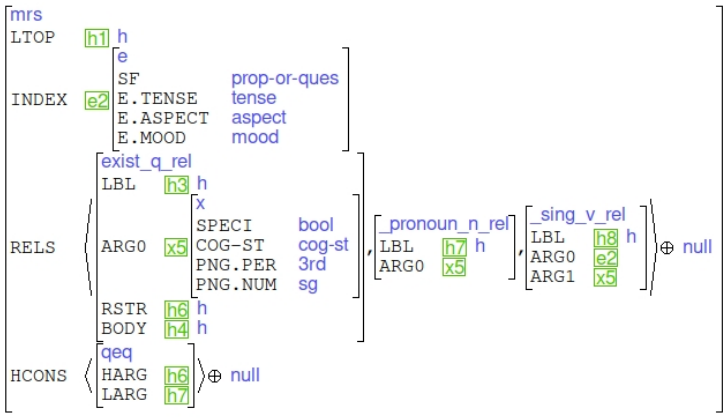
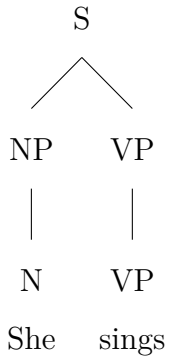


Figure 2.2: Simple MRS for *She sings* (Generated by LKB)

(4)



The examples demonstrate how HPSG captures linguistic information. The parse tree shows a sketch of the syntactic structure, while the MRS provides a semantic representation. In the parse tree provided, the structure is divided into two main parts: NP (Noun Phrase) for *She* and VP (Verb Phrase) for *sings*. Within these structures, each word is associated with a complex set of features, including part of speech, grammatical number, person, tense, and so on. These features are hierarchically organized, which allows for the specification of agreement and other syntactic phenomena.

The MRS indicates that the sentence has a top handle (h), an event (e) of singing in the present tense, and a singular third-person subject (x). The top handle (h) serves as an

entry point to the scope tree for traversal through the sentence's semantic relations. The INDEX (e), representing an event denoted by the sentence, such as singing, identifies the sentence's main action. It carries tense information (present) and corresponds to the event or eventuality introduced by the predicate in the sentence. The INDEX is one of the variables used to represent events or entities in a sentence; in this case, it corresponds to the event introduced by the verb *sings*. A sentence may have multiple such variables if it involves more than one event or referential expression.

The distinction between the syntactic structure displayed by the parse tree and the semantic representation offered by MRS highlights the cross-linguistic applicability of the MRS framework. While the parse tree illustrates how sentences are organized syntactically, which can vary across languages due to differing grammatical rules and structures, MRS focuses on capturing the underlying semantic content of sentences. This semantic layer abstracts away from surface syntactic variation. It enables representations that are often more comparable across languages than their syntactic structures, while still reflecting language-specific properties.

2.7 The Grammar Matrix

The LinGO Grammar Matrix (Bender et al., 2002, 2010; Zamaraeva et al., 2022) is a framework for supporting the rapid development of broad-coverage, precision grammars for diverse languages. It provides a web-based questionnaire interface that guides users through a typologically informed set of linguistic phenomena, such as word order, case, agreement, and valence alternations, and produces a grammar specification known as a `choices` file. This file encodes information about the target language's syntactic, morphological, and lexical properties. The customization system uses the `choices` file to generate a machine-readable grammar in the Head-driven Phrase Structure Grammar (HPSG) and Minimal Recursion Semantics (MRS) formalisms.

In the AGGREGATION project, the Grammar Matrix is not used through the user-facing questionnaire interface. Instead, the system infers linguistic properties from IGT data

and automatically constructs a `choices` file, which is then submitted to the customization system to produce an implemented grammar. These inferred specifications span multiple linguistic domains. The resulting grammars can be used with parsing and generation tools, such as ACE and LKB, to analyze the input data or generate new utterances.

Consider the following snippets from a sample `choices` file for a simplified fragment of English:

(5) *An example of a* `choices` *file:*

```
section=lexicon
  noun1_name=3rd-pronoun
    noun1_feat1_name=person
    noun1_feat1_value=3rd
    noun1_feat2_name=number
    noun1_feat2_value=sg
  noun1_det=imp
    noun1_stem1_orth=She
    noun1_stem1_pred=_pronoun_n_rel

section=morphology
  verb-pc1_name=pernum
  verb-pc1_obligatory=on
  verb-pc1_order=suffix
  verb-pc1_inputs=verb
    verb-pc1_lrt1_name=3sg
      verb-pc1_lrt1_feat1_name=person
      verb-pc1_lrt1_feat1_value=3rd
      verb-pc1_lrt1_feat1_head=subj
```

```
    verb-pc1_lrt1_feat2_name=number
    verb-pc1_lrt1_feat2_value=sg
    verb-pc1_lrt1_feat2_head=subj
    verb-pc1_lrt1_lri1_inflecting=yes
    verb-pc1_lrt1_lri1_orth=s
  verb-pc1_lrt2_name=pl
    verb-pc1_lrt2_feat1_name=number
    verb-pc1_lrt2_feat1_value=pl
    verb-pc1_lrt2_feat1_head=subj
    verb-pc1_lrt2_lri1_inflecting=no
  verb-pc1_lrt3_name=non-3rd
    verb-pc1_lrt3_feat1_name=person
    verb-pc1_lrt3_feat1_value=1st, 2nd
    verb-pc1_lrt3_feat1_head=subj
    verb-pc1_lrt3_lri1_inflecting=no
```

The first half of the snippet illustrates entries in the `lexicon` section of the `choices` file, which specifies lexical types and associated stems for categories like nouns, verbs, and adjectives. In this framework, a lexical entry connects surface forms (e.g., `She`) to syntactic and semantic properties used in the grammar. Each stem is associated with an `orth` field (the orthographic form) and a `pred` field, which specifies a surface predicate introduced by that lexical item in the semantics. These surface predicates follow a standardized naming convention (e.g., `_pronoun_n_rel`) and correspond to elementary predications in MRS, each of which introduces an intrinsic variable (typically of type `x` for nominals or `e` for events) and may relate to additional arguments. The `noun1_name` field identifies a lexical type, while the `feat` fields specify morphosyntactic features such as person and number associated with that type.

The latter half of the snippet comes from the `morphology` section of the `choices` file, which defines morphotactic properties, how morphemes are ordered and combined, for verbs in this case. In the Grammar Matrix framework, morphology is handled by specifying position classes, lexical rule types, and lexical rule instances. A position class, such as `verb-pc1`, groups affixes that occupy the same syntactic slot in the structure of a word (Goodman, 2013). The `verb-pc1_inputs` field identifies the lexical types (here, verbs) that these affixes apply to.

Each lexical rule type within a position class (e.g., `verb-pc1_lrt1`) defines a morphological pattern conditioned by grammatical features like person and number. The lexical rule instances, such as `verb-pc1_lrt1_lri1`, specify whether the rule is inflecting and, if so, provide the orthographic realization of the morpheme (e.g., the suffix `s` used for third-person singular present tense forms). This setup allows for fine-grained control over which forms appear under which grammatical conditions. The example includes rule types for third-person singular, plural, and non-third-person forms, showing how the system encodes basic inflectional paradigms.[6]

The morphological rules specified in the `choices` file interface directly with syntactic representations in the grammar through feature structures. Each affix introduced by a lexical rule instance contributes not only surface form (orthography) but also grammatical features, such as person, number, or case, that are passed up to the syntactic level. These features are encoded with heads for the parser to enforce agreement constraints during syntactic analysis. Additionally, the valence of each verb is defined in the lexicon section by specifying the number and type of syntactic arguments it takes (e.g., intransitive, transitive). As a result, the morphology and syntax components of the `choices` file jointly determine how words are formed and how they participate in larger syntactic structures.

[6]The Grammar Matrix customization system does not model morphophonological alternations. Users are encouraged to integrate a separate morphophonological analyzer if such phenomena are relevant for the language being modeled.

2.8 Parsing & Testing

Once a grammar has been generated by the Grammar Matrix customization system, the ACE (Answer Constraint Engine; Crysmann and Packard (2012)) and ART (Automated Regression Testing)[7] tools are used to parse and generate sentences and to evaluate grammar behavior.

ACE is a processor for HPSG grammars with MRS semantics. It is used to apply the grammar to input sentences, producing syntactic and semantic analyses in the form of parse trees and MRS representations. ART is a testing utility that facilitates interaction with ACE by automating batch parsing and generation tasks. It submits test items to ACE and records the results in a TSDB (TestSuite DataBase) profile with a consistent format for inspecting parses, checking generation outputs, and identifying mismatches between expected and actual analyses.

2.9 Linguistics 567

The Linguistics 567 (LING 567) course (Bender, 2023) at the University of Washington exemplifies the practical application of both manual and automated grammar generation methods. The coursework component involves students building implemented grammars for diverse languages. From 2019 to 2023, the course integrated the AGGREGATION system by providing students with automatically constructed grammar specifications to use as a starting point for their projects.

However, using AGGREGATION-generated `choices` files has revealed challenges in balancing the complexity and usability of these automatically inferred grammars. In the class of 2023, when I took this class, some large choice files were characterized by excessive noise, introducing a large number of morphological position classes without sufficient generalization, which can lead to high ambiguity and a less tractable grammar. Conversely, other files may be oversimplified models of the language, which results in grammars that lack as

[7] https://sweaglesw.org/linguistics/libtsdb/art.html

good as coverage compared to the large ones. These extremes pose challenges for students; overly complex grammars require extensive refactoring, while overly simple ones may not serve as a viable basis for further development. Students sometimes choose to avoid using the AGGREGATION-generated morphology part, opting to start that from scratch.

Despite these challenges, a subset of the generated grammars demonstrates reasonable coverage with manageable ambiguity, and can be developed further with relatively little modification. These more successful outcomes motivate my research presented in this thesis, where I investigate how to prepare data for the AGGREGATION system in ways that improve the quality of the resulting grammars. Rather than assuming perfect input or unrestricted annotation resources, the focus is on identifying data preparation strategies that are feasible within typical constraints of time and available documentation.

Chapter 3

METHODOLOGY

This chapter introduces the methodology that I use to investigate the research question: *What data preparation practices support effective use of the AGGREGATION system, given the constraints of limited time and resources?* This inquiry acknowledges the practical challenges linguists face in collecting and preparing data for the generation of machine-readable grammars. The objective is to identify effective data preparation strategies to enhance the AGGREGATION system's performance and output quality. This chapter outlines an approach to examining input dataset characteristics, such as the number of Interlinear Glossed Texts (IGTs), the source of Part-of-Speech (POS) tags, and other features, assessing their impact on grammar generation efficiency. Employing quantitative and qualitative analyses, it details the independent (Section 3.1.1) and dependent (Section 3.1.2) variables, the data preprocessing procedures (Section 3.3), the experimental setup (Section 3.4), and the evaluation criteria (Section 3.5) for the grammars generated in terms of coverage, ambiguity, noise, and efficiency, including a case study on the Meitei language (Section 3.6).

3.1 Variables of Interest

I consider a set of variables to evaluate the performance and quality of grammars generated by the AGGREGATION system. These variables are categorized into independent variables, which are input dataset characteristics, and dependent variables, which are outcomes reflecting the grammar's performance.

3.1.1 Independent Variables (X variables)

Various quantitative independent variables directly measure the input datasets' measurable properties, which including:

- **Number of Interlinear Glossed Texts (IGTs)**: The total count of IGTs in a dataset.
- **Number of Grams**: The total count of unique grams in the dataset.
- **Average IGT Length in Words**: The total word count across all IGTs divided by the *Number of IGTs*.

$$\text{Average IGT Length in Words} = \frac{\text{Total Word Count}}{\text{Number of IGTs}}$$

- **Average IGT Length in Morphemes**: Similar to the above, but considering morphemes (both stems and affixes) instead of words.

$$\text{Average IGT Length in Morphemes} = \frac{\text{Total Morpheme Count}}{\text{Number of IGTs}}$$

- **Type Stems Ratio**: This metric compares the number of unique word forms to the total number of stem types identified.

$$\text{Type Stems Ratio} = \frac{\text{Number of Distinct Word Forms}}{\text{Number of Distinct Stems}}$$

  A higher ratio indicates that a smaller set of stems appears in many different inflected forms. A lower ratio suggests that each stem has relatively few inflected variants.

- **Affix Ambiguity**: This metric measures the degree to which a single affix form corresponds to multiple grammatical meanings (grams) within the dataset. It is calculated as the number of distinct affix spellings divided by the number of distinct grammatical labels (grams) associated with them.

$$\text{Affix Ambiguity} = \frac{\text{Number of Distinct Affix Spellings}}{\text{Number of Distinct Grams}}$$

A higher ratio indicates that affix forms are reused across a broader range of grammatical functions and suggests greater morphological ambiguity. A lower ratio implies a tighter, more consistent mapping between affix forms and their grammatical meanings.

- **Allomorph Ratio**: The average number of distinct morphemes associated with each unique gloss. This is calculated by dividing the total number of distinct morphemes that appear across all glosses by the number of unique glosses in the dataset.

$$\text{Allomorph Ratio} = \frac{\text{Total Number of Distinct Morphemes Across All Glosses}}{\text{Number of Unique Glosses}}$$

- **POS Tags**: This is a categorical variable indicating the source of POS tags for the dataset. It can be *INTENT* or *linguistis*.

3.1.2 Dependent Variables (Y Variables)

Dependent variables include measures of grammar quality and efficiency, such as:

1. **Coverage**: Calculated as the proportion of sentences for which the system can produce at least one parse, relative to the entire collection of sentences under consideration.

$$\text{Coverage} = \frac{\text{Number of Sentences for Which at Least One Parse Was Produced}}{\text{Total Number of Sentences in Test Set}}$$

2. **Ambiguity**: This variable captures the system's tendency to generate multiple parse trees or readings for a single input sentence. It is detailed through:

   (a) **Average Readings per Sentence Parsed**: This calculates the mean number of parse trees produced for each sentence that the system successfully parsed.

$$\text{Average Readings} = \frac{\text{Total Number of Parses Generated for Parsed Sentences}}{\text{Number of Sentences Successfully Parsed}}$$

(b) **Maximum Readings**: Identifies the maximum number of parse trees generated by the system for any single sentence within the test set.

(c) **Logarithmic Inverse Average Readings (LIAR)**: Quantifies ambiguity by applying a logarithmic transformation to the average number of readings per parsed sentence to make high levels of ambiguity more interpretable and differences easier to compare.

$$\text{LIAR} = \frac{1}{\log(\text{Average Readings per Sentence Parsed} + 1)}$$

3. **Grammar Complexity**: Reflects the structural complexity and potential over-generation in a grammar as encoded in its `choices` file. The following metrics are computed by parsing the file and counting relevant entries:

(a) **Number of Noun/Verb Types**: The number of lexical types declared under the `section=lexicon`, identified by lines like `noun1_name=...` or `verb1_name=...`.

(b) **Average Number of Noun/Verb Stems per Type**: Computed as the total number of stem entries (e.g., `noun1_stem1_orth=...`, `verb1_stem1_orth=...`) divided by the number of corresponding lexical types. For instance:

$$\text{Average Noun Stems per Type} = \frac{\text{Total Number of Noun Stems}}{\text{Number of Noun Types}}$$

(c) **Number of Noun/Verb Inflections**: The number of position classes declared (e.g., lines like `noun-pc1_name=...`, `verb-pc1_name=...`).

(d) **Average Number of Noun/Verb Inflections per Position Class**: Calculated as the number of lexical rule types (e.g., `noun-pc1_lrt1_orth=...`, `verb-pc1_lrt1_orth=...`) divided by the number of position classes. For example:

$$\text{Average Verb Inflections per Position Class} = \frac{\text{Total Number of Verb LRTs}}{\text{Number of Verb Position Classes}}$$

(e) **Morphological Ambiguity**: Defined as the total number of verb inflectional rule instances divided by the number of unique orthographic forms for those instances.

$$\text{Morphological Ambiguity} = \frac{\text{Total Verb Rule Instances}}{\text{Unique Orthographic Forms}}$$

A higher ratio suggests repeated surface forms across multiple lexical rule types, which may reflect overgeneration or redundancy. A lower ratio implies greater orthographic diversity and may indicate a more targeted morphological specification.

4. **Inference Time**: Indicates the total amount of time for the AGGREGATION system to infer grammar and create a `choices` file from enriched Xigt dataset.

In later parts of this thesis, I use italics to distinguish specific variable names from their general conceptual usage. For example, *Coverage* refers to the specific variable used in modeling and analysis, whereas "coverage" (non-italicized) refers more broadly to the general notion of parse coverage. This convention applies similarly to variables such as *Affix Ambiguity*, *Inference Time*, and others.

3.2 Data Selection

This section elaborates on my reasons for selecting datasets from the AGGREGATION data repository for this study. While evaluating the AGGREGATION data repository's 36 available corpora, several challenges emerged. I found some datasets, despite being fully prepared (from the original format to enriched Xigt) in the past, incompatible with the current specifications of the AGGREGATION system. A few datasets, though fully prepared, contained too few well-aligned IGTs to allow the AGGREGATION system to generate meaningful grammars.

After a comprehensive review, 18 out of the 36 corpora were selected for this study. The chosen datasets represent various language families and formats, as detailed in Table 2.1 in Section 2.2. It is essential to acknowledge the extensive efforts of the linguists who collected and prepared these datasets. Their contributions form the foundation of this research and enable the exploration of data preparation strategies for grammar generation.

3.3 Data Preprocessing

The study commenced with a corpus comprising data from 18 datasets as shown in the Table 2.1. Out of these, 7 datasets have adequate linguist provided Part-of-Speech (POS) tags. The term "adequate" in this context indicates that the remaining 11 datasets either lacked linguist-provided POS tags or their POS information was excluded due to various factors, such as misalignment between the tags and the corresponding text.

To ensure consistency in data analysis, all raw linguistic data were converted into the Xigt format. The conversion is achieved by using various transformation algorithms appropriate to the original format of the dataset.[1] An additional preprocessing step was performed for datasets with available linguist-provided POS tags: a second version of the Xigt files was created without the original POS tiers. This allowed for a uniform approach in the subsequent enrichment phase.

I prepared three distinct groups of input for analysis in this study:

- Input A: Encompasses the datasets of the 11 languages that either lacked linguist-provided POS tags or were not included due to issues.

- Input B: Contains the datasets of the 7 languages with linguist-provided POS tags, maintained in their original form.

- Input C: Includes the 7 datasets from Input B after removing the linguist-provided POS tags.

[1] https://git.ling.washington.edu/agg/aggregation/-/blob/master/data_preparation.txt

This process resulted in 25 Xigt datasets representing 18 languages ready for enrichment. The enrichment step involved using the INTENT tool to generate POS tags for the datasets in Input A and Input C and to obtain syntactic dependency structures for all datasets.

Table 3.1 and Table 3.2 present a statistical overview of linguistic properties across the selected 18 datasets ordered by *Number of IGTs*, each labeled by their ISO 693-3 codes. This overview showcases the wide range of complexity and diversity across these datasets and variations in morphological and lexical characteristics that are essential for later computational analysis of interpreting the performance and outcomes of the AGGREGATION system's experiments.

ISO	Number of IGTs	Number of Distinct Stems	Number of Distinct Affix Types	Number of Distinct Word Types	Number of Grams
kre	178	132	33	271	30
ikx	183	237	24	474	42
esu	196	111	126	329	31
yak	229	209	159	618	39
nuk	649	472	323	1700	29
wbl	770	234	60	677	34
wmb	797	338	182	1083	53
lez	1531	1188	384	3618	49
bbl	1683	700	335	2562	42
mni	1785	931	377	5619	45
tdh	1850	663	211	2431	12
aqn	2204	489	246	1449	24
erk	2240	1219	122	5513	57
ybh	2346	1023	409	4539	65
qux	4633	2142	473	10602	89
tsz	5685	1139	331	5517	67
ctn	10779	2022	547	9519	79
yaq	10971	4348	503	11290	65

Table 3.1: Dataset Overview - Part 1

The *Number of IGTs* column reflects the total count of glossed examples available for each dataset, with figures ranging from 178 for Panara (kre) to a substantial 10,971 for Haiki (yaq). *Number of Distinct Stems* and *Number of Distinct Affix Types* offer insights into the morphological richness within each language's dataset, where, for instance, Haiki (yaq)

displays a high count with 4,348 stems and 503 affix types. Diversity in the fully inflected forms is captured under *Number of Distinct Word Types*. The *Number of Grams* indicates the variety of grammatical particles or units identified, which, in the case of Yaoyos Quechua (qux), amounts to 89.

ISO	Average IGT Length in Words	Average IGT Length in Morphemes	Type Stems Ratio	Affix Ambiguity Ratio	Allomorph Ratio
kre	5.90	6.04	1.297	1.234	1.333
ikx	3.21	3.45	1.062	1.652	1.146
esu	2.46	6.06	1.585	2.844	1.382
yak	6.58	10.68	1.658	1.642	1.315
nuk	6.55	12.27	1.954	5.547	1.362
wbl	4.18	7.48	1.860	0.858	1.165
wmb	3.68	5.65	1.858	4.361	1.354
lez	12.23	18.44	1.800	1.915	1.388
bbl	5.78	9.55	2.040	0.900	1.112
mni	8.90	19.30	3.054	0.998	1.475
tdh	5.00	7.79	2.027	0.942	1.276
aqn	3.99	5.17	1.754	2.910	1.183
erk	11.06	15.45	2.550	2.447	1.380
ybh	4.69	9.68	2.291	1.735	1.287
qux	4.42	9.77	2.412	2.578	1.510
tsz	3.55	6.69	2.539	2.042	1.544
ctn	4.48	7.53	2.512	3.467	1.574
yaq	7.10	8.43	1.442	1.191	3.690

Table 3.2: Dataset Overview - Part 2

The metrics *Average IGT Length in Words* and *Average IGT Length in Morphemes* provide an average measure of sentence complexity, with dataset like Lezgi (lez) averaging 12.23 words and a 18.44 morphemes per IGT.

The *Type Stems Ratio* quantifies the diversity of surface word forms found in the `words` tier relative to the total number of stem types in the dataset. In this context, word forms refer to fully inflected surface realizations found in the words tier, while stems represent canonical or base forms underlying those word forms. A higher ratio in a dataset suggests greater morphological variation, where individual stems occur in multiple inflected forms. A lower ratio implies that most stems appear with fewer variants, potentially indicating limited inflection or sparse annotation. For example, the Type-Stem Ratio for Meitei (mni) is 3.054, indicating that each base stem form is associated with approximately three distinct inflected word forms on average.

The *Allomorph Ratio* and *Affix Ambiguity* both quantify aspects of morphological complexity, but they focus on different relationships between form and meaning.

The *Affix Ambiguity* captures the degree of morphological variation associated with grammatical labels (grams) within each dataset. Specifically, it measures how many different affix spellings are used to express the set of grammatical functions observed in the data. A higher ratio indicates that each grammatical label tends to be realized by a greater number of distinct affix forms, suggesting a high degree of allomorphic variation. Conversely, a lower ratio suggests a more consistent and constrained mapping between grammatical categories and affix forms. For example, in the dataset for Nuuchahnulth (nuk), an *Affix Ambiguity* score of 5.547 indicates that each distinct grammatical label is, on average, realized by more than five different affix spellings.

The *Allomorph Ratio* quantifies the average number of allomorphs, or distinct morphemes that express the same grammatical meaning, associated with each gloss in a dataset. In this case, the same surface form appearing with different glosses is treated as multiple allomorphs, because each reflects a different grammatical function. This metric emphasizes how varied the realizations of a single gloss can be. A high *Allomorph Ratio* suggests that individual glosses

are realized by many different allomorphs, indicating rich inflectional variation or irregularity. A high *Affix Ambiguity*, on the other hand, suggests that affix forms are reused across different grammatical functions, reflecting greater form-meaning ambiguity or multifunctionality. Low values for either metric point to more consistent form-function mappings. For example, in the Haiki (yaq) dataset, the *Allomorph Ratio* is 3.69, meaning that, on average, each gloss is realized by more than three distinct allomorphs. This indicates a high variation in how grammatical meanings are expressed morphologically.

The enriched Xigt files were segregated into training and test datasets differently for each input group. While Input groups A and B were split into 90% training and 10% test datasets through a random selection process, Input C, a derivative of Input B with replaced POS tags, required a non-random approach to maintain alignment with its counterpart. Thus, the division was applied consistently across Input groups B and C using identical indices, where specific entries or positions from Input B were used to extract the 10% subset, which was then identically applied to Input C. This ensures that the test datasets for both Input B and C are directly comparable, sharing the same textual data but differing in the source of their POS tags.

A thorough examination of the data, supported by visual analysis from Figure 3.1, reveals a close similarity in the distribution of quantitative parameters between the 10% test datasets and the full datasets. The data is illustrated through 16 subplots arranged in a 4x4 grid, forming eight pairs of box plots. Each pair is designed to compare the distribution of a specific linguistic variable between the 10% test datasets and the full datasets. The first and third columns of subplots represent the 10% datasets, while the second and fourth columns depict the full datasets. Within each row, the pairings consist of one subplot from the 10% dataset and its adjacent subplot from the full dataset. The variables include the number of morphemes, stems, distinct word types, grams, and average lengths in both morphemes and words. The data is organized on the x-axis by ascending order of the *Number of IGTs*, from the smallest (10% kre = 17; all kre = 178) to the largest (10% yaq = 1097; all yaq = 10971). The comparability of these distributions affirms the representativeness of the

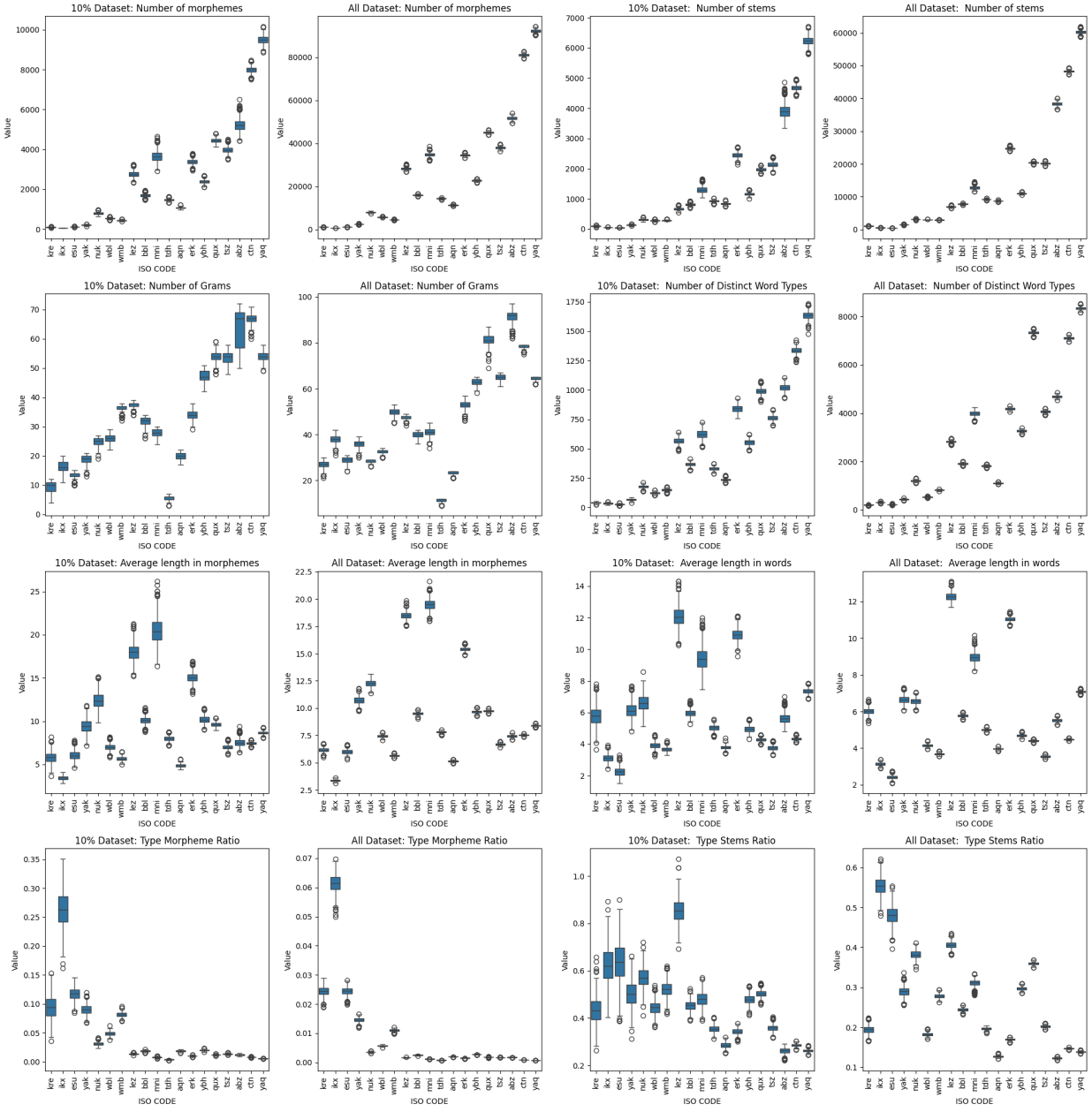


Figure 3.1: 10% vs. All Dataset Bootstrap with Resampling Variables Distribution

10% test data, indicating that it can be considered a reliable subset reflective of the larger dataset's characteristics.

3.4 Grammar Generation

This section outlines the experimental procedure for generating grammar specification files from enriched Xigt training sets using the AGGREGATION inference system. To investigate how dataset characteristics influence grammar quality, the system was tested across a wide variety of input conditions by generating grammars from randomly sampled IGT subsets.

3.4.1 3000 Random Samples

The experimental design involved generating 3000 randomly sampled datasets for each training corpus. This was done to explore a broad space of input conditions and to capture variability across multiple dataset configurations. The choice of 3000 runs reflects a practical compromise: fewer samples would risk underrepresenting the diversity of input variables and amplifying the impact of outliers, while more would significantly increase computational cost, as each grammar takes approximately 2 to 60 minutes to infer. With 3000 samples, I can achieve sufficient statistical coverage while keeping the overall workload manageable. All training samples were drawn using bootstrap sampling with replacement. This approach is especially important for smaller datasets, as it enables the creation of many distinct training sets by allowing repeated or omitted IGTs in each sample and simulates different subspaces of the input data.

Figure 3.2 presents boxplots of key input variables across 3000 random samples per dataset, including the variation in *Number of IGTs*, number of grams, and average lengths in words and morphemes. The smooth and continuous progression of medians across datasets with the wide interquartile ranges demonstrates that the sampling approach effectively spans a representative and diverse range of input conditions. This confirms that the 3000-sample design is sufficient to support a robust analysis of how input characteristics influence grammar generation outcomes.

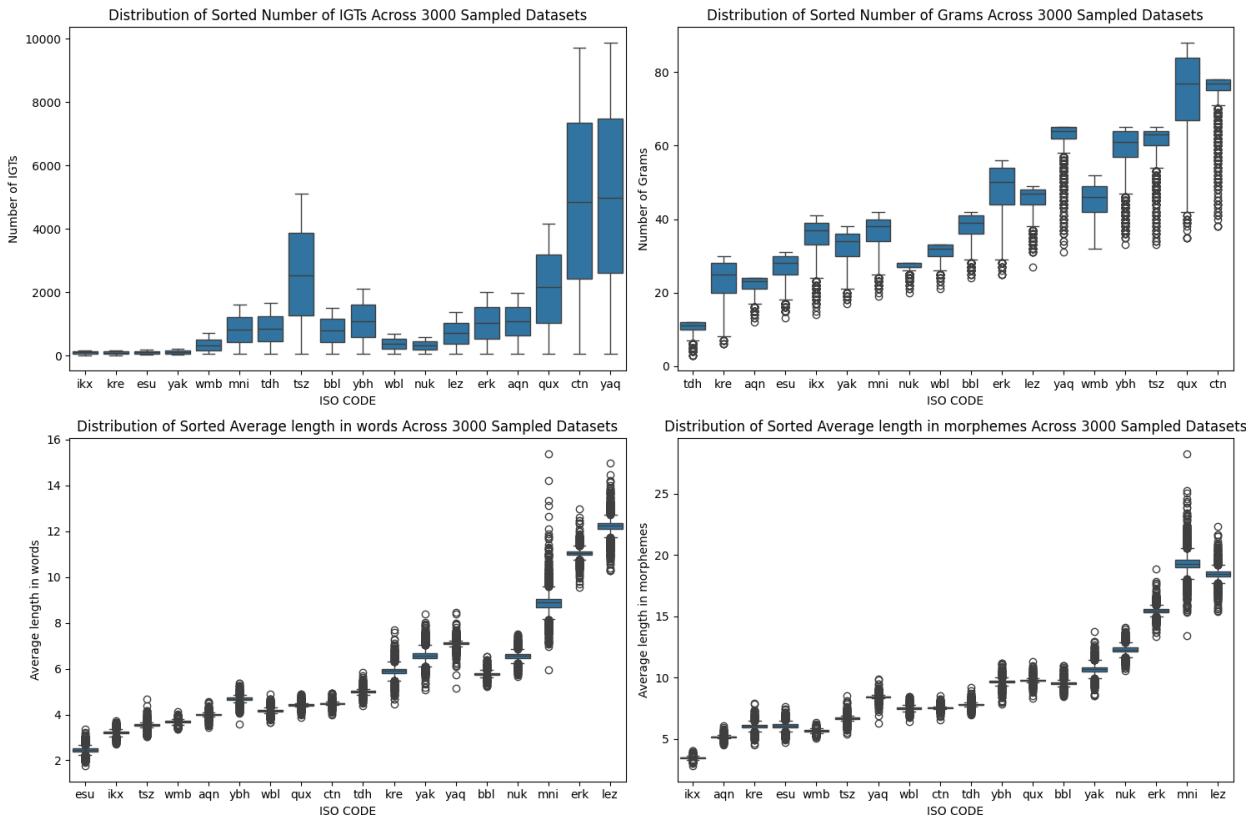


Figure 3.2: Distribution of Sorted Input Variables Across 3000 Sampled Datasets

Why not systematically test all possible combinations of input parameters, in a fashion similar to a grid search in machine learning? While this might seem like a straightforward way to identify the most effective input conditions, it misunderstands the nature of the research question. The goal here is not to find the single "best" subset of IGTs, but rather to understand how general properties of a dataset, such as size, morphological complexity, and lexical diversity, affect the quality of the resulting grammar. To do this, it is more useful to observe how these factors vary across many plausible samples than to exhaustively evaluate every possible combination. Moreover, given the time required to generate a grammar from even a single sample (ranging from 2 to 60 minutes), a fully exhaustive strategy would be computationally prohibitive. A randomized sampling approach offers a more practical and informative alternative.

The grammar generation experiments involve creating many training subsets by randomly sampling a number x of IGTs from each full dataset. x is chosen independently for each sample run. To avoid generating grammars from datasets that are too small to be informative, x is constrained to fall between a dynamic lower bound and the dataset's full size. Specifically, the lower bound is set to the smaller of 10% of the dataset size or 10 IGTs:

$$\min\left(10, \lfloor 0.1 \times \text{size of dataset} \rfloor\right) \leq x \leq \text{size of dataset} \tag{3.1}$$

3.4.2 AGGREGATION Parameters

The AGGREGATION inference system offers a set of parameters that can be fine-tuned to optimize grammar inference from linguistic data. These parameters allow for customization of the inference process. For this study, a standardized set of parameter values was selected based on insights from previous research. However, it is important to admit that some of these parameters are factors in answering the research question, but because of the scope of this study, I kept these as constant values and only considered variables in the linguistic corpora. Table 3.3 is a summary of the AGGREGATION system parameters and the values for this experiment. To preserve meaningful variation in output metric, all grammars were

generated with `compression_rounds` set to 1. This setting limits the amount of class merging during the inference process. Higher compression levels would reduce the number of inferred categories, such as position classes, lexical rule types, or feature combinations, resulting in simpler grammars.

Table 3.3: Configuration Parameters

Parameter	**Value**	**Description**
`iso_code`	`"example"`	Language ISO code.
`xigt_path`	`"/path/to/xigt/data/"`	Path to the enriched Xigt input data.
`agg_config_path`	`"/path/to/config/"`	Directory path for AGGREGATION configuration files.
`inference`	`true`	Enables the inference process to generate grammars.
`compression_rounds`	`1`	Number of rounds for compression, affecting class merging.
`output_dir`	`"/path/to/output/"`	Directory where the output files will be stored.
`graph`	`false`	Disables graph output, which visualizes the grammar.
`hyphens`	`true`	Treats hyphens as valid characters within words, affecting tokenization.
`precluster`	`None`	Disables preclustering of data before inference.
`glosses`	`true`	Enables the use of glosses in the inference process.
`compression`	`0.2`	Sets the threshold for merging classes based on overlap.

Continued on next page

Parameter	Value	Description
lexitem_classes	true	Enables classification of lexical items into distinct classes.
all_stems_occur_bare	true	Assumes all stems can occur without affixes.
ignore_chars	None	None of the characters are ignored during analysis.
ungrammatical	"*!?"	Marks characters used to indicate ungrammatical data.
allomorphs	None	Disables the consideration of allomorphs in the analysis.
boundaries	true	Presence of explicit word boundaries.
allow_difference	5	Sets the maximum character-level difference allowed between the concatenation of morphemes and the surface word form during alignment validation.
escape_special_characters	true	Ensures special characters are properly handled
wordlist	''	Specifies a wordlist for filtering which word forms are included in the generated choices file (no filtering by default).
infer_case	gram	Directs the system to infer case marking from grammatical markers.

3.5 Evaluation Workflow

In this section, I describe the multi-stage evaluation workflow designed to analyze how structural features of input datasets influence the quality of grammars produced by the AGGREGATION system. The overall analysis proceeds in a sequence of structured steps, combining statistical modeling, model-informed filtering, and targeted visualization.

I begin by compiling grammars and extracting output metrics and structural statistics for each dataset (Section 3.5.1). To explore predictive structure, I train XGBoost models to rank the relative importance of input features (Section 3.5.2), and assess whether certain output metrics reflect test-set artifacts through correlation analysis (Section 3.5.3). I then compare XGBoost-derived feature rankings with empirical correlations to detect overfitting and guide metric selection (Section 3.5.4).

Then, I jointly apply scatterplot-based visual diagnostics and single-predictor mixed-effects models to examine the shape, consistency, and marginal effects of X-Y relationships across datasets (Section 3.5.5), and extend these models with linguistically motivated interaction terms to capture non-additive behavior (Section 3.5.6). Finally, I evaluate the effect of POS annotation source via paired comparison on matched datasets (Section 3.5.7).

3.5.1 Grammar Compilation and Metric Extraction

For each sample, I run the AGGREGATION system to generate a corresponding `choices` file, which specifies lexical types, morphological rules, and syntactic options. I compile these files using the LinGO Grammar Matrix customization system, which constructs machine-readable grammars in HPSG formalism.

I evaluate each compiled grammar using the Answer Constraint Engine (ACE; Crysmann and Packard (2012)) by parsing a held-out test portion comprising 10% of the original IGT data from the same dataset. I log the parsing results using the Automated Regression Testing (ART)[2] framework, which generates a TSDB profile containing coverage and ambiguity

[2]https://sweaglesw.org/linguistics/libtsdb/art.html

metrics.

In parallel, I run a custom script to extract structural statistics from the `choices` files, which reflects the quality and complexity of the generated grammar. These statistics include the number of verb types, number of morphological position classes, and the number of lexical rule types associated with each position class. Together, the TSDB-derived quality metrics and the structure-derived complexity indicators form the complete feature set for the modeling and analysis steps described in subsequent sections.

3.5.2 Feature Importance Modeling

To identify which structural features of the input datasets most strongly influence specific output metrics, I trained a separate gradient-boosted regression model (XGBoost; Chen and Guestrin 2016) for each of the output metrics. In this setup, I aim to generate a ranked importance distribution over the 10 structural input variables for cross-metric comparisons and identification of predictors with consistent influence across quality dimensions.

XGBoost is a tree-based ensemble learning algorithm that constructs a sequence of decision trees, with each tree trained to reduce the residual errors left by the previous ones. The model selects split points that minimize a loss function (mean squared error) based on both the feature value and its context within the other predictors. As such, feature importance in XGBoost is inherently context-sensitive: a variable may have low marginal correlation with the outcome, and be consistently useful in interaction with other variables. This makes XGBoost suitable for my setting, where structural properties such as affix diversity and morpheme length are expected to have non-additive effects on output metrics.

For each model, I used the `XGBRegressor` implementation with the following parameters: a maximum tree depth of 6, 100 boosting rounds, a learning rate of 0.1, and the `hist` tree method for histogram-based training. All models were executed on GPU to accelerate computation across multiple output dimensions. I split each dataset into 80% training and 20% test partitions.

Feature importance scores were extracted using the gain-based metric, which quantifies

the total contribution of each feature to loss reduction over all decision splits. I normalized the gain values so that the importances for each model sum to 1. The resulting importance matrix, indexed by input variables and output metrics, captures which predictors are quantitatively more influential than others under the XGBoost modeling framework regardless of whether the relationship is linear or non-linear.

In later stages, I use these importance rankings to guide the selection of structural predictors for more interpretable mixed-effects models. While XGBoost itself is not explanatory in the statistical sense, it has a complementary role in flagging non-linear or interaction-based patterns that may not surface in traditional regression analysis. In this way, feature importance modeling serves both as a standalone diagnostic and as a heuristic input to the further modeling pipeline.

3.5.3 Correlation Analysis

To assess whether grammar performance metrics are influenced by structural properties of the evaluation data (i.e., the test set), rather than reflecting generalization from the training distribution, I conducted a Pearson correlation analysis (Joseph Lee Rodgers, 1988) between test-set structural features and average output metrics. The primary goal of this analysis is to identify grammar output metrics (*Y variables*) that may be sensitive to test-set artifacts, in order to exclude them from subsequent modeling stages focused on training data structure.

Ideally, generated output metrics should reflect the inference power of the AGGREGATION system given training data structure. However, if these metrics correlate strongly with features of the test set, they may be capturing surface-level parsing ease rather than generalization quality. For example, grammars may appear more accurate simply because the test set contains shorter or structurally simpler sentences.

To mitigate this risk, I compute test-set averages for each of the ten structural input features (*X variables*). For the same 25 datasets, I also calculate the average value of each grammar output metric (*Y variable*) by aggregating its results over 3000 grammar runs trained on randomized subsets of the 90% training partition.

This produces a 25-row dataset, where each row corresponds to a dataset, and contains both the average test-set features and the corresponding grammar performance outcomes. I then compute Pearson correlation coefficients between all pairs of X and Y variables. Metrics that show strong correlation with any test-set structural feature are flagged as potentially *artifact-prone*, and are excluded from downstream modeling stages. In contrast, metrics that are weakly or inconsistently correlated with test-set structure are retained, as they are more likely to reflect structural properties learned from the training data.

3.5.4 Rank Consistency Evaluation

The previous two sections present two complementary approaches to understanding which input features most influence grammar output metrics. However, these two analyses are not directly comparable in magnitude or scale. XGBoost feature importance captures the extent to which a feature contributes to prediction through potentially complex, non-linear decision splits. In contrast, Pearson correlation only reflects linear associations between test-set features and grammar performance outcomes.

Despite these methodological differences, comparing the two can offer insights into the source of observed variation in evaluation results. If the same features are ranked highly by both analyses, this suggests that output metrics may be driven by properties of the test set rather than generalizable patterns inferred from training data. Conversely, divergence between the two rankings would indicate that the XGBoost model's behavior is influenced by structural patterns learned from the training data, not simply by characteristics of the test sentences.

To quantify this alignment, I compute Spearman rank correlation (ρ) between the XGBoost-derived feature ranking and the test-set correlation ranking for each output metric. This scale-independent comparison reveals which grammar output metrics may be overly sensitive to test-set artifacts (high ρ) and which reflect deeper generalization patterns (low ρ). These insights help inform the selection of reliable output metrics for downstream modeling.

Based on this criterion, I select a single representative output metric from each perfor-

mance dimension (coverage, ambiguity, grammar complexity) for use in downstream statistical modeling. Preference is given to metrics that exhibit weak or non-significant correlations with test-set features, and low agreement in rank ordering with test-set-driven results. For the efficiency dimension, only one metric, *Inference Time*, is available. I therefore include it as the representative indicator of inference efficiency.

3.5.5 Marginal Effects and Dataset-Level Visualization

To assess the individual influence of structural predictors on grammar output metrics, I jointly applied two complementary techniques: univariate linear mixed-effects modeling and dataset-level scatterplot visualization. Mixed-effects models are statistical models designed to handle data with grouped or hierarchical structure, such as observations drawn from multiple datasets. In the present context, each grammar evaluation is linked to a specific dataset, and these datasets differ in many uncontrolled ways: typological profile, size, annotation quality, etc. A standard linear regression would treat all observations as independent, which risks inflating the apparent effect of structural features that happen to correlate with certain high-performing datasets.

To address this, I use a linear mixed-effects model, which includes both *fixed effects* (shared across all datasets) and *random effects* (specific to each dataset).[3] The fixed effect estimates the global influence of a given structural variable X_i on an output metric Y, while the random effect accounts for dataset-level baseline shifts. In practical terms, this allows me to ask: "Does X_i tend to influence Y in the same direction, even after correcting for dataset-specific properties?" Another question to ask is: "Which structural features show strong enough marginal effects to warrant inclusion in more complex multivariate models

[3]ANCOVA (Fisher, 1992) can include dataset identity as a fixed factor to control for baseline differences, but this treatment assumes that the datasets used represent the entire population of interest. That is, the ANCOVA model estimates a separate effect for each dataset without assuming they are drawn from a broader distribution. As a result, the inferences apply only to the observed datasets, and not to any potential new or unseen datasets. In contrast, linear mixed-effects models treat dataset as a random effect, modeling it as a random draw from a wider population of low-resource languages. This allows for generalization beyond the specific samples used, and better reflects the cross-linguistic scope of this study.

later on?"

Technically, the fixed-effect component estimates a single regression slope β_i for the predictor X_i, assuming this effect holds across all datasets. Meanwhile, the random-effect component introduces a dataset-specific intercept term u_j for each dataset j, drawn from a normal distribution: $u_j \sim \mathcal{N}(0, \sigma_{\mathrm{u}}^2)$. This allows the model to account for systematic baseline differences between datasets.

The overall model can be expressed as:

$$Y_{ij} = \beta_0 + \beta_i X_{ij} + u_j + \varepsilon_{ij} \tag{3.2}$$

where Y_{ij} is the output metric for the ith observation in the jth dataset, u_j is the random intercept for dataset j, and ε_{ij} is the residual error term.

For each of the selected output metrics, I tested the influence of each structural predictor X_i using a fixed-effect specification, with a random intercept for each dataset, identified by ISO code. This model structure accounts for variation in baseline performance across datasets while estimating whether each predictor contributes meaningfully to differences in grammar quality.

Formally, for each X_i, I tested the following hypotheses:

$$\begin{aligned} \textbf{Null Hypothesis } (H_0): &\quad \beta_i = 0 \quad \text{(No effect after accounting for dataset)} \\ \textbf{Alternative Hypothesis } (H_1): &\quad \beta_i \neq 0 \end{aligned} \tag{3.3}$$

I standardized all variables prior to estimation in order to compute comparable coefficients, and also reported raw (unstandardized) coefficients to preserve interpretability in the original scale. These results identify which input features show linear associations with output metrics while accounting for dataset-level variance.

In parallel, I constructed grids of scatterplots to visualize the relationship between structural predictors and output metrics across individual datasets. This step serves both as an exploratory diagnostic and a preparatory filter for downstream modeling. Each grid focuses

on a single output metric (Y), with rows indicating individual datasets and columns representing top-ranked input features (X), ordered left to right by their marginal effect size from single-predictor mixed-effects models. Each cell contains a bivariate scatterplot showing the X-Y relationship within one dataset.

This visualization enables several layers of interpretation. First, it reveals cross-dataset consistency: I can assess whether a given X-Y relationship holds across most datasets, or whether it is highly variable and specific to individual datasets. Second, it makes visible the shape of effect to help me detect whether structural features have roughly linear influences on Y variables or exhibit more complex patterns.

Third, the grids support outlier detection by highlighting datasets that deviate strongly from the overall trend. Such deviations may reflect structural divergence, annotation inconsistencies, or dataset-specific noise. Finally, the scatterplots offer initial clues about potential interaction signals, where the effect of one input feature appears to depend on the level of another.

Because the univariate models assume linearity and additivity, they do not capture interaction effects, threshold behaviors, or non-monotonic trends observed in the earlier visualization phase, to address these limitations, I extend the analysis in the next subsection through multivariate modeling with selected interaction terms, supported by targeted visualizations that reveal complex variable interplay.

3.5.6 Multivariate Modeling with Interaction Terms

To address limitations of purely additive modeling and to capture potential interactions between structural features, I extend the linear mixed-effects framework to include selected multivariate combinations. This step is not intended to exhaustively test all possible feature pairs, but to investigate theoretically plausible and empirically promising interactions.

I begin by selecting candidate feature combinations based on two main criteria: (i) linguistic interpretability, and (ii) empirical relevance. Rather than mechanically enumerating all feature pairs, I focus on combinations that are meaningful from a linguistic standpoint.

For example, datasets with high values of *Average IGT Length in Morphemes* can exhibit agglutinative morphology, encoding rich grammatical information within words. If such datasets also show high *Affix Ambiguity*, this suggests that affix forms are reused across multiple grammatical functions, increasing morphological ambiguity. In contrast, a combination of high *Average IGT Length in Morphemes* and low *Affix Ambiguity* typically reflects a more transparent morphological system, where each affix maps consistently to a single grammatical meaning. Second, I assess empirical relevance by drawing on results from earlier stages. Specifically, I prioritize variables that show strong marginal effects in the univariate mixed-effects models.

After selecting candidate interactions, I fit extended mixed-effects models that include both the main effects and their corresponding interaction terms. As in the univariate case, each model includes a random intercept for dataset identity to control for baseline variation across datasets. The difference is that I now include two fixed-effect predictors and their interaction term to test whether the effect of one variable depends on the level of another.

The overall model can be expressed as:

$$Y_{ij} = \beta_0 + \beta_1 X_{1,ij} + \beta_2 X_{2,ij} + \beta_3 (X_{1,ij} \times X_{2,ij}) + u_j + \varepsilon_{ij} \tag{3.4}$$

where Y_{ij} is the output metric for the ith observation in the jth dataset, $X_{1,ij}$ and $X_{2,ij}$ are standardized structural predictors, and $(X_{1,ij} \times X_{2,ij})$ represents their interaction. As before, u_j is the random intercept for dataset j (with $u_j \sim \mathcal{N}(0, \sigma_u^2)$), and ε_{ij} is the residual error.

To evaluate whether the added complexity is justified, I compare model fit using the Akaike Information Criterion (AIC; (Akaike, 1974)) and the Bayesian Information Criterion (BIC; (Schwarz, 1978)). Both are penalized-likelihood metrics that balance model fit against model complexity: AIC emphasizes predictive accuracy, while BIC penalizes complexity more strictly, favoring parsimonious explanations. If the inclusion of interaction terms leads to lower AIC/BIC values, this suggests a better tradeoff between explanatory power and model simplicity.

Once model selection is complete, I interpret the resulting multivariate models using

interaction visualizations, which help illustrate how the influence of one structural variable depends on the value of another. It also serves as a diagnostic tool to show whether certain structural feature combinations amplify or suppress each other's effects on output metric, and whether such patterns generalize across the datasets.

3.5.7 POS Annotation Comparison

To evaluate how part-of-speech (POS) tag source affects AGGREGATION grammar quality, I conducted a paired t-test analysis (Ruxton, 2006) comparing grammars trained on manual POS annotations (Group B) with those trained on automatic POS tags produced by the INTENT system (Group C). In contrast to previous analyses that focus on structural properties of the datasets, because POS source is not itself a structural feature, I constructed matched dataset pairs for a controlled comparison. This experiment isolates the impact of annotation quality by holding all other variables constant, where each pair shares the same underlying IGT data, training/test splits, and sentence content.

To measure the impact of POS type, I computed the difference in evaluation scores between the two versions for each sample:

$$\Delta = \text{Score}_{\text{manual POS}} - \text{Score}_{\text{intent POS}} \tag{3.5}$$

I applied this delta computation to the grammar quality metrics identified from Rank Consistency Evaluation process. I excluded *Inference Time* from this analysis because it does not directly depend on POS tagging. According to the results that I will present in the Chapter 5, *Inference Time* is primarily shaped by factors such as sentence length, grammar ambiguity, and lexical diversity. These are the elements determined by the dataset's structural content rather than its annotation source.

3.6 Case Study: Meitei Language Application

To complement the large-scale evaluation of AGGREGATION performance across datasets, I conducted a case study on the Meitei (mni) dataset to evaluate the practical usability of AGGREGATION-generated grammars. The methodology of this case study aims to simulate a realistic grammar engineering scenario in which a linguist must refine an automatically generated grammar. The goals of this study are twofold: (1) to understand the types of structural noise and errors introduced by AGGREGATION and (2) to assess how input characteristics affect downstream usability and refinement effort.

3.6.1 Sample Selection

I conducted a focused case study using six grammar samples. These were selected from the broader 3000-run sampling experiment to reflect variation across two main factors: the *Number of IGTs* used in training and the resulting grammar quality. The six samples include three from Input Group B (using linguist-provided POS tags) and three from Input Group C (using INTENT-generated POS tags), forming three matched pairs for direct comparison.

Each pair consists of grammars trained on the same IGT subset, differing only in the POS tagging method. The selected pairs are:

- **FullSet-ManualPOS** and **FullSet-IntentPOS**: Trained on the complete set of 1596 IGTs in the Meitei dataset, representing maximal available data.

- **BestCoverage-ManualPOS** and **BestCoverage-IntentPOS**: Selected for optimal parsing performance, achieving the highest coverage and lowest ambiguity within their respective POS tagging conditions.

- **SmallMaxCoverage-ManualPOS** and **SmallMaxCoverage-IntentPOS**: Trained on a minimal input size of 295 IGTs; among all such low-data samples, these achieved the highest parsing success.

3.6.2 Evaluation Framework

The analysis combined both quantitative and qualitative methods. Quantitatively, I measured parsing performance using ACE and ART by capturing metrics defined in Section 3.1.2. Qualitatively, each grammar was examined for structural correctness and linguistic plausibility. I focused on common sources of noise and error, including:

- Affix misclassification across lexical categories.
- Incorrect categorization of case markers.
- Misidentification of determiners and quantifiers.
- Incomplete or inaccurate representation of person-number distinctions in pronouns.
- Incorrect lexical category assignments that lead to phrase structure errors (e.g., verb phrases being analyzed as noun phrases).

Each of these areas was manually evaluated by inspecting the contents of the grammar's `choices` file and the outputs of the parser on targeted test sentences. The cleaning effort required for each sample was also documented to assess practical usability.

3.7 Summary

In this chapter, I have outlined the methodological framework used to investigate how different data preparation strategies affect the performance of the AGGREGATION system. The approach combines both quantitative and qualitative analysis to examine the relationship between linguistic input features and grammar generation outcomes. I selected 18 datasets from a broader data repository, applied consistent preprocessing steps, and created enriched and non-enriched versions to compare the impact of POS tag sources.

To evaluate the influence of input characteristics, I used randomized sampling and trained grammars under standardized conditions. Metrics such as coverage, ambiguity, and complexity were extracted using ACE and ART, while XGBoost models and mixed-effects regression were used to identify key predictors and interaction effects. A paired t-test analysis allowed me to compare grammar quality between manual and automatically generated POS tags. Finally, I conducted a case study on the Meitei language to explore how these findings apply in a practical grammar development scenario.

To operationalize this evaluation at scale, I developed a dedicated experimental pipeline capable of managing thousands of grammar inferences and evaluations across multiple datasets. The following chapter describes the design and implementation of this pipeline.

Chapter 4

EXPERIMENTAL PIPELINE IMPLEMENTATION

In this chapter, I present the design and implementation of a scalable and reproducible experimental pipeline for evaluating the AGGREGATION grammar inference system. As I investigate how dataset properties affect the quality of AGGREGATION-generated grammars, conducting large-scale experiments across thousands of data samples and multiple datesets needs substantial engineering support. The pipeline development addressed key challenges in automation, parallelization, storage management, and evaluation, and evolved through multiple iterations, from the original AGGREGATION scripts `agg-startup`,[1] to an enhanced Python wrapper `agg-script`,[2] to a fully modular Java-based framework called `AGGFlow`.[3]

I start with reviewing limitations of the original AGGREGATION pipeline (Section 4.1) and how `agg-script` improved usability and execution efficiency (Section 4.2). I then describe the design goals and system architecture of `AGGFlow` (Section 4.3) with the responsibilities of each core module and the engineering decisions made to support the experiments in this thesis.

4.1 Challenges with Original AGGREGATION Pipeline

The original AGGREGATION pipeline, `agg-startup`, faced several challenges that hindered efficient experimentation. A significant challenge was the component separation, including data preparation, grammar inference, and evaluation, requiring manual invocation of each

[1] https://git.ling.washington.edu/agg/agg-startup

[2] https://git.ling.washington.edu/agg/agg-scripts

[3] https://git.ling.washington.edu/agg/agg-scripts/-/tree/ltxom-thesis

component and configuration for various data formats (Toolbox, ELAN, Flex, etc.). Another challenge was that the original pipeline's linear/single processing setup was unsuitable for the larger scale of this research due to excessive processing times.

4.2 Improvements with AGG-Script

To resolve the challenges stated in the previous section, before conducting this research, I worked on the development of `agg-script`, a Python-based pipeline that I refactored to include several key enhancements: a dependencies manager, concurrent processing, systematic modularization, and a flexible configuration system. It utilizes AnaConda for virtual environment management, installs packages from both standard repositories and up-to-date GitHub sources for AGGREGATION, Grammar Matrix, Xigt, INTENT, and MOM, and manages Linux executables such as ART and ACE along with their environment variables.

One of the key improvements in `agg-script` is its modular design, which separates the pipeline into discrete components for Data Preparation, AGGREGATION Inference, and Parsing. Each module is independently configurable via a YAML file, which allows different types of linguist-users to work independently on specific parts of the pipeline.

However, to meet the full scope of this thesis, particularly the need to generate and evaluate thousands of grammars across 25 datasets, perform structured sampling, aggregate and compress outputs, and manage disk usage, `agg-script` is insufficient. It lacked critical functionalities such as fine-grained data sampling, experiment resumption, integrated file management for compressed outputs, and automated evaluation using test suites. These requirements motivated the design of a more robust architecture: `AGGFlow`, a Java-based pipeline built on the principles of modularity, scalability, and reproducibility to support the complex experimental framework of this thesis.

4.3 Experiment Architecture

`AGGFlow`, based on `agg-script`, is implemented in Java and adheres to Object-Oriented Programming principles, packaging IGT and related structures into Java Beans for reasons

such as encapsulation, modularity, and reusability. It consists of AGGREGATION Manager, Choices Combinator, Choices Parser, and ACE Runner components.

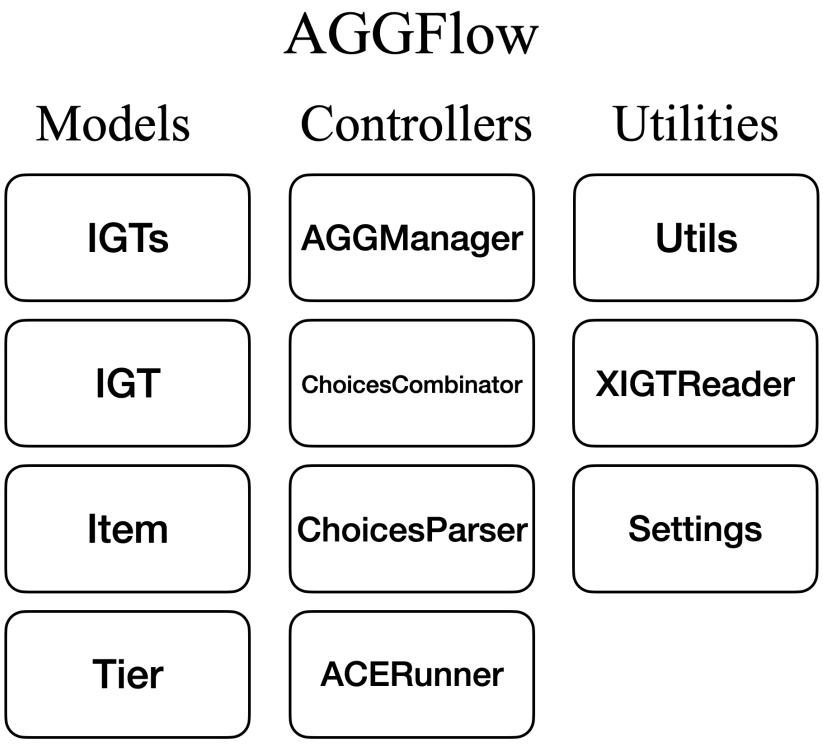


Figure 4.1: Overview of the Experiment Architecture.

Figure 4.1 presents an overview of the modules in `AGGFlow`. At the core of the system are Java Bean classes `IGT` and `IGTs`, which represent an individual interlinear glossed text and a collection of such texts, respectively. Each `IGT` instance contains a list of `Tier` objects, where each `Tier` represents a specific annotation layer in the Xigt format, such as words, morphemes, glosses, or syntactic annotations.

Within each `Tier`, a series of `Item` objects store the actual annotation content, alongside attributes such as alignment, segmentation, part-of-speech (POS), and grammatical relations for fine-grained and structured access to every layer of linguistic annotation. For instance,

tiers can encode a mapping between glosses and morphemes, while items may store additional metadata like dependency information. The `Tier` and `Item` classes closely mirror the Xigt data model.

The `IGT` class includes key methods such as `initialization()` for computing statistics (e.g., number of stems and morphemes) and building gloss-based feature lists, and `toXIGT()` for exporting annotated data in Xigt format. These representations and methods provide the foundation for data sampling, grammar inference, and performance evaluation throughout the experimental pipeline.

4.3.1 AGGREGATION Manager

The AGGREGATION Manager is responsible for configuration management, data processing, AGGREGATION inference system execution, and experiment execution control, including task queue management and execution. It also offers platform-specific execution handling, post-processing, output management, and the ability to resume experiments from breakpoints. The file management system, given limited storage resources, optimizes file handling to avoid redundant data and quickly reproduce datasets with minimal storage footprint. This principle continues in the subsequent modules.

AGGREGATION Manager supports two experimental modes: random sampling and preloading based on previously sampled IGT sets. Random sampling is used for the experiments on Input A and B groups, while preloading is used for Input C group to maintain consistency across samples.

Parameters such as system selection, maximum threads, number of experiments/samples, and options to delete training data files after each fold inference are configurable. The system supports Windows or Linux, with maximum threads adjusted according to CPU constraints. The number of experiments/samples parameter is used to determine the suitable sample size. For this thesis, as discussed in the previous chapter, it is set to 3000 samples. The training datasets deletion option is an optimization for storage space management; enabling this option does not mean the training sets are irrecoverable, as the training set IGTs are

saved as index sets for the original training files to be reconstructed through the IGTs class's *recoverIGTsFromResults* method.

Upon running the AGGREGATION Manager with specified parameters, a thread pool is created, consisting of the specified maximum threads plus the main thread. The main thread generates a queue of all tasks with their requirements and parameters, then distributes and monitors each thread's work until the queue is emptied, and all tasks are completed by the AGGREGATION Manager.

Each thread automatically generates the necessary configuration files for AGGREGATION, dynamically combining fixed parameters from a template file with dynamic parameters related to input/output file paths. This file path management is facilitated by a file management system that assigns a fold number to each sampling, such as "aqn233" for the 233rd sampling of the "aqn" dataset, to manage specific configuration parameters as discussed in Section 3.3.2.

4.3.2 Choices Combinator

The Choices Combinator is responsible for combining grammar specification files into a single compact database to facilitate efficient disk operations. Since the AGGREGATION Manager creats one grammar specification file per run, and there are 25 datasets and 3000 samples for each dataset, it leads to 75000 files used by subsequent grammar generation and evaluation modules. To improve its efficiency, as part of the file management system, the Choices Combinator organizes and packages each dataset's 3000 grammar specification files into one compacted choice file for ease of access in later tasks.

4.3.3 Choices Parser and ACE & ART Runner

The Choices Parser validates and generates grammars from the compacted choice files, mirroring functionalities supported by the AGGREGATION Manager. The ACE & ART Runner parses test suites using the generated grammars. Both modules implement similar execution handling, post-processing, output management, and breakpoint resumption capabilities

as the AGGREGATION Manager. Notably, since ACE and ART software is compiled for Linux systems, they are executed on Windows through calls to the Windows Subsystem for Linux (WSL), enabling cross-platform functionality.

4.3.4 Utilities

The utilities classes, `Utils`, `XIGTReader` and `Settings`, serve helper methods within the `AGGFlow` system for parsing linguistic data and managing configuration. The `XIGTReader` class reads Xigt-formatted XML files, converts each `<igt>` element into a structured `IGT` object and encapsulates tiers such as words, morphemes, and glosses. It ensures only valid IGTs are retained via an initialization check and returns the full set as an `IGTs` object for downstream processing. The `Settings` class provides configurable constants for tier IDs (e.g., `w` for words, `m` for morphemes), affix separators, and ANSI color codes for console output.

4.3.5 Configuration

The experiments were conducted on the datasets listed in Table 2.1. In the actual experiment running environment, datasets suffixed with an "N" (e.g., `mniN`, `ybhN`) to indicate their classification as the input group C variants. For each dataset, a single 90%/10% train/test split was created from the IGTs, with 10% of the data held fixed as the evaluation test set. The remaining 90% was used for 3000 experimental runs, each involving a random sample (with replacement) drawn from the training set.

Each AGGREGATION training run was configured with a consistent set of parameters. A compression ratio of 0.2 was applied, as specified in the `AGGManager` configuration. All experiments were run on a Windows platform using the Windows Subsystem for Linux (WSL), which helped the execution of ACE and ART for parsing and evaluation tasks. To optimize system performance given the hardware constraints, a 13^{th} Gen Intel Core i5-13500 processor, 64 GB of RAM, and data stored on a Samsung 870 SSD, a thread limit of six concurrent threads was enforced. To conserve storage space, training data

was deleted after each inference; however, the original IGTs remained recoverable via the `recoverIGTsFromResults` method in the `IGTs` class. For data handling, random sampling was employed for Input A and B datasets, while Input C datasets utilized preloaded IGT mappings from the `AGG-IGT-Mapping` directory to maintain consistency across runs.

The `AGGManager` dynamically generated configuration files for each run, based on a common template that included ISO-specific parameters as well as fold-specific information. The fold IDs were named following the pattern `isoFoldID`, such as `mni203`, for tracking of individual experimental runs.

For each run, the subset of IGTs designated for training was serialized into Xigt XML format and processed by the AGGREGATION system. This stage generated a grammar specification file called `latest_choices`. All configuration parameters, such as file paths and system settings, were dynamically generated and validated using Java-based helper methods in the `AGGManager`.

After grammar inference, the `ChoicesCombinator` was used to merge all generated `latest_choices` files into a compacted file per dataset. The `ChoicesParser` then parsed the combined file and generated customized grammars using the Grammar Matrix customization system (`matrix.py`). This step resulted in TDL-based grammar files for each experimental fold, which were prepared for the next stage in the pipeline.

The `ACERunner` component executed ACE and ART through WSL for parsing the grammars against the evaluation test suite. Test suites were automatically generated from 10% of the original IGTs using the `Main.java:createTestSuites` method. After each complete execution of a training sample (covering conversion to Xigt, grammar inference via AGGREGATION, grammar creation the Grammar Matrix, and evaluation using ACE and ART), a marker file named `OK` was written to the output directory. This flag indicated that the run had successfully finished. In cases where the full set of 3000 runs could not be completed in a single session, the presence of this file allowed the system to skip already completed runs upon restarting to avoid redundant re-execution.

Throughout the pipeline execution, the system utilized thread-safe queues and a moni-

toring thread to efficiently manage fold-specific output directories and ensure that resources were not exceeded. A dedicated monitoring thread ensured that tasks were completed before the next steps began to prevent overuse of system resources.

After each experimental run, intermediate large files (such as `data.dat`) were removed to conserve disk space, while the results were archived in well-organized directories for easy retrieval and further analysis. Each output was saved in a structured directory format to help downstream evaluation and profiling.

4.4 Summary

In this chapter, I described the design and implementation of a modular, scalable experimental pipeline developed to support the large-scale evaluation of AGGREGATION-based grammar generation. Starting from the limitations of the original `agg-startup` scripts, I outlined how `agg-script` improved usability and modularity, and why further requirements led to the development of `AGGFlow`. The architecture of `AGGFlow` addresses key needs including sampling, automation, concurrent execution, storage optimization, and evaluation tracking across 75,000 experimental runs.

Core components such as the AGGREGATION Manager, Choices Combinator, Choices Parser, and ACE Runner were described in detail, highlighting how they interact to streamline grammar inference and evaluation. Additional utility classes and configuration strategies ensure that the system remains reproducible, efficient, and adaptable to diverse datasets and experimental conditions.

With this infrastructure in place, I now turn to the results of these experiments. In the next chapter, I analyze the outputs generated by the pipeline to assess which linguistic input features most influence grammar performance, using a combination of feature importance modeling, correlation analysis, mixed-effects modeling, and a detailed case study.

Chapter 5

EXPERIMENTAL RESULTS

In this chapter, I report the results of the evaluation workflow described in Section 3.5, which was conducted across 25 datasets with 3,000 samples each, totaling 75,000 individual grammar generation and parsing runs. I begin by presenting the results of the feature importance modeling, correlation analysis, and rank consistency evaluation, which collectively narrow down the set of *Y Variables* for further analysis. I then analyze dataset-level patterns using scatterplot grids and marginal effect modeling (Section 5.4), followed by multivariate modeling with interaction terms to capture joint effects between features (Section 5.5). Next, I summarize the paired analysis comparing the impact of manual versus automatic part-of-speech annotations. Finally, I present a focused case study of the Meitei dataset to illustrate how specific structural properties interact with AGGREGATION performance.

5.1 Feature Importance Distributions

To begin the analysis, I present the normalized feature importances derived from the XGBoost models described in Section 3.5.2. For each of the 15 *Y variables*, I trained an individual XGBoost model and extracted the relative contribution of each *X variable* using the gain-based importance measure. These values were normalized to sum to 1 for each output and then producing a matrix that highlights how strongly each input feature contributed to reducing prediction error across different grammar quality metrics under the framework of the XGBoost model.

Figure 5.1 displays this matrix. Several strong patterns emerge. The feature *Number of Grams* dominates in predictions of *Verb Inflections per PC*, *Morphological Ambiguity*, and *Number of Verb Types*. *Average IGT Length in Words* is highly influential for *Coverage*

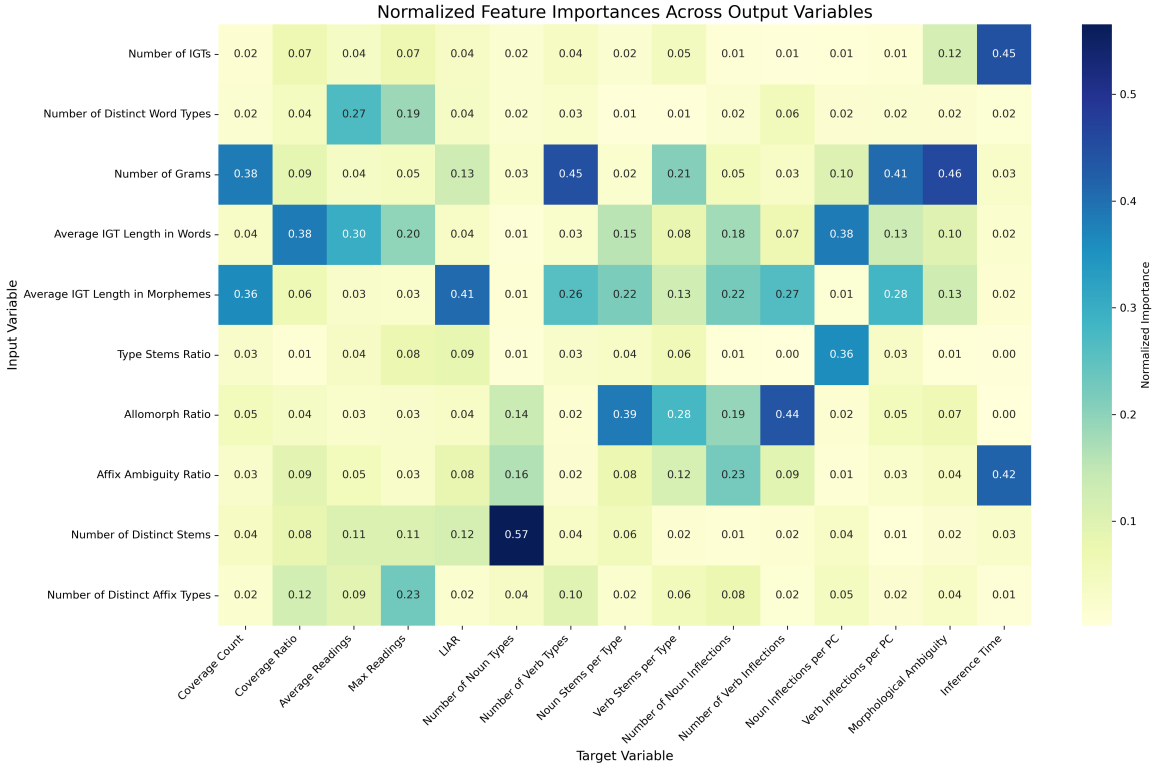


Figure 5.1: Normalized Feature Importances Across Output Variables (XGBoost Gain)

Ratio, *Average Readings*, and *Noun Inflections per PC*. Additionally, *Average IGT Length in Morphemes* contributes significantly to *LIAR*, *Verb Inflections*, and *Noun Inflections*. For efficiency modeling, *Inference Time* is most sensitive to *Number of IGTs*.

While these results reflect which *X Variable* the XGBoost models quantitatively relied upon for each prediction task, they do not establish whether those features are truly indicative of underlying linguistic structure or merely proxies for dataset characteristics. For instance, if datasets with more morphemes consistently lead to grammars with higher coverage, it remains an open question whether this trend reflects structural richness or test-set simplicity. This raises the broader question of whether model performance is shaped by linguistic structure or by dataset quality.

To address this ambiguity, in the next sections, I investigate whether test set structure alone correlates with output metrics. If grammar quality can be explained by test input characteristics, then some of the learned associations may be driven by surface-level dataset artifacts rather than generalizable structural effects.

5.2 Test Set Structure and Output Correlation

To investigate whether grammar quality outcomes are influenced by surface properties of the test data rather than structural characteristics learned from training, I computed Pearson correlation coefficients between test set features and grammar output metrics. This analysis is based solely on the 10% held-out test set for each dataset, which contains a fixed set of IGTs used in evaluation.

Figure 5.2 shows the Pearson correlation matrix between average structural properties of the held-out test sets and grammar quality metrics. Several strong associations emerge. For example, *Coverage Count* shows moderately strong correlations with *Number of IGTs* ($r = 0.68$), *Number of Grams* ($r = 0.64$), and *Number of Distinct Word Types* ($r = 0.56$). These patterns suggest that larger test sets may naturally lead to higher Coverage Count, simply by offering more opportunities for a sentence to fall within the grammar's coverage. In contrast, increased morphological richness would typically be expected to make parsing more difficult,

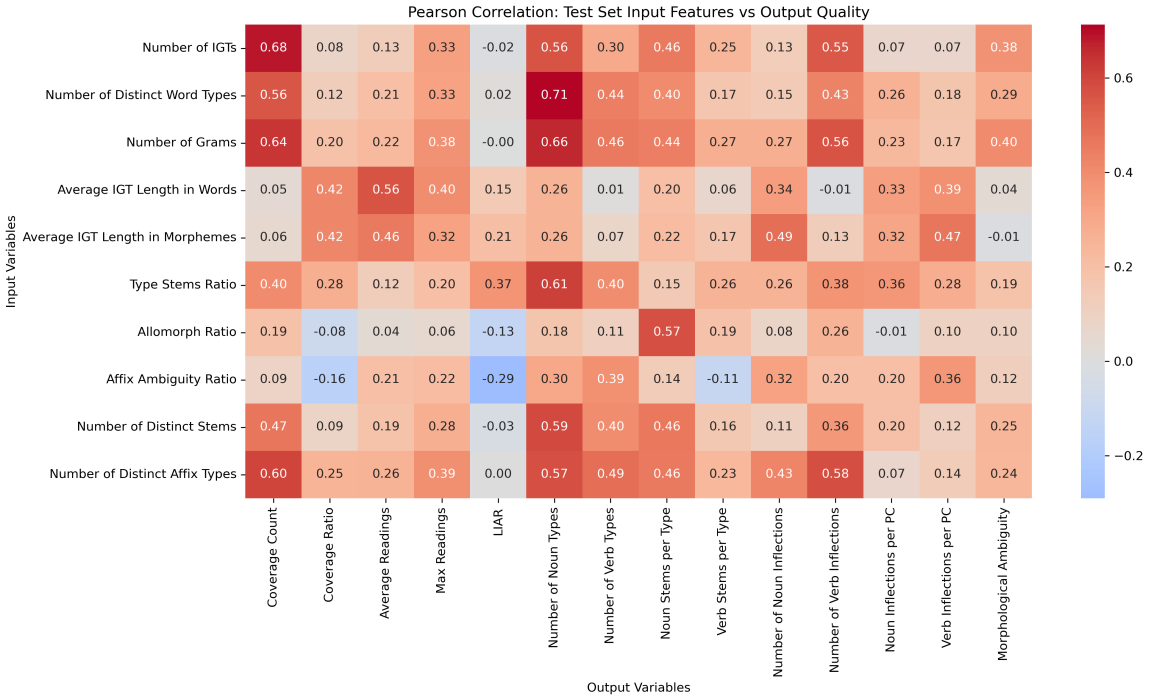


Figure 5.2: Pearson Correlation Between Test Set Structure and Output Quality Metrics

not easier. Therefore, any positive correlation between structural complexity and coverage may reflect dataset overlap or distributional artifacts, rather than true improvements in generalization.

Metrics such as *Average Readings* and *Maximum Readings* show moderate correlations ($r \approx 0.3$-0.5) with sentence length and morpheme counts, hinting at a possible link between structural richness and ambiguity. However, these associations are not uniformly strong across features.

In contrast, metrics like *LIAR* and *Morphological Ambiguity* exhibit weak or inconsistent correlations with all test-set variables ($|r| < 0.4$), suggesting their variation is less likely to be driven by superficial properties of the evaluation data.

However, interpreting these correlations in isolation is difficult. A high test-set correlation may indicate an artifact-prone metric, but it could also reflect overlap between training and test distributions. Conversely, a weak correlation might result from noisy measurements. To resolve this ambiguity, I next examine whether the same structural features are also ranked as important by the model when trained on training set variation. If a grammar metric is influenced by the same variables in both training and test settings, this suggests that it is globally sensitive to dataset artifacts. If not, then it is more likely to reflect generalizable structural patterns.

5.3 Rank Consistency Between Modeling and Test Correlations

To assess this alignment, I compute Spearman rank correlations between the feature importance scores derived from XGBoost (Section 3.5.2) and the test-set-based correlation coefficients discussed above. Each grammar quality metric is thus associated with two ranked lists of predictors, one from model-based training variation, and one from test-set structure.

The results, summarized in Table 5.1, indicate which output metrics are most similarly influenced by test-set structure and model behavior. Metrics with high rank consistency (e.g., *Max Readings* ($\rho = 0.588$, $p = 0.074$), *Verb Inflections per PC* ($\rho = 0.564$, $p = 0.090$), and *Noun Inflections per PC* ($\rho = 0.552$, $p = 0.098$)) are more likely to reflect dataset-

Output Variable	**Spearman ρ**	**p-value**
Max Readings	0.587879	0.073878
Verb Inflections per PC	0.563636	0.089724
Noun Inflections per PC	0.551515	0.098401
Average Readings	0.503030	0.138334
Number of Noun Inflections	0.466667	0.173939
Number of Verb Types	0.309091	0.384841
Coverage Ratio	0.236364	0.510885
LIAR	0.200000	0.579584
Verb Stems per Type	0.090909	0.802772
Number of Noun Types	-0.030303	0.933773
Morphological Ambiguity	-0.042424	0.907364
Noun Stems per Type	-0.066667	0.854813
Coverage Count	-0.357576	0.310376
Number of Verb Inflections	-0.660606	0.037588

Table 5.1: Spearman rank correlation between feature importance and test-set correlations.

specific artifacts. Conversely, those with low or non-significant rank alignment (e.g., *LIAR* ($\rho = 0.200$, $p = 0.580$); *Morphological Ambiguity* ($\rho = -0.042$, $p = 0.907$)) are considered more robust and artifact-resistant.

Although these low ρ values are accompanied by high p-values (indicating a lack of statistical significance), this is desirable in this context. Here, it is not testing whether a relationship exists, but rather using the absence of a consistent relationship as a filtering criterion. A high p-value suggests that the ranking of input features in the test-set-based correlations is not predictive of model behavior, meaning that the metric is less likely to be driven by test-set artifacts. In other words, weak and statistically non-significant rank agreement is evidence that the output metric captures information derived from training data rather than superficial properties of the evaluation inputs.

Based on this analysis, I select representative three output metrics from converage, ambiguity, grammar complexity dimensions, to carry forward into downstream modeling. For the ambiguity dimension, I choose *LIAR*, which shows the weakest alignment with test-set predictors. For grammar complexity dimension, I select *Morphological Ambiguity*, based on both its low test-set correlation and minimal rank consistency.

For the coverage dimension, although *Coverage Count* shows the lowest rank correlation, it is discarded due to its dependency on test-set size, which varies significantly across dataset and undermines comparability. Instead, I retain *Coverage*, which is normalized and more interpretable across datasets.

Finally, for the efficiency dimension, I use *Inference Time*, the only available metric in this category. In addition, it has no meaningful correlation with any structural property of the test set.

These selected metrics are used in the following sections to model the structural predictors of grammar performance.

5.4 Dataset-Level Visualization via Scatterplot Grids and Marginal Effects

To identify which structural features have the strongest individual influence on grammar output metrics, I begin by estimating their marginal effects using univariate linear mixed-effects models. As described in Section 3.5.5, for each output metric, I fit a series of univariate linear mixed-effects models, each estimating the effect of a single structural variable while controlling for dataset-level variation. These results are summarized in Tables 5.2 - 5.5, with one table per output variable. Each table lists the raw and standardized coefficients for the top ten predictors, ranked by absolute standardized effect size.

In parallel, I construct dataset-level scatterplot grids to visualize the relationships captured by these models. Each grid targets a single Y variable and shows one scatterplot per dataset-predictor pair. Rows correspond to datasets, and columns show the top-ranked predictors. These visualizations help diagnose cross-dataset consistency, detect non-linear or interaction effects, and identify outliers or anomalies. To ensure readability, each grid is split into multiple figures with no more than six datasets per figure and restricted to the top 5 predictors. Plots for the remaining predictors (ranks 6-10) are included in Appendix A.1.

5.4.1 Predictors of Coverage

Table 5.2 presents the results of the univariate mixed-effects models for *Coverage*. All top predictors show a strong and statistically significant positive association with coverage, indicating that datasets with richer or more varied structural properties lead to grammars with broader parsing coverage.

Figure 5.3 and Figure 5.4 visualize these relationships across the datasets. Despite variation in dataset sizes and sentence structures, the dominant pattern is a positive trend across predictors and datasets.

The most influential predictor is *Number of Grams*, followed by *Number of Distinct Affix Types* and *Type Stems Ratio*. These are all intuitive: a higher number of grams or affix types provides AGGREGATION with more morphological and grammatical information to

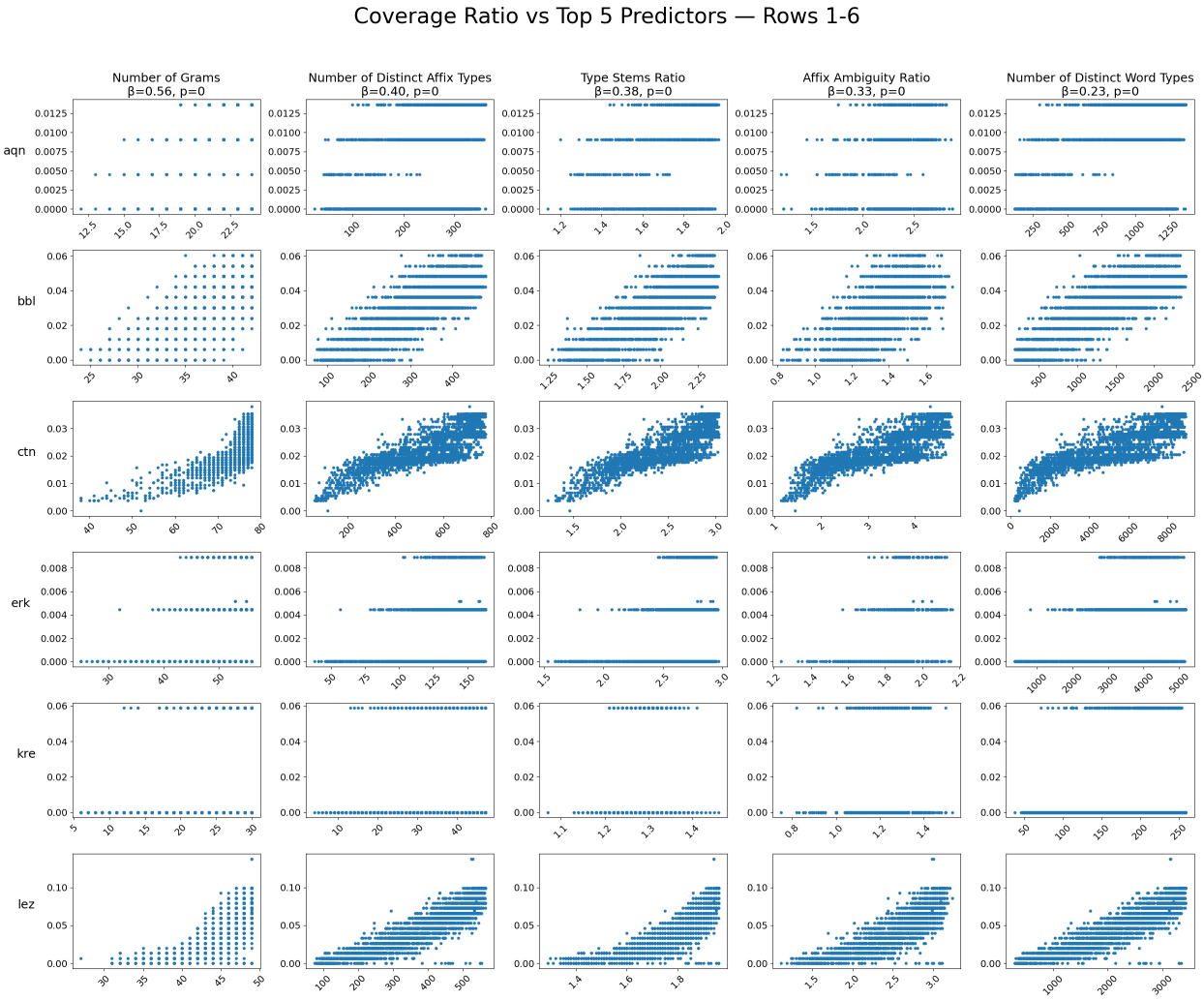


Figure 5.3: Scatterplot grid for *Coverage* vs top 5 structural predictors (Rows 1-6).

Input Variable	Raw Coef.	Std. Coef (Abs)	Std. Error	p-value
Number of Grams	0.00053	0.563	0.00979	< 0.001
Distinct Affix Types	3.75e-05	0.399	0.00459	< 0.001
Type Stems Ratio	0.01152	0.382	0.00512	< 0.001
Affix Ambiguity Ratio	0.00505	0.327	0.00616	< 0.001
Distinct Word Types	1.75e-06	0.232	0.00465	< 0.001
Distinct Stems	2.98e-06	0.197	0.00539	< 0.001
Number of IGTs	1.36e-06	0.143	0.00442	< 0.001
Allomorph Ratio	0.00292	0.100	0.00730	< 0.001
IGT Length (Words)	0.00067	0.097	0.03318	0.003
IGT Length (Morphemes)	0.00029	0.072	0.03557	0.042

Table 5.2: Mixed-effects model results predicting *Coverage* from structural features ranked by absolute standardized coefficient.

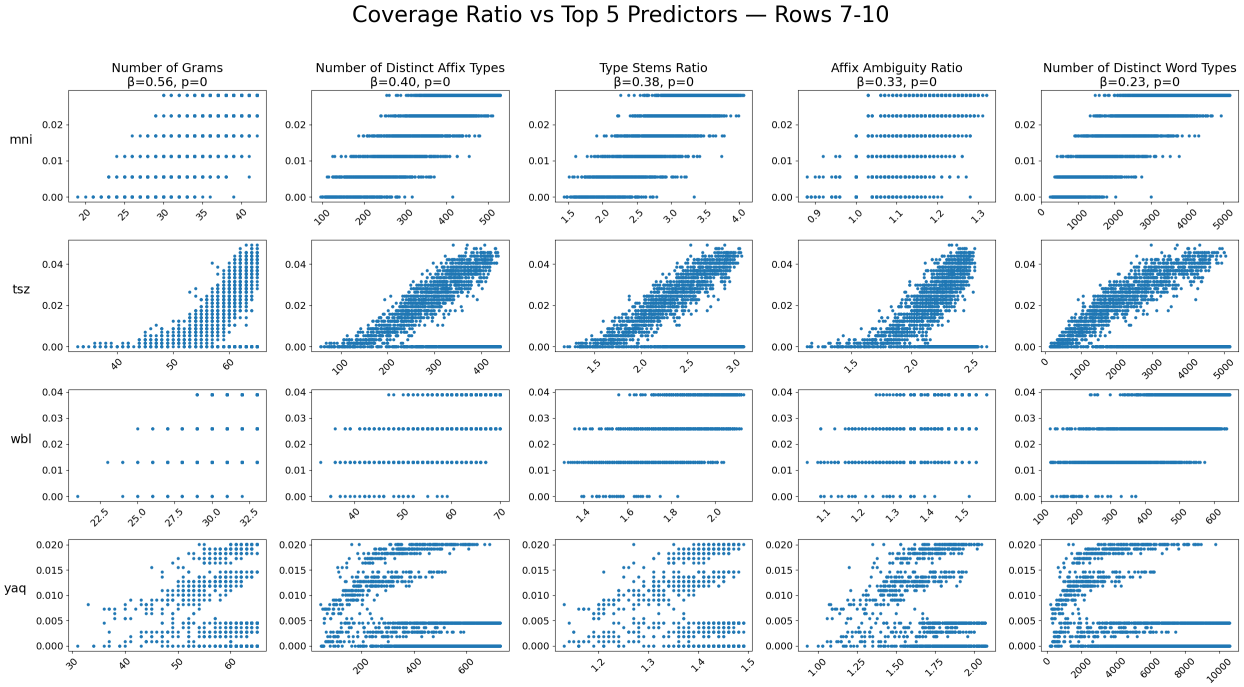


Figure 5.4: Scatterplot grid for *Coverage* vs top 5 structural predictors (Rows 7-12).

create grammars with more morphological rules that capture more of the test set. Similarly, a higher *Type Stems Ratio*, defined as the number of distinct word forms per stem, indicates that a lexicon with multiple inflected forms for a given stem helps expand the grammar and, consequently, parsing coverage.

Interestingly, *Affix Ambiguity Ratio* also shows a strong positive association with *Coverage*. One possible interpretation is that when affix forms are reused across multiple grammatical functions, the AGGREGATION system responds by generating a broader set of morphological rules (e.g., by assigning the same affix form to multiple position classes). This leads to grammars with more flexible rule inventories, which in turn capture more sentence structures during parsing. However, it is also possible that this effect simply reflects an increase in overall affix inventory size: datasets with high affix ambiguity may also contain more distinct affixes overall. Since both *Affix Ambiguity Ratio* and *Number of Distinct Affix Types* are positively correlated with coverage, further multivariate modeling is needed to isolate their respective contributions (Section 5.5.3).

In the scatterplots, I also observe that even datasets with sparse outputs (e.g., kre in Figure 5.3), where test items receive either 0 or 1 parse, still exhibit visible minor upward trends. This consistency, even with small data size, strengthens the case that the effect of these features is real and not dataset-specific noise.

Interestingly, while both *Distinct Word Types* and *Distinct Stems* are positively associated with coverage, their effects are weaker than those of affix-related predictors and the *Number of Grams*. This suggests that increasing lexical diversity through more word types or stems results in smaller returns compared to structural features. One possible explanation is a relatively small number of distinct stems and word forms can already support the induction of a usable lexicon. In contrast, affix inventory and the grams directly support the learning of morphological rules. Perhaps, more test sentences fail to receive parses because the generated grammar lacks the morphological rules required to analyze complex forms, rather than because lexical entries are missing. Further analysis could help verify this hypothesis.

Overall, the results for *Coverage* suggest that both volume (e.g., grams, affixes) and di-

versity (e.g., ambiguity, morphological variation) contribute meaningfully to grammar generation quality in low-resource settings.

5.4.2 Predictors of LIAR

Input Variable	**Raw Coef.**	**Std. Coef (Abs)**	**Std. Error**	**p-value**
Allomorph Ratio	-0.18909	0.270	0.00768	< 0.001
Type Stems Ratio	-0.14991	0.207	0.00566	< 0.001
Number of Distinct Stems	-7.48e-05	0.206	0.00573	< 0.001
Number of Distinct Word Types	-3.75e-05	0.206	0.00498	< 0.001
Number of Grams	-0.00441	0.196	0.01070	< 0.001
Number of Distinct Affix Types	-0.00043	0.192	0.00516	< 0.001
Number of IGTs	-4.33e-05	0.189	0.00467	< 0.001
Affix Ambiguity Ratio	-0.04683	0.126	0.00670	< 0.001
Average IGT Length in Morphemes	-0.00451	0.047	0.03768	0.214
Average IGT Length in Words	-0.00121	0.007	0.03514	0.837

Table 5.3: Mixed-effects model results predicting *LIAR* from structural features ranked by absolute standardized coefficient.

Table 5.3 presents the results of the univariate mixed-effects models for *LIAR*. Here, lower values of *LIAR* indicate greater parsing ambiguity, as more parses (readings) are generated per test sentence. The overall trend reveals that structural complexity (both morphological and lexical) is strongly associated with increased ambiguity.

The most influential predictor is *Allomorph Ratio* ($|\beta| = 0.270$), which exhibits a consistent negative association with *LIAR*. As seen in Figure 5.5 and 5.6, samples with higher *Allomorph Ratio* tend to produce lower *LIAR* values (more parses per sentence). This aligns with linguistic expectations: more form variants per gloss increase the combinatorial possibilities in morphological rules, resulting in more readings being generated by the grammar.

The second strongest predictor is *Type Stems Ratio* ($|\beta| = 0.21$), reflecting the average number of distinct word forms per stem. This variable also shows a steady negative trend

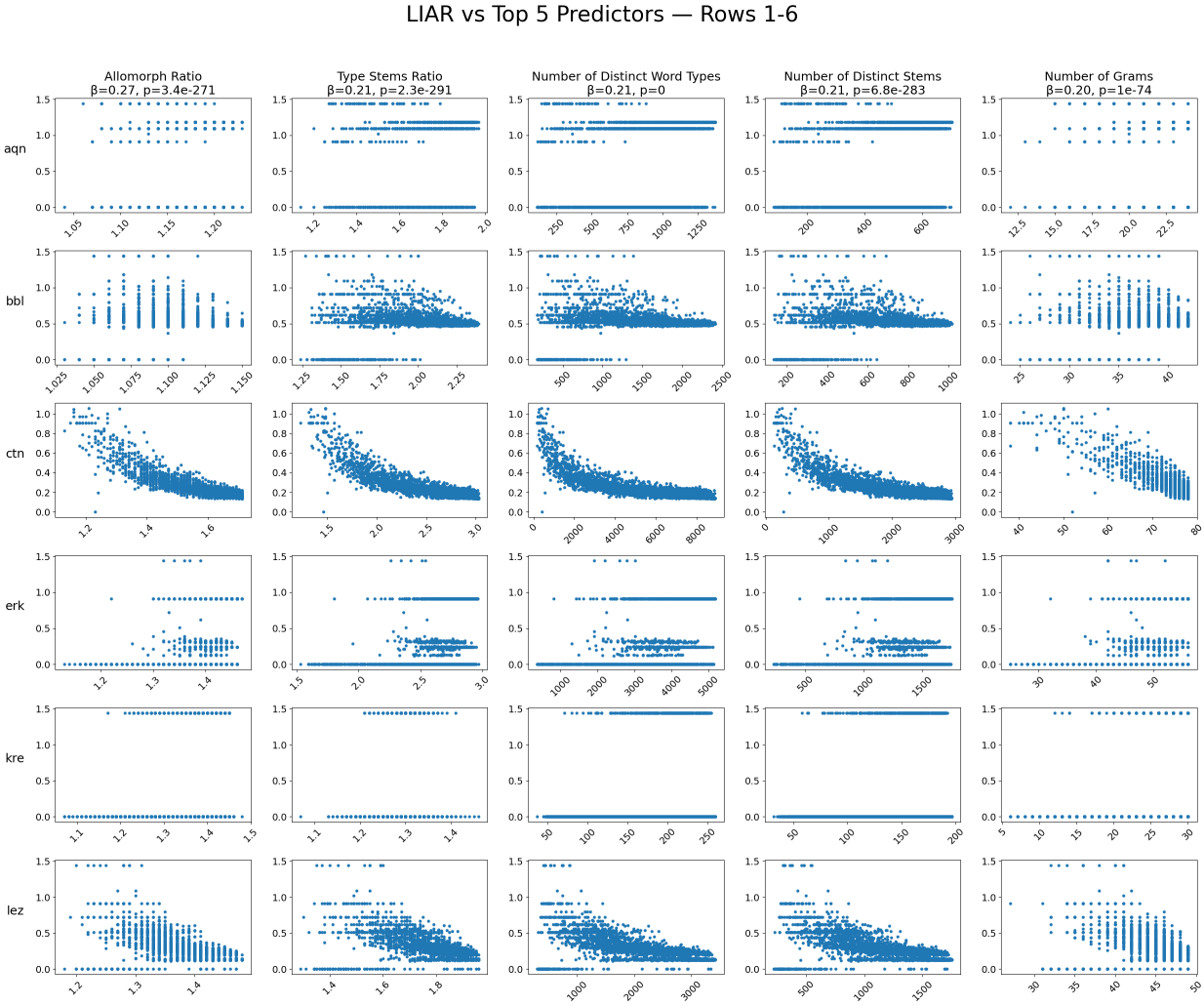


Figure 5.5: Scatterplot grid for *LIAR* vs top 5 structural predictors (Rows 1-6).

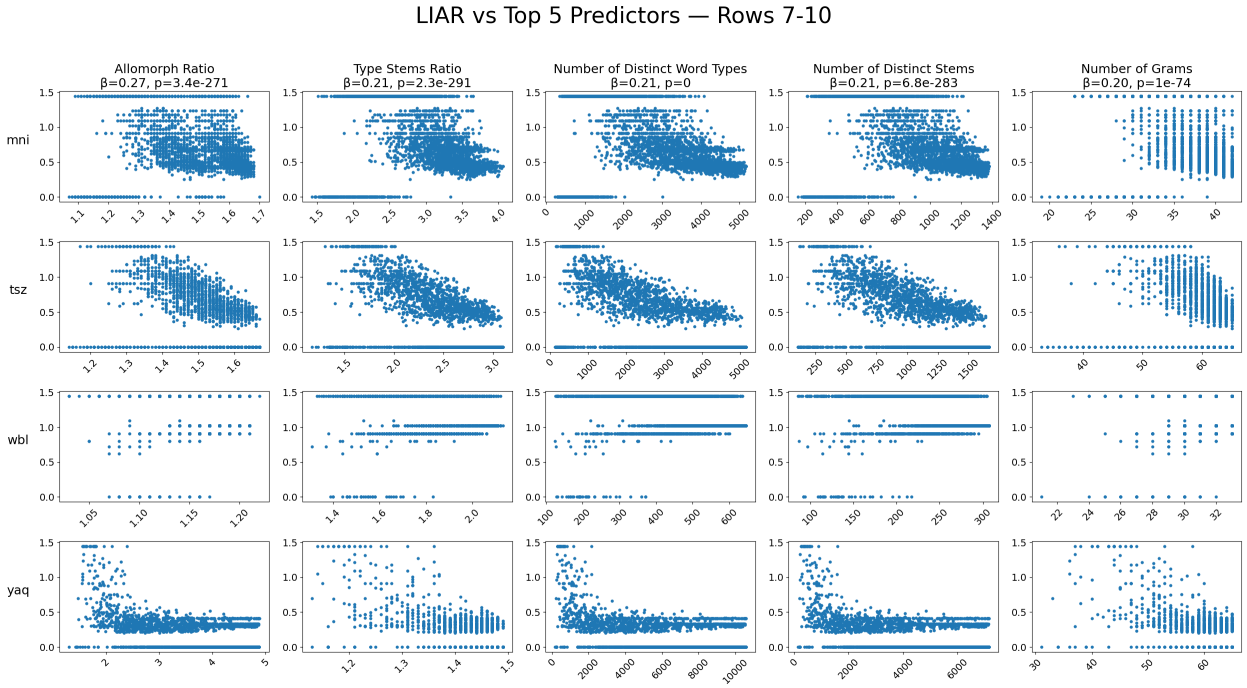


Figure 5.6: Scatterplot grid for *LIAR* vs top 5 structural predictors (Rows 7-12).

with *LIAR*. Higher inflected word type diversity per stem contributes to richer morphological rules generation, again leading to greater ambiguity during parsing.

Following closely are *Number of Distinct Word Types*, *Distinct Stems*, and *Number of Grams*, all with standardized coefficients around $|\beta| = 0.20$. These indicators of lexical and grammatical diversity also demonstrate consistent negative relationships with *LIAR*, implying that denser and more varied training data lead to more ambiguous grammar.

This pattern also resonates with the earlier observation that *Distinct Word Types* and *Distinct Stems* are positively associated with *Coverage*, but their effects are weaker than those of affix-related predictors and grams. In the case of *LIAR*, these lexical predictors do show relatively stronger influence. This asymmetry suggests that while a broader lexicon adds some value to both coverage and ambiguity, it is not the most efficient lever for improving grammar quality. A grammar built from a moderately sized lexicon can already generalize effectively if it is supported by a structurally rich inventory of affixes and grams. This insight motivates a practical design strategy: prioritize structural diversity (particularly affixes and gloss-tier grammatical functions) over maximal lexical inventory when developing grammars in low-resource settings. However, I will conduct a case study to examine this issue more closely in Section 5.7, and return to it in the broader context of cross-linguistic design strategies in Chapter 6.

Visual trends across the scatterplots are generally consistent. Most datasets exhibit clear downward trends, especially those with larger total number of IGT such as ctn, tsz, and mni. A few datasets, such as kre and wbl, show horizontal banding patterns because their test sets are small and contain only a few successfully parsed sentences. Nevertheless, the overall directionality remains consistent.

While these features strongly predict ambiguity independently, future multivariate modeling is necessary to establish their relative contributions more precisely (Section 5.5.3 - 5.5.3). For example, consider the potential interaction between *Allomorph Ratio* and *Number of Grams*. Linguistically, a dataset with both high allomorphic variation (many surface forms per gloss) and a rich inventory of grammatical categories (many distinct grams) may

lead the system to generate grammars that produce more parses per sentence. In such grammars, the large number of possible combinations between affix forms and grammatical functions could amplify syntactic ambiguity. This leads to the hypothesis that the negative impact of *Allomorph Ratio* on *LIAR* may be stronger in datasets with a large number of grams.

5.4.3 Predictors of Morphological Ambiguity

Input Variable	**Raw Coef.**	**Std. Coef (Abs)**	**Std. Error**	**p-value**
Number of Grams	0.01213	0.481	0.00549	< 0.001
Type Stems Ratio	0.20196	0.249	0.00295	< 0.001
Number of Distinct Affix Types	0.00057	0.224	0.00269	< 0.001
Number of Distinct Word Types	4.57e-05	0.224	0.00260	< 0.001
Number of Distinct Stems	8.62e-05	0.212	0.00304	< 0.001
Allomorph Ratio	0.15997	0.204	0.00417	< 0.001
Affix Ambiguity Ratio	0.08486	0.204	0.00357	< 0.001
Number of IGTs	4.96e-05	0.193	0.00246	< 0.001
Average IGT Length in Morphemes	0.00453	0.042	0.02093	0.045
Average IGT Length in Words	0.00703	0.038	0.01954	0.054

Table 5.4: Mixed-effects model results predicting *Morphological Ambiguity* from structural features ranked by absolute standardized coefficient.

Morphological Ambiguity is defined as the ratio of total verb rule instances to the number of unique orthographic forms. A higher value indicates that the same orthographic form maps to multiple rules, suggesting either true morphological ambiguity or overgeneralization. Conversely, a lower value reflects a more deterministic mapping, where each surface form is associated with fewer competing morphological rules.

Table 5.4 shows the results of the univariate mixed-effects models. The strongest predictor is *Number of Grams* ($|\beta| = 0.481$), indicating that datasets with richer grammatical label

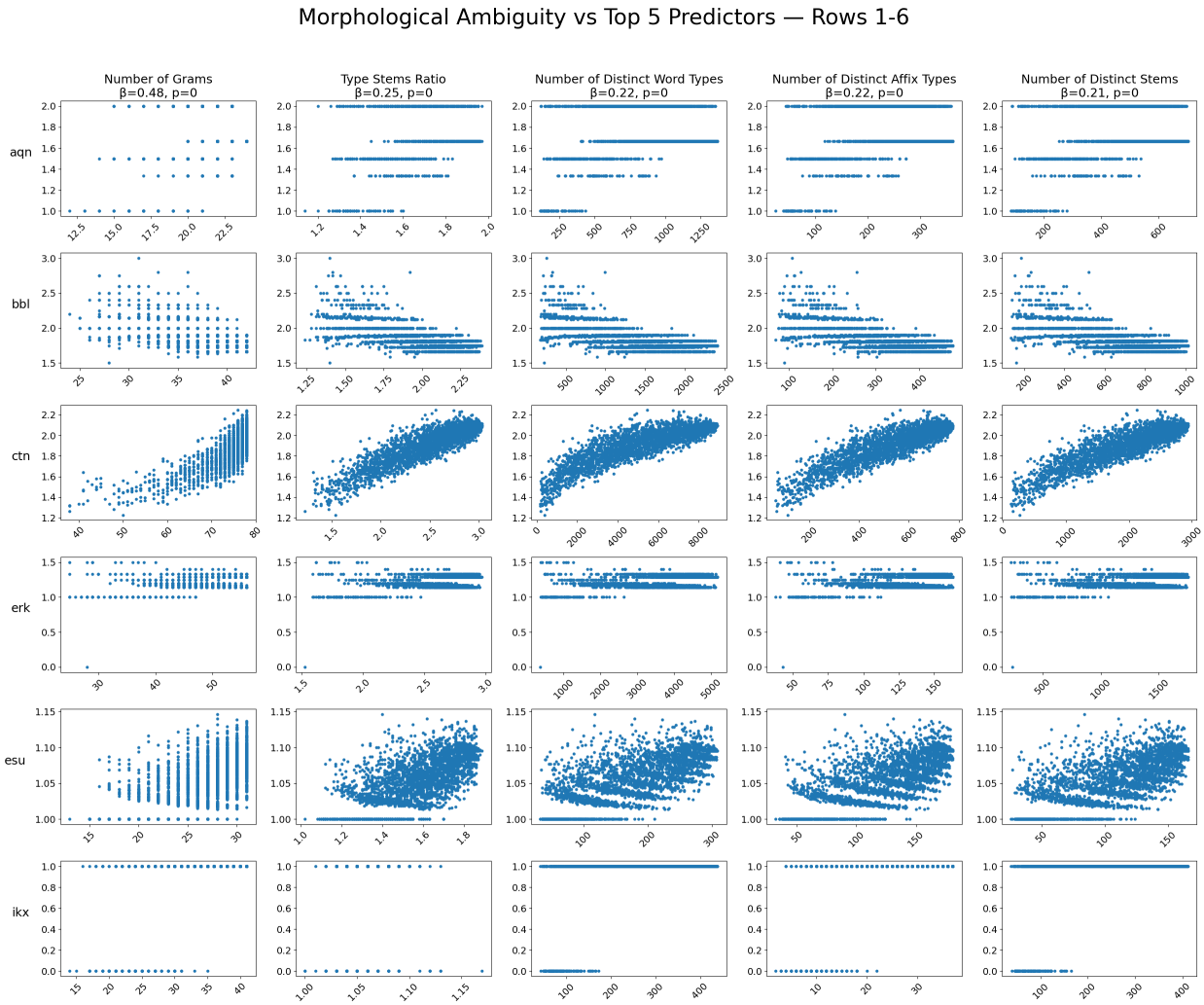


Figure 5.7: Scatterplot grid for *Morphological Ambiguity* vs top 5 structural predictors (Rows 1-6).

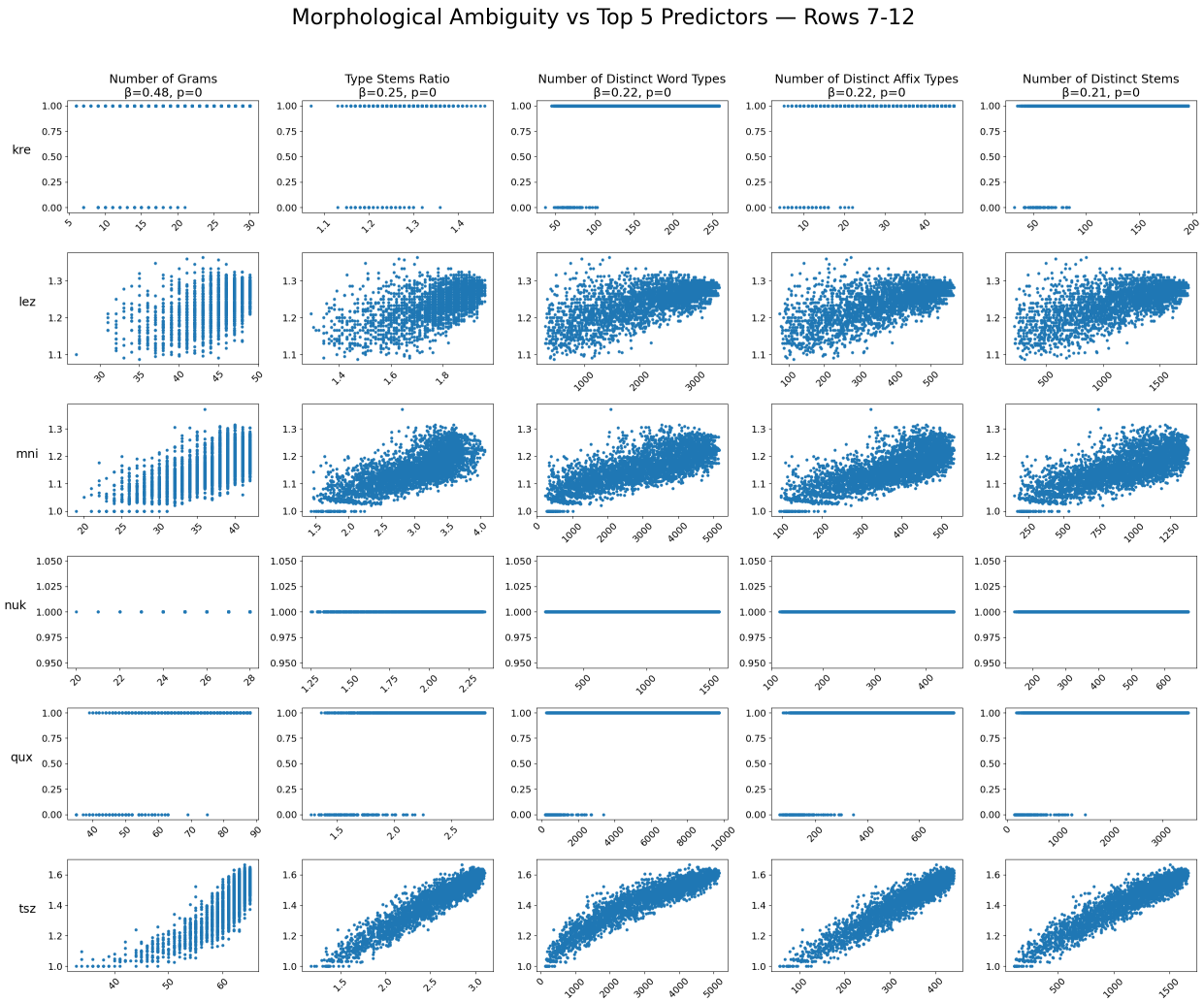


Figure 5.8: Scatterplot grid for *Morphological Ambiguity* vs top 5 structural predictors (Rows 7-12).

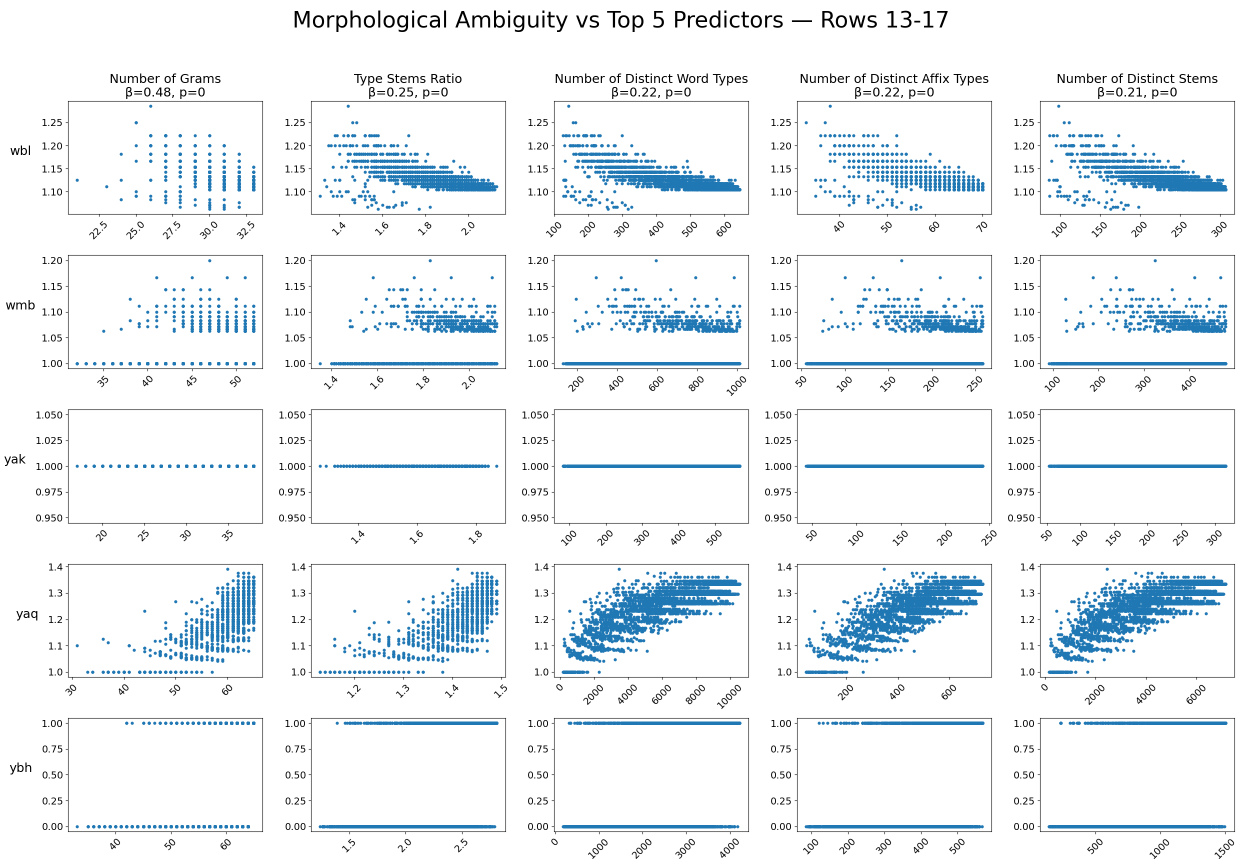


Figure 5.9: Scatterplot grid for *Morphological Ambiguity* vs top 5 structural predictors (Rows 13-18).

inventories tend to lead to grammars with higher morphological ambiguity. This likely reflects increased functional competition: as more grammatical distinctions are encoded, the system must express a larger set of functions using a limited set of surface forms, leading to greater reuse of forms across rules and hence increased ambiguity.

Type Stems Ratio ($|\beta| = 0.249$) also shows a positive effect. Datasets where each stem surfaces in more distinct forms tend to exhibit higher ambiguity, as surface forms reappear across morphological contexts. Similarly, features such as *Distinct Word Types*, *Distinct Affix Types*, and *Distinct Stems* (all $|\beta| \approx 0.22$) indicate that greater lexical and morphological diversity increases the number of possible rule-form alignments, raising grammar complexity levels.

Figures 5.7-5.9 further illustrate these trends. Most datasets exhibit clear monotonic upward relationships, particularly those with larger training sets and richer morphological structures such as ctn, tsz, and mni. In contrast, some datasets such as ixk and yak display flat or banded distributions. This reflects limitations in the generated grammars. Specifically, their morphology sections are small or absent, resulting in consistently low or even zero morphological ambiguity values.

Interestingly, a few datasets (bbl, wbl, and wmb) deviate from the global trend, showing weak or even negative correlations between structural predictors and *Morphological Ambiguity*. Upon closer inspection, these datasets share typological characteristics that may explain the divergence.

These datasets rely on compositional reuse of affix templates. That is, instead of introducing new morphological rules for each function or form, they use a small set of general-purpose affixes in predictable combinations. For example, in wbl, the form *j-a-w-ing-r* (`DEM-MED-PRO-NO_GLOSS-DAT`) expresses demonstrative, distance, pronoun, connective, and case features all within a single word. In wmb, constructions like *irri-n* (`3.PL.S.NP-PROG`) and *bungmanya-nka* (`old.woman.II-DAT`) encode subject, tense, and case in compact agglutinative chains.

In bbl, the same suffixes (*-i*, *-n*, *-x*) appear across many words (e.g., *anik'o-i-n*, *qa-i-n*,

kotam-e-x), but the grammar links them to shared lexical rules. As a result, the number of unique rule instances grows slowly, even as the number of observed surface forms increases rapidly through recombination.

These encoding strategies has a direct effect on *Morphological Ambiguity*. When grammatical functions are expressed using highly reusable affix sequences, the numerator (rule instances) grows slowly, while the denominator (surface forms) grows more quickly. As a result, the overall ratio decreases, even though the underlying morphological system is structurally complex.

This explains the observed negative correlations: datasets like wbl, wmb, and bbl achieve expressive power through efficient reuse of morphological information, compressing multiple meanings into tightly packed, templatic word forms. Structural predictors such as *Number of Stems* or *Affix Types* increase as the system becomes more complex, but the grammar complexity metric decreases when the same rules are reused across more combinations.

5.4.4 Predictors of Inference Time

Input Variable	**Raw Coef.**	**Std. Coef (Abs)**	**Std. Error**	**p-value**
Number of IGTs	0.150	0.750	0.00331	< 0.001
Affix Ambiguity Ratio	230.87	0.714	0.00585	< 0.001
Number of Distinct Word Types	0.096	0.604	0.00424	< 0.001
Number of Distinct Affix Types	1.121	0.571	0.00451	< 0.001
Number of Grams	10.94	0.557	0.01032	< 0.001
Type Stems Ratio	328.88	0.521	0.00518	< 0.001
Number of Distinct Stems	0.157	0.497	0.00530	< 0.001
Allomorph Ratio	223.01	0.366	0.00751	< 0.001
Average IGT Length in Words	4.719	0.032	0.03472	0.349
Average IGT Length in Morphemes	2.704	0.032	0.03716	0.385

Table 5.5: Mixed-effects model results predicting *Inference Time* from structural features ranked by absolute standardized coefficient.

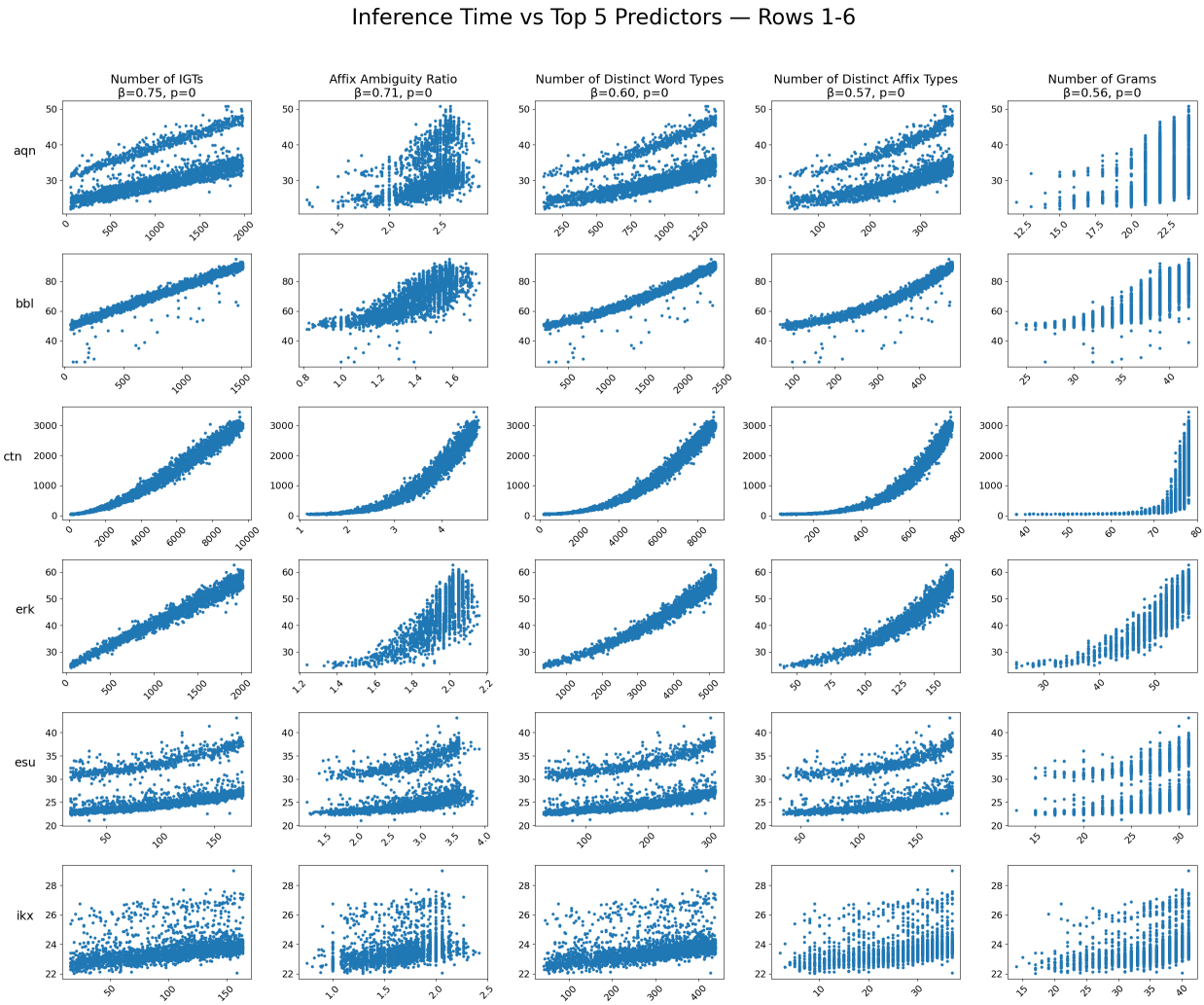


Figure 5.10: Scatterplot grid for *Inference Time* vs top 5 structural predictors (Rows 1-6).

Figures 5.10 through 5.12 show the relationship between *Inference Time* and the top five structural predictors. As shown in Table 5.5, all five variables have standardized coefficients above $|\beta| = 0.56$, which indicate a strong and consistent relationship.

The strongest predictor is *Number of IGTs* ($|\beta| = 0.75$), which aligns with intuition: larger training datasets impose greater computational loads, resulting in longer inference times. Other top predictors, *Affix Ambiguity Ratio*, *Number of Distinct Word Types*, *Number*

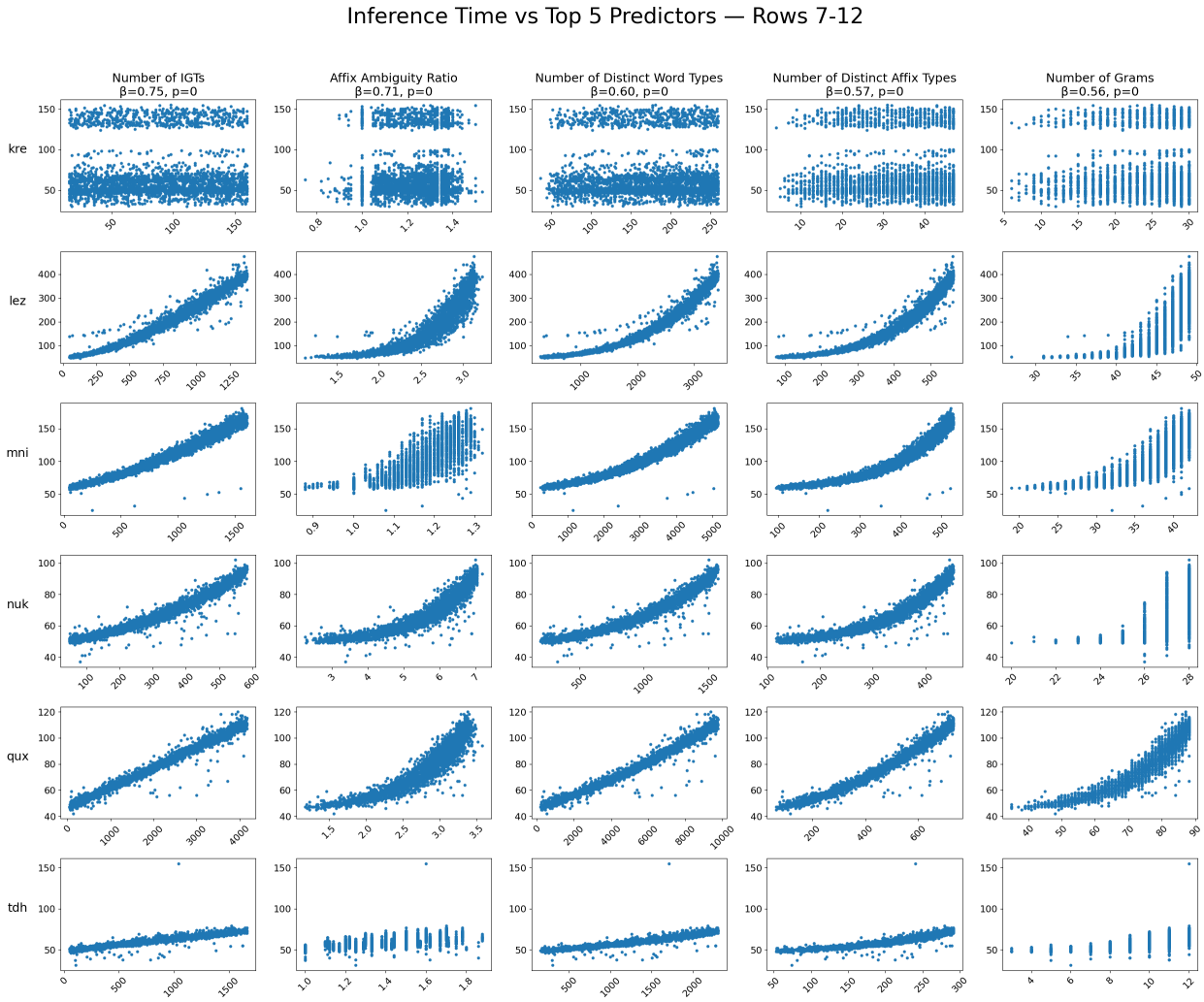


Figure 5.11: Scatterplot grid for *Inference Time* vs top 5 structural predictors (Rows 7-12).

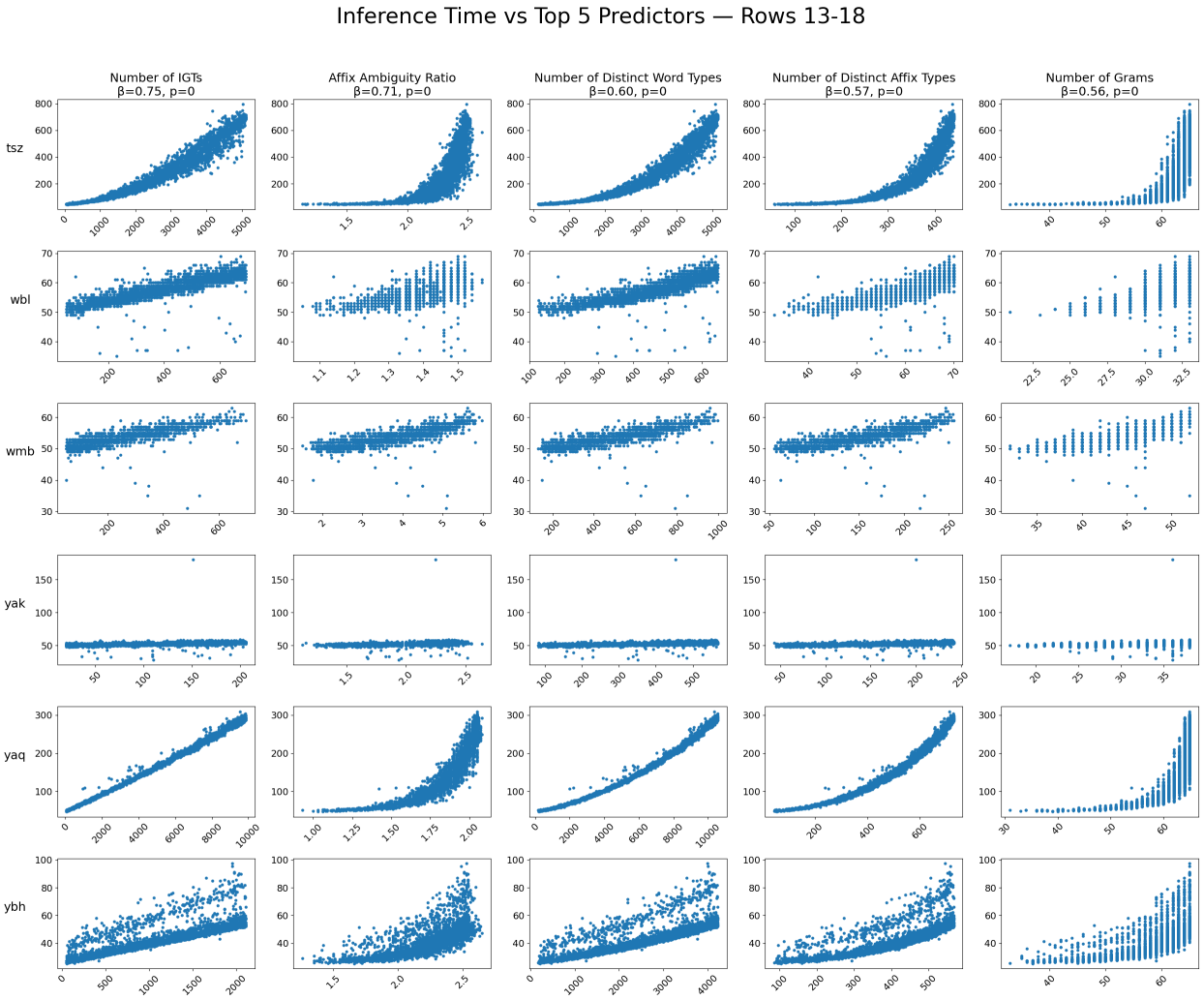


Figure 5.12: Scatterplot grid for *Inference Time* vs top 5 structural predictors (Rows 13-18).

of Distinct Affix Types, and *Number of Grams*, also show positive associations, suggesting that morphological and lexical complexity also affect inference cost.

In datasets such as ctn and tsz, scatterplots reveal that *Inference Time* scales roughly linearly with *Number of IGTs* up to a moderate size (hundreds to a few thousand), but increases more sharply in datasets with a larger number of IGTs. This nonlinear acceleration, seen in ctn and yaq, suggests that AGGREGATION system's computational costs under large-scale training data may grow superlinearly, potentially due to internal memory usage or increased syntactic backtracking during parsing.

Affix Ambiguity Ratio is also particularly influential ($|\beta| = 0.71$), ranking second among all predictors. In several datasets (e.g., tsz, qux, yaq), it exhibits a nonlinear positive relationship with inference time. When individual affixes map to multiple grammatical functions, the system must consider more possible feature combinations, which may exponentially expands the set of valid rules and significantly increases computation time.

An additional pattern observed in datasets like aqn, esu, kre, and ybh is the presence of distinct clusters in the scatterplots: multiple linear segments offset vertically, even within similar ranges of predictors. These stepwise distributions suggest the existence of potential threshold effects caused by external factors.

One explanation is that these shifts are not linguistic in origin but technical. System-level events such as CPU throttling, memory paging, or concurrent thread contention may intermittently degrade the system's performance. Notably, these clustering effects do not appear in other metrics (e.g., *LIAR* or *Morphological Ambiguity*), as *Inference Time* is also uniquely sensitive to runtime system dynamics.

Taken together, the univariate mixed-effects models reveal that a range of structural features are strong individual predictors of grammar performance across all output metrics. However, because linguistic features often co-vary and may influence each other's impact, in the next section, I extend the analysis to multivariate models that include interaction terms to examine how combinations of predictors jointly shape output metrics in non-additive ways.

5.5 Multivariate Modeling with Interaction Terms

In this section, I focus on identifying and evaluating interaction effects between structural features of datasets that may jointly influence the performance of grammars generated by the AGGREGATION system.

5.5.1 Confirming Predictor Utility via Stepwise Selection

Before examining individual interaction effects, I first run a stepwise feature selection procedure using AIC minimization to assess the overall predictive utility of structural variables for each grammar output metric. The goal is to verify that multiple structural properties of the input data contribute meaningfully to grammar outcomes. In each case, the model including top five predictors achieves the lowest AIC, indicating that the explanatory power of the variables is not limited to the specific pairs chosen for interaction modeling.

For *Coverage*, the best five-variable model results in an AIC of 79755.92 (compared to a baseline of 87586.18); for *LIAR*, the stepwise model achieves 97046.40 (baseline: 101091.82); for *Inference Time*, 48925.19 (baseline: 60579.44); and for *Morphological Ambiguity*, 30008.34 (baseline: 33377.70). These reductions confirm that each outcome is shaped by interrelated structural features.

While the remainder of this section concentrates on the interaction effects between two variables at a time, this stepwise analysis shows that multiple variables may interact in shaping grammar performance. The decision to restrict the current analysis to two-way interactions is driven by interpretability and scope, but future work could explore higher-order effects in a more systematic framework.

5.5.2 Selection Rationale and Variable Justification

Rather than exhaustively enumerating all possible feature combinations, I concentrate on a small set of linguistically motivated pairings. Each selected interaction involves two structural predictors that both demonstrated strong marginal effects in the preceding mixed-

effects models and exhibit plausible functional interplay within a grammar inference context. By explicitly modeling these interactions, I aim to capture non-additive effects that may be obscured under purely additive specifications.

For *Coverage*, I examine the interaction between *Number of Distinct Affix Types* and *Affix Ambiguity*. Earlier, I noted that *Affix Ambiguity* was positively associated with *Coverage*, potentially because ambiguous affixes, those reused across multiple grammatical roles, lead the system to generate more flexible morphological rules that apply to a wider range of forms. However, this effect may not be independent of the overall size of the affix inventory. A larger set of affixes increases the likelihood of ambiguity, so the observed effect of *Affix Ambiguity* on *Coverage* may simply reflect the influence of affix richness on the instance of morphological rules generated by the system, which in turn affects parsing outcomes.

To disentangle these possibilities, I include an interaction term between *Affix Ambiguity Ratio* and *Number of Distinct Affix Types*. This tests whether the effect of affix ambiguity on parsing coverage depends on the size of the affix inventory (i.e., whether ambiguity only contributes meaningfully in grammars with sufficiently rich morphological systems). A significant interaction would indicate that ambiguity is not uniformly beneficial, but instead interacts with morphological richness to shape grammar effectiveness.

For *LIAR*, I consider two interaction structures involving *Allomorph Ratio*. The first is with *Type Stems Ratio*. This pairing captures a hypothesis in morphosyntactic complexity: when a dataset exhibits both variation in stem realizations (many distinct word forms per stem) and extensive allomorphy, the AGGREGATION system must resolve a large number of surface form variants associated with different grammatical functions, increasing syntactic ambiguity. Allomorphy may help the system learn more flexible morphological rules when considered alone, but when combined with high stem variation, it can lead to overgeneration and a loss of grammatical distinction. The second interaction is between *Allomorph Ratio* and *Number of Grams*. Here, I want to test test whether the potential for allomorphy to support flexible rule learning diminishes in datasets with a large number of grammatical labels. In such cases, the number of possible combinations between affix forms and grammatical

functions may overwhelm the system's ability to generalize effectively, resulting in increased ambiguity.

For *Inference Time*, I include an interaction between *Number of IGTs* and *Affix Ambiguity*. Scatterplots suggested a non-linear relationship in which parsing time remains stable across increasing dataset size when ambiguity is low, but grows steeply under high ambiguity. This interaction tests whether *Affix Ambiguity* amplifies the computational cost of scaling up data.

Finally, for *Morphological Ambiguity*, where I could not intuitively identify a clear theoretical pairing, I adopt a data-driven default: the interaction between *Number of Grams* and *Type Stems Ratio*, both identified as top individual predictors. This interaction captures the idea that grammatical richness (more distinct grams) combined with greater morphological form diversity (more surface forms per stem) leads to a denser space of morphological rules. In such settings, especially when training data is limited, the system may generate overlapping or ambiguous rules for similar surface forms, increasing the chance that a single form maps to multiple morphological analyses.

These five interactions serve as focused cases for multivariate modeling.

5.5.3 Interaction Modeling Results

To assess whether these five interactions bring non-additive effects on grammar performance, I first fit mixed-effects models for these pairs. Each interaction was tested for statistical significance and assessed in terms of its incremental explanatory value (via ΔAIC) over baseline additive models.

All five core interactions demonstrate significant effects ($p < 0.001$), and contribute meaningful improvements in AIC relative to their additive counterparts. For *Coverage*, the interaction between *Distinct Affix Types* and *Affix Ambiguity Ratio* is statistically significant and contributes an improvement in model fit (ΔAIC = -2073.56). The negative interaction coefficient (-0.06) suggests a diminishing return effect: while each predictor individually has a positive or neutral influence on coverage, their combination slightly reduces the overall

Y Variable	**X1 (Main Effect)**	**X2 (Main Effect)**	**Coef X1**	**Coef X2**	**Coef Interaction**	p **(Interaction)**	**Δ AIC**
Coverage	Distinct Affix Types	Affix Ambiguity	0.43	-0.019	-0.06	$< .001$	-2073.56
LIAR	Allomorph Ratio	Type Stems Ratio	-0.70	-0.08	-0.627	$< .001$	-3089.80
LIAR	Allomorph Ratio	Number of Grams	-0.09	-0.10	-0.123	$< .001$	-94.87
Morph. Ambig.	Number of Grams	Type Stems Ratio	0.27	0.11	0.115	$< .001$	-3369.12
Inference Time	Number of IGTs	Affix Ambig. Ratio	0.29	0.26	0.602	$< .001$	-116752.99

Table 5.6: Interaction model summaries for five selected predictor pairs.

gain. This pattern suggests that affix ambiguity may help the system produce more flexible morphological rules when the affix inventory is small, but in affix-rich contexts, it may instead lead to structural overgeneration, ultimately reducing parsing coverage.

For *LIAR*, both interaction terms are statistically significant, and the effect size is notably larger for the *Allomorph Ratio* and *Type Stems Ratio* pairing (-0.627, ΔAIC = -3089.80), which aligns with theoretical expectations about form-function explosion. The alternative pairing with *Number of Grams* has a smaller, though still significant, interaction effect (-0.123, ΔAIC = -94.87), suggesting that in datasets with a large number of grams, the presence of extensive allomorphy contributes more strongly to overgeneration, likely due to the increased number of possible combinations between form variants and grammatical labels.

In the *Morphological Ambiguity* model, the interaction between *Number of Grams* and *Type Stems Ratio* is strongly positive and statistically significant (ΔAIC = -3369.12). This supports the hypothesis that rich grammatical marking, when combined with a diverse set of word forms, contributes to increased lexical rule redundancy, that is, more rules are needed to account for overlapping form-function mappings, particularly when surface forms multiply through productive derivation.

For *Inference Time*, the interaction between *Number of IGTs* and *Affix Ambiguity Ratio* has the strongest absolute model improvement of all (ΔAIC = -116752.99), confirming that computational load scales non-linearly when ambiguity increases alongside dataset size.

In the next sections, I will provide detailed analyses of each interaction pair, combining visualizations with targeted interpretation to illustrate how specific structural combinations influence grammar performance, and explain the linguistic mechanisms underlying the statistical patterns observed above.

Coverage: Distinct Affix Types × Affix Ambiguity

Figure 5.13 visualizes how *Coverage* varies as a function of both *Distinct Affix Types* and *Affix Ambiguity*. Each point represents a model trained on a dataset variant, with color encoding the level of *Affix Ambiguity*.

This gradient reveals a clear interaction pattern: while an increase in *Distinct Affix Types* is generally associated with higher *Coverage*, this positive effect is modulated by the level of *Affix Ambiguity*. For grammars with low *Affix Ambiguity* (dark blue), coverage increases steadily as the number of affix types grows. In contrast, for grammars with high *Affix Ambiguity* (orange to red), the benefit of additional affix types diminishes.

This pattern visually confirms the negative interaction term observed in the mixed-effects model (Table 5.6). While increasing the number of *Distinct Affix Types* generally expands the expressiveness of the generated grammar and improves *Coverage*, this effect is moderated by the level of *Affix Ambiguity*. In datasets where the same affix frequently maps to multiple grammatical functions, the AGGREGATION system struggles to assign clear, functionally distinct rules to each form. As a result, the learned grammar may fail to include certain lexical or morphological rules required to analyze test examples. This explains why coverage gains diminish or reverse when affix ambiguity rises in morphologically rich settings.

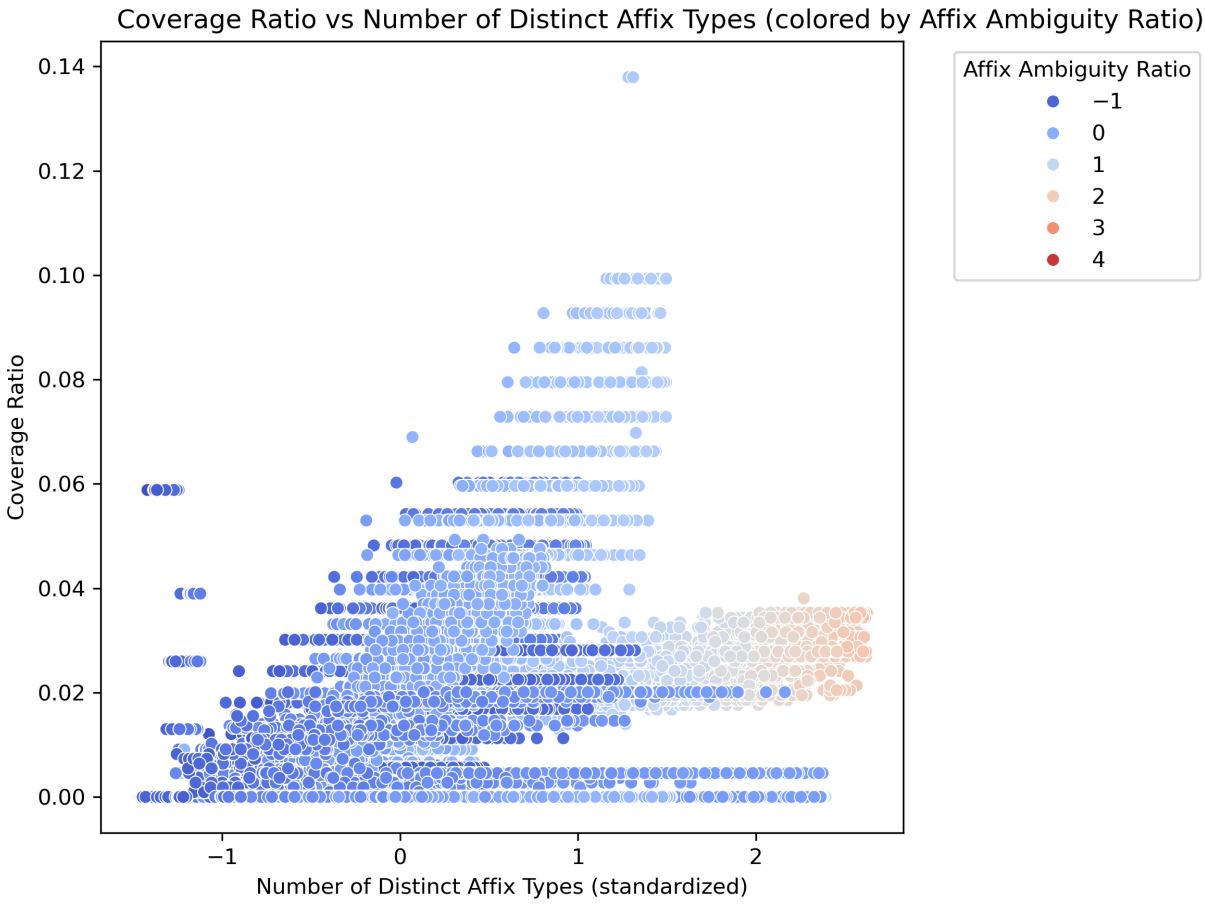


Figure 5.13: Interaction between Distinct Affix Types and Affix Ambiguity on Coverage.

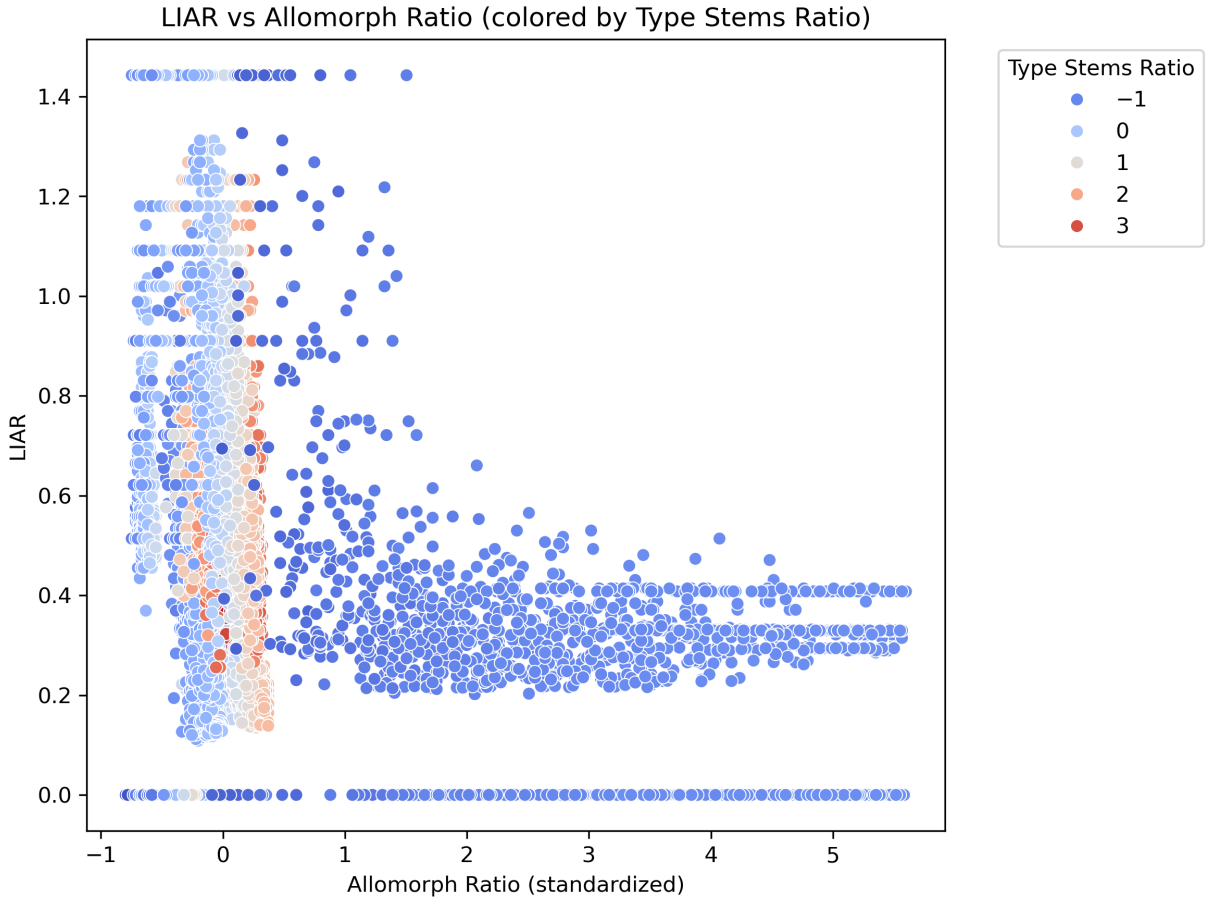


Figure 5.14: Raw interaction plot between *Allomorph Ratio* and *Type Stems Ratio* on *LIAR*, before outlier removal.

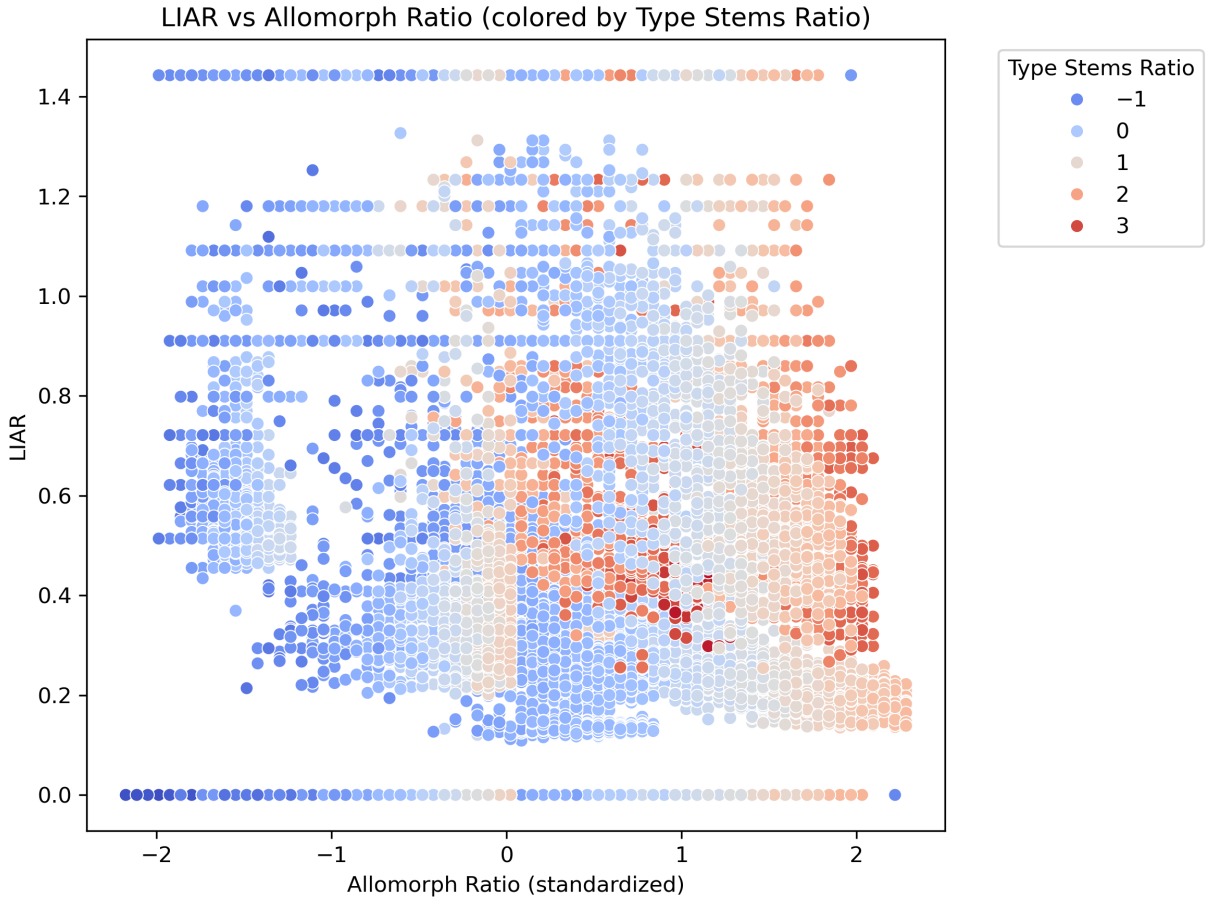


Figure 5.15: Interaction between *Allomorph Ratio* and *Type Stems Ratio* on *LIAR*, with *yaq* outlier removed.

LIAR: Allomorph Ratio × Type Stems Ratio

Figures 5.14 and 5.15 visualize the interaction between *Allomorph Ratio* and *Type Stems Ratio* in predicting *LIAR*. In the first figure, extremely high *Allomorph Ratio* values (primarily from the *yaq* dataset) obscure the overall pattern. After removing this outlier in the second figure, the interaction becomes clearer: the horizontal axis now spans a standardized range, and the color gradient for *Type Stems Ratio* is more uniformly distributed.

In the cleaned plot, a consistent downward trend in *LIAR* is visible: datasets with both high *Allomorph Ratio* and high *Type Stems Ratio* (red curves, right side) exhibit lower *LIAR* values, indicating greater structural ambiguity. The interaction plot visually supports this interpretation: the slopes diverge across color bands, with red curves dropping more steeply as *Allomorph Ratio* increases, which is the evidence of a non-additive effect. Mixed-effects model estimates confirm this pattern: both main effects are negative (*Allomorph Ratio*: $\beta = -0.70$, *Type Stems Ratio*: $\beta = -0.08$), and the interaction is strongly negative ($\beta = -0.63$), with a substantial AIC improvement (ΔAIC = -3089.80). This supports a compounding effect on ambiguity.

Linguistically, this pattern reflects a morphosyntactic overload effect: high allomorphy increases the number of surface forms associated with each gloss, while a high *Type Stems Ratio* reflects many distinct word forms derived from the same stem. When both conditions are present, the space of possible mappings between forms and grammatical functions expands rapidly, leading the system to generate a large number of morphological rules. This results in grammars that produce more structural analyses per sentence, increasing ambiguity.

LIAR: Allomorph Ratio × Number of Grams

To further assess parsing ambiguity, I examine the interaction between *Allomorph Ratio* and *Number of Grams* in predicting *LIAR*. Figure 5.16 plots standardized *Allomorph Ratio* against raw *LIAR*, with line color encoding standardized *Number of Grams* (blue = low, red = high).

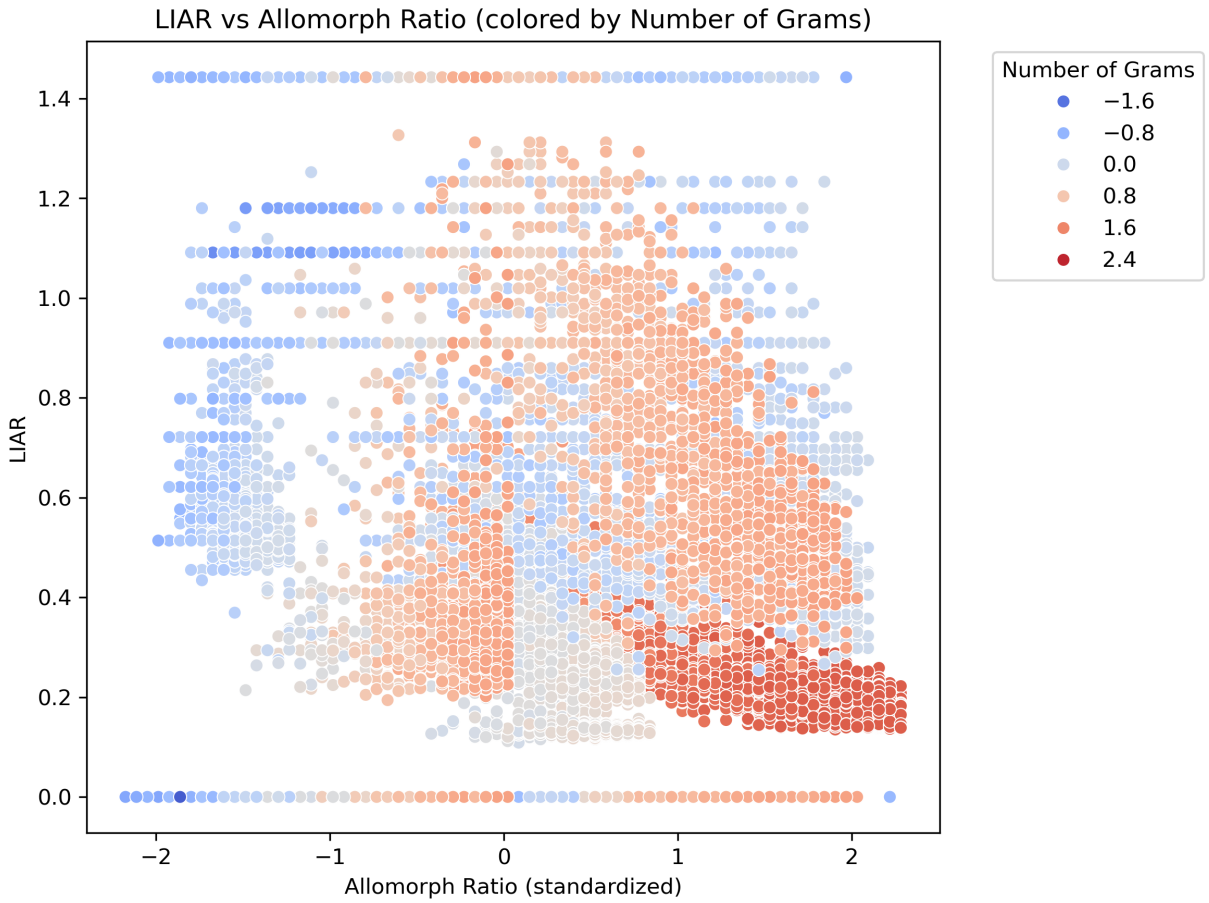


Figure 5.16: Interaction between *Allomorph Ratio* and *Number of Grams* on *LIAR* (excluding yaq).

Both predictors show negative main effects: *Allomorph Ratio* ($\beta = -0.095$) and *Number of Grams* ($\beta = -0.104$) are independently associated with lower *LIAR*, indicating higher ambiguity. The interaction term is also negative and significant ($\beta = -0.123$, ΔAIC = -94.87), suggesting a compounding effect. This is clear on the right side of Figure 5.16, where high-Grams curves (in red) descend more steeply as allomorphy increases.

This interaction captures a combinatorial overload: in datasets with both rich grammatical marking and high allomorphy, each morpheme must be evaluated across a wide range of possible grammatical functions. The result is an exponential growth in form-function mappings. This interaction thus reflects a structural property of morphosyntactically complex systems, and highlights the difficulty of grammar generation under high morphological and grammatical diversity.

Morphological Ambiguity: Number of Grams × Type Stems Ratio

To examine sources of morphological ambiguity in grammar construction, I model *Morphological Ambiguity* as a function of *Number of Grams*, *Type Stems Ratio*, and their interaction. Figure 5.17 includes all datasets, revealing a dominant outlier (mni) with a high average *Type Stems Ratio* ($\approx +3$). This dataset shows a strong red band near the center of the x-axis, indicating that the dataset has a high *Type Stems Ratio* at a moderate level of *Number of Grams*. To assess general patterns, I exclude mni in Figure 5.18.

The cleaned plot reveals a more interpretable structure: datasets with high *Type Stems Ratio* are now more evenly distributed across the upper-right quadrant. The overall trend shows that *Morphological Ambiguity* increases with both predictors, consistent with the model's positive main effects ($\beta_{Grams} = +0.265$, $\beta_{Type\ Stems\ Ratio} = +0.110$) and a strong positive interaction ($\beta = +0.115$, ΔAIC = -3369.12). This suggests that grammatical richness and stem variation jointly contribute to increased rule complexity in the generated grammars.

In contrast to earlier interactions that reflected ambiguity driven by combinatorial explosion, this pattern appears additive: datasets with many grams and high stem variation

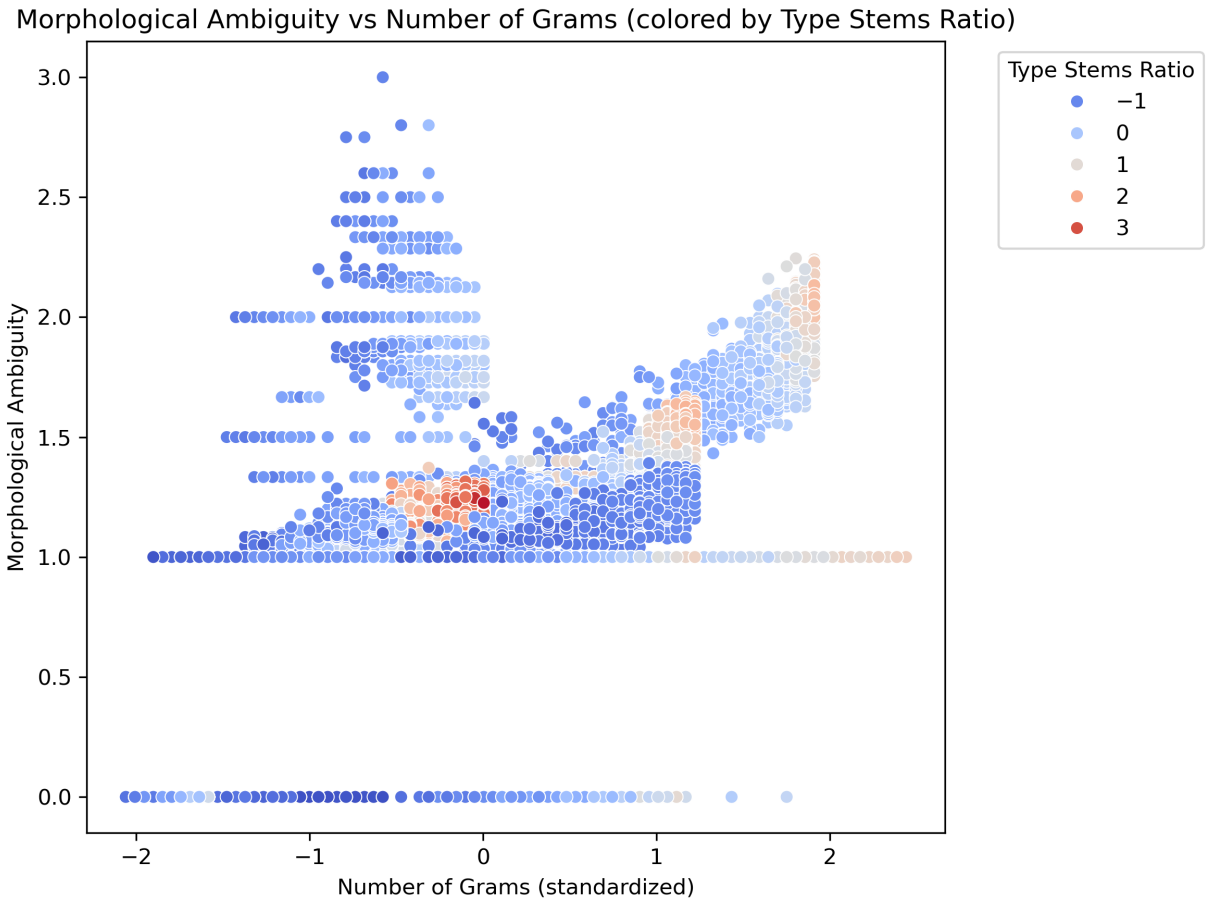


Figure 5.17: Interaction between *Number of Grams* and *Type Stems Ratio* on *Morphological Ambiguity* (including mni).

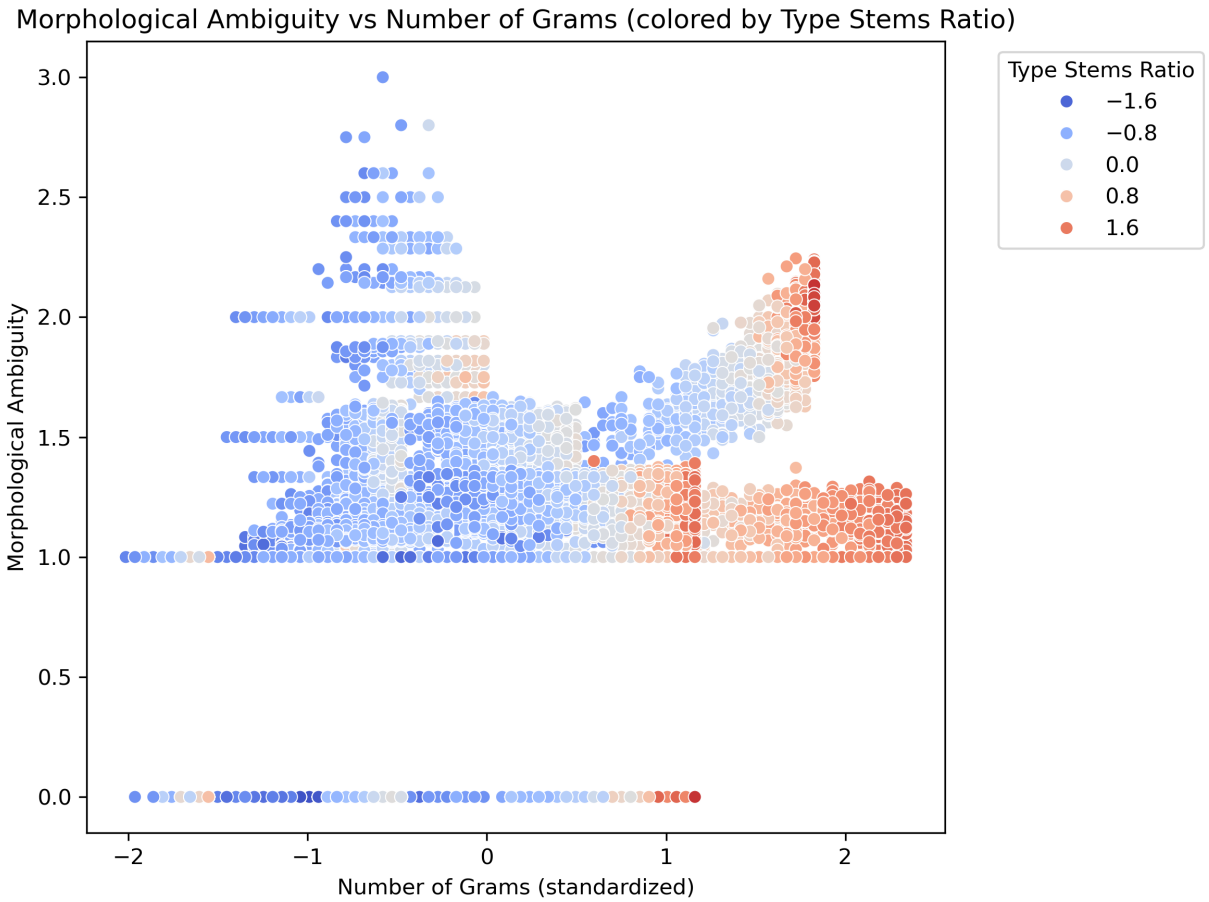


Figure 5.18: Interaction between *Number of Grams* and *Type Stems Ratio* on *Morphological Ambiguity* (excluding mni).

lead the system to instantiate a larger number of lexical rules. As more rules are generated to handle similar surface forms, *Morphological Ambiguity* increases due to overlapping or redundant mappings between forms and functions within the grammar.

Inference Time: Number of IGTs × Affix Ambiguity Ratio

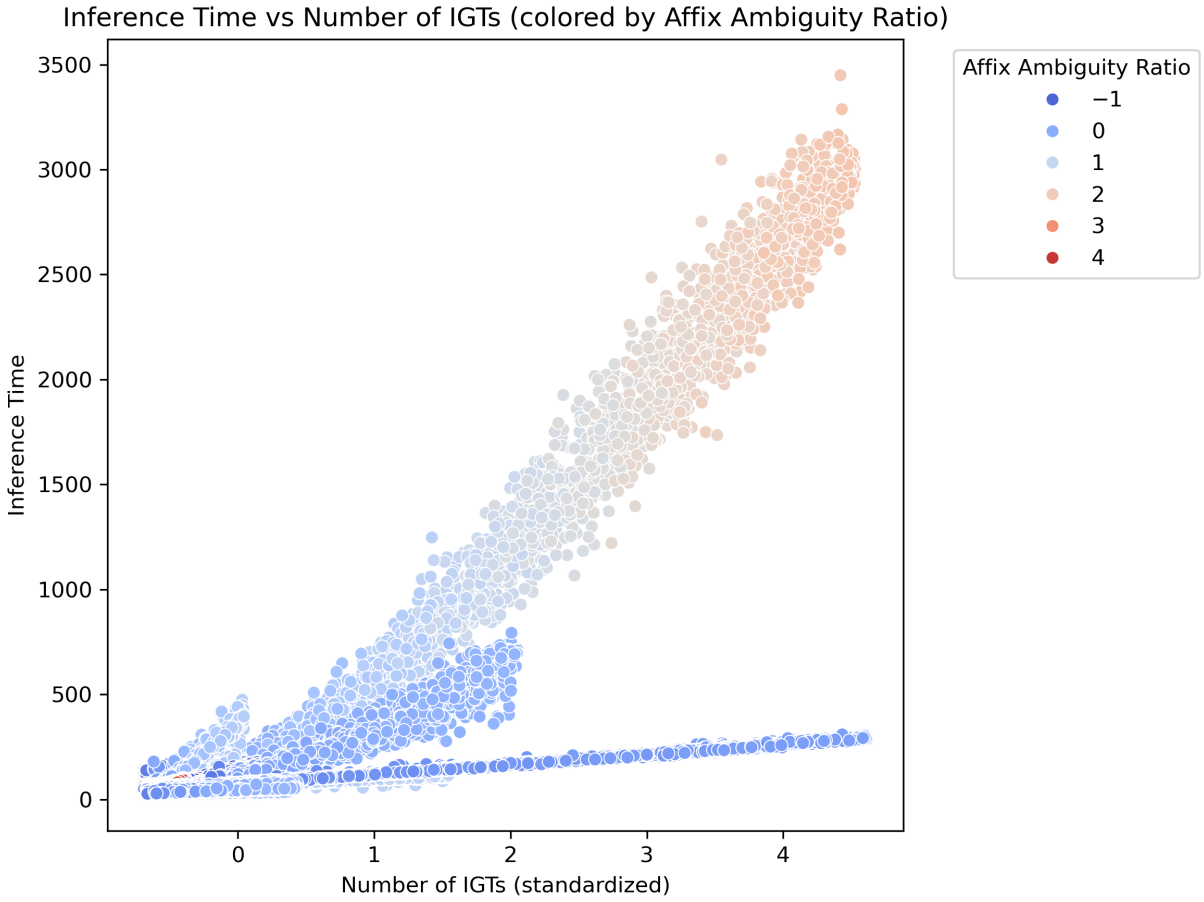


Figure 5.19: Interaction between *Number of IGTs* and *Affix Ambiguity Ratio* on *Inference Time*.

To examine how inference cost scales with both dataset size and morphological structure, I model *Inference Time* as a function of *Number of IGTs*, *Affix Ambiguity Ratio*, and their

interaction. Figure 5.19 plots standardized *Number of IGTs* (x-axis) against raw *Inference Time* (y-axis), with color indicating standardized *Affix Ambiguity Ratio* (blue = low, red = high).

The overall trend is linear: inference time rises with dataset size. However, the rate of increase depends on morphological ambiguity. For datasets with low affix ambiguity (blue lines), runtime grows slowly. In high-ambiguity contexts (red lines), inference time accelerates sharply. This pattern reflects a compounding interaction: ambiguity amplifies the processing cost of scale. Regression results mirror the visual pattern as previously reported.

Computationally, this effect demonstrates that ambiguity in affix-to-function mappings is not a fixed cost; instead, it scales with dataset size. Each ambiguous affix introduces multiple possible analyses, and as the number of IGTs increases, these ambiguities are repeated across more contexts. This leads to a larger hypothesis space for the system to account for during inference. As a result, inference time grows faster especially in morphologically ambiguous datasets.

5.6 Impact of POS Tag Source: Paired Analysis of Manual vs. Automatic Annotations

While previous sections focused on structural predictors of grammar quality across dataset, POS tag source represents a different type of variable. It does not alter the structural properties of the input data, such as morpheme counts, affix diversity, or sentence complexity, but instead affects the accuracy and consistency of the annotations used during grammar generation. To isolate its effect, I conduct a controlled paired analysis that compares manual versus automatically projected POS tags within matched training and test conditions.

As described in Section 3.5.7, I created aligned data pairs across seven representative datasets, with each pair sharing the exact same IGTs, word segmentation, and train/test splits. One version used linguists manually assigned POS tags (Group B), while the other used POS tags automatically provided by the INTENT (Group C). Since only the POS labels differed, any observed performance differences are attributable to the annotation quality, not

to structural variation.

To quantify this impact, I compute a delta score for each output metric on a per-sample basis:

$$\Delta = \text{Score}_{\text{manual}} - \text{Score}_{\text{INTENT}} \tag{5.1}$$

I measured the impact of POS source on three output metrics: *Coverage*: The proportion of test items that were successfully parsed. *Morphological Ambiguity*: The ratio of total verb inflectional rule instances to the number of unique surface forms. A higher value indicates greater rule overlap and surface ambiguity in the generated grammar. *LIAR (Logarithmic Inverse Average Readings)*: A metric of parse-time ambiguity, computed as $\frac{1}{\log(\text{Average Readings per Sentence Parsed}+1)}$. Higher *LIAR* values indicate fewer readings per sentence and thus lower ambiguity.

5.6.1 Group-Level Results

Table 5.7 summarizes the aggregate differences across samples. On average, manual POS tags led to an increase in *Coverage* ($\Delta = +0.0078$), a rise in Morphological Ambiguity ($\Delta = +0.090$), and a corresponding improvement in *LIAR* ($\Delta = +0.214$). All effects were statistically significant ($p < 0.001$), with moderate effect sizes for coverage and Morphological Ambiguity.

Metric	**Mean** Δ	**Std. Dev**	**Paired** t p	**Wilcoxon** p	**Cohen's** d
Coverage	0.0078	0.0110	< 0.001	< 0.001	0.71
Morph. Ambiguity	0.0901	0.1863	< 0.001	< 0.001	0.48
LIAR	0.2138	0.6262	< 0.001	< 0.001	0.34

Table 5.7: Paired comparison of grammar quality metrics: Manual vs. INTENT POS tags

These results show that manual POS tags increase the total rule overlap in the grammar,

as reflected in the higher Morphological Ambiguity. This suggests that richer or more detailed POS tags allow the system to create more rules that map to the same surface forms. While this raises ambiguity at the rule level, it also allows the grammar to accommodate a wider range of syntactic configurations during parsing.

The *LIAR* metric, which reflects parse-time ambiguity for test sentences, increases under manual POS annotations. Since higher *LIAR* values indicate fewer *Average Readings per Sentence Parsed*, this implies that the generated grammars, though more structurally complex, are actually more selective and precise during parsing. This contrast suggests that while automatic POS tags may reduce certain forms of syntactic noise or ambiguity during grammar generation, they do not provide sufficiently rich or consistent syntactic cues to support effective disambiguation during parsing.

5.6.2 Dataset-Specific Effects

Table 5.8 shows dataset-level Δ values for all three metrics. Datasets like wbl, aqn, and tsz show consistent improvement across all three metrics when using manual POS tags compared to INTENT-generated tags. In contrast, datasets such as tdh and ybh showed no measurable change, as neither version successfully parsed any test items (due to small test sets). Interestingly, mni exhibits a different case where *LIAR* decreased under manual tags, and I will explore a case study on it in the folllowing section.

Manual POS annotations consistently improve grammar quality in terms of both structural coverage and test-time disambiguation. Although manual POS tags lead to grammars with greater rule overlap for the same surface forms, this complexity appears beneficial. In short, higher ambiguity at the rule level does not necessarily imply worse parsing behavior, when guided by higher-quality annotations, the system is able to generate complex but effective grammars.

ISO CODE	Coverage Δ	Morph. Ambiguity Δ	LIAR Δ
aqn	0.0068	0.0442	0.6703
erk	0.0032	0.2670	0.4541
mni	0.0085	0.1012	−0.4424
tdh	0.0000	0.0000	0.0000
tsz	0.0079	0.1453	0.0305
wbl	0.0287	0.1237	1.1589
ybh	0.0000	0.0000	0.0000

Table 5.8: Dataset-level effects of POS source on grammar quality (Manual - INTENT)

5.7 Case Study Overview

In this case study, I evaluate the usability of a sample of AGGREGATION-generated grammars for the Meitei dataset through a combination of structural indicators and error patterns that serve as proxies for editing effort. Two potential use cases for AGGREGATION are for students learning grammar engineering or linguists seeking to construct a grammar based on a collection of IGTs. The AGGREGATION-generated grammar serves as a starting point, but users need to identify noise, clean up inconsistencies, and refine linguistic structures to develop a more accurate grammar. Through this analysis, I aim to provide insights into which types of datasets introduce more noise and which take advantage of AGGREGATION's strengths to generate a usable grammar with less effort. These findings will help linguists make informed decisions on how to best utilize AGGREGATION-generated grammars in their work.

I examine six grammar samples based on different dataset characteristics. The selected samples are drawn from two groups: Input Group B, which uses linguist-provided POS tags, and Input Group C, which uses INTENT-generated POS tags. I analyze three samples from each group: a high-resource grammar trained on all available IGTs (FullSet-

ManualPOS/FullSet-IntentPOS), a grammar that achieved the best overall performance across all metrics (BestCoverage-ManualPOS/BestCoverage-IntentPOS), and a low-resource grammar that reaches maximum parsing with minimal training data (SmallMaxCoverage-ManualPOS/SmallMaxCoverage-IntentPOS).

My evaluation includes both quantitative and qualitative methods. The quantitative analysis measures numeric coverage and ambiguity, while in the qualitative grammar assessment, I manually check specific linguistic issues such as affix misclassification, case affixes, determiners, person-number-gender categorization, errors in phrase structure, and additional cleaning effort required. By analyzing these factors, I identify the sample SmallMaxCoverage-ManualPOS as requiring minimal manual correction.

I find that larger datasets, such as FullSet-ManualPOS, result in more ambiguity. INTENT-generated POS tags help reduce ambiguity, but linguist-provided tags lead to better coverage. The smallest dataset, SmallMaxCoverage-ManualPOS, performs surprisingly well in parsing accuracy despite having the fewest IGTs. This suggests that larger dataset size alone does not guarantee better results.

My findings suggest that starting with a smaller, cleaner dataset may be more effective than using a large, noisy dataset. Linguist-provided POS tags improve coverage, while INTENT-generated tags produce cleaner morphological structures. Linguists may need to balance these factors depending on their specific goals. In the following sections, I provide a detailed breakdown of the quantitative and qualitative analysis with specific examples illustrating the challenges and insights gained from this study.

5.7.1 Samples Selection

I selected six grammar samples to examine how different input dataset characteristics influence AGGREGATION's performance. The selection was based on two factors: the *Number of IGTs* and the performance metrics associated with each grammar. Three samples come from Input Group B, which uses linguist-provided POS tags, and three from Input Group C, which uses INTENT-generated POS tags. To enable a direct comparison, I grouped the

six samples into three matched pairs based on training size and grammar quality.

The first pair, FullSet-ManualPOS and FullSet-IntentPOS (originally indexed as `mni-798`), represent grammars trained on the full 1596 IGTs available in the Meitei dataset, using manual and INTENT POS tags respectively.[1] The second pair, BestCoverage-ManualPOS and BestCoverage-IntentPOS (`mni-1434`), were selected not for size but for performance: these achieved the highest coverage and lowest ambiguity among all 3000 sampled grammars under their respective POS settings. The third pair, SmallMaxCoverage-ManualPOS and SmallMaxCoverage-IntentPOS (`mni-1256`), were drawn from training sets of only 295 IGTs, the smallest size that still allowed for at least one successful parse. Among all such minimal training runs, these samples achieved the highest parse count.

Table 5.9 summarizes the input and output statistics of these samples.

[1]The index number (e.g., 798) comes from a random draw among the 3000 sampled grammars. Its numerical relationship to the full IGT count ($798 \times 2 = 1596$) is purely coincidental.

Variable	FullSet Manual	BestCov Manual	SmallMax Manual	FullSet Intent	BestCov Intent	SmallMax Intent
Input Variables						
Number of IGTs	1596	451	295	1596	451	295
Distinct Stems	1378	642	563	1378	642	563
Distinct Affix Tokens	529	306	261	529	306	261
Distinct Word Types	5147	1848	1470	5147	1848	1470
Number of Grams	42	31	36	42	31	36
Avg. Length (words)	8.90	8.50	9.62	8.90	8.50	9.62
Avg. Length (morphemes)	19.30	18.80	21.25	19.30	18.80	21.25
Type Stems Ratio	3.74	2.88	2.61	3.74	2.88	2.61
Allomorph Ratio	1.67	1.31	1.26	1.67	1.31	1.26
Affix Ambiguity	1.28	1.08	1.03	1.28	1.08	1.03
Output Variables						
Coverage Count	5	5	5	3	2	2
Avg. Readings	8.60	1.20	2.20	1.00	1.00	1.00
Maximum Readings	36	2	6	1	1	1
Noun Types	277	144	124	39	22	23
Verb Types	88	45	42	23	15	13
Avg. Noun Stems	1.52	1.32	1.39	2.33	1.32	1.83
Avg. Verb Stems	2.73	2.36	2.07	2.52	1.53	1.54
Noun Inflections	42	26	21	29	16	17
Verb Inflections	56	30	38	27	22	17
Avg. Noun Lex Rule Types	1.21	1.15	1.38	1.17	1.25	1.12
Avg. Verb Lex Rule Types	1.32	1.57	1.24	1.85	1.45	1.88
Morphological Ambiguity	1.17	1.09	1.12	1.09	1.03	1.07

Table 5.9: Statistics of Meitei Grammar Samples for Case Study

5.7.2 Quantitative Analysis

Sample ID	**A Readings**	**B Readings**	**C Readings**	**D Readings**	**E Readings**
FullSet-ManualPOS	36	1	2	2	2
BestCoverage-ManualPOS	2	1	1	1	1
SmallMaxCoverage-ManualPOS	6	1	1	1	2
FullSet-IntentPOS	0	1	1	0	1
BestCoverage-IntentPOS	0	1	1	0	0
SmallMaxCoverage-IntentPOS	0	1	1	0	0

Table 5.10: Meitei (mni) Case Study Parsing Results

The parsing results, shown in Table 5.10, show that grammars from the ManualPOS group generally achieved higher coverage by successfully parsing more IGTs than their counterparts from the IntentPOS group. However, they also exhibited higher ambiguity with multiple possible readings for specific inputs.

Among the samples, FullSet-ManualPOS exhibited the highest level of ambiguity. It generated an average of 8.6 readings per parsed sentence, with a maximum of 36 in sentence A. The diversity in lexical types and inflections inferred from a large *Number of IGTs* can contribute to increased ambiguity.

In contrast, BestCoverage-ManualPOS demonstrated a balance between coverage and ambiguity despite being trained on only 451 IGTs. Compared to SmallMaxCoverage-ManualPOS, which has a higher average ambiguity and was trained on fewer IGTs, BestCoverage-ManualPOS exhibited lower ambiguity. This suggests that dataset size alone is not the most critical factor in balancing accuracy and ambiguity in AGG-generated grammars. However, given the small number of samples, this pattern may be influenced by randomness. The complexity of IGT structures in the dataset could be a contributing factor, as SmallMaxCoverage-ManualPOS

has a higher *Average IGT Length in Words* than BestCoverage-ManualPOS (see Table 5.9). This indicates that more complex sentence structures in the training data could lead to increased ambiguity in the generated grammar.

The smallest dataset, SmallMaxCoverage-ManualPOS, achieved maximum coverage with only 295 IGTs, just 18% of the full dataset. This highlights that it is possible to generate compact and effective grammars when given a carefully selected subset of data. While this result partly reflects the advantage of retrospective selection (i.e., identifying the subset that gives the best coverage with minimal input), it also reveals a deeper insight: the remaining 82% of the dataset, in this context, likely contributes more noise than useful grammatical phenomena. This points to the substantial potential of dataset refinement: selecting smaller, structurally coherent subsets can lead to more usable grammars with less post-editing effort.

Comparing the ManualPOS and IntentPOS groups in the Meitei case study reveals a pattern that contrasts with the general trend observed across other datasets. In all other datasets, as shown in Section 5.6.2, linguist-provided POS tags improved both coverage and disambiguation, as evidenced by positive Δ values in both *Coverage* and *LIAR* (e.g., aqn, wbl, and tsz in Table 5.8). These results suggest that manual annotations generally lead to more structurally expressive and accurate grammars, even when they increase morphological rule complexity.

However, Meitei behaves differently. While manual POS tags did increase coverage, they also led to higher morphological ambiguity and a decrease in *LIAR*. This diverges from the broader trend and suggests that the POS tagging scheme used in the manual Meitei annotations may introduce inconsistencies or overspecifications. Another possibility is that at least some of the ambiguity is legitimate, which reflects actual grammatical complexity rather than annotation artifacts. To investigate this divergence, I turn to a detailed qualitative analysis of the Meitei samples and focus on the cleaning effort required to reduce ambiguity, and assess whether the increased complexity under ManualPOS reflects useful grammatical structure or annotation-driven noise.

5.7.3 Qualitative Analysis

The qualitative analysis involved the examination of AGGREGATION-generated grammars to identify noises and errors and assess the extent of manual correction required. This evaluation focused on affix misclassification, case affix inconsistencies, determiner misidentification, person-number-gender (PNG) representation, and errors in phrase structure. Finally, I summarized the level of manual effort required to address these specific phenomena across different samples, using them as proxies for the broader cleaning demands of each sample.

Affix Misclassification

Affix misclassification emerged as a common issue, particularly in samples with larger training datasets. In these samples, AGGREGATION frequently assigned the same affix to multiple grammatical categories, which led to inflated ambiguity. While some affixes in Meitei are legitimately part of both nominal and verbal morphology, others were incorrectly duplicated across categories. To ensure a fair assessment, I manually inspected duplicated affix forms to determine whether their presence in multiple grammatical roles reflected actual functional overlap. I found no clear cases of such valid multifunctional use in the samples I checked, so for this study, I assume that each affix form should belong to only one grammatical category.

Sample ID	Total Duplicated Affixes	Affixes in Both Categories	Average Number of PC per Affix
FullSet-ManualPOS	35	29	2.40
BestCoverage-ManualPOS	16	12	2.31
SmallMaxCoverage-ManualPOS	18	14	2.28
FullSet-IntentPOS	21	16	2.19
BestCoverage-IntentPOS	7	5	2.00
SmallMaxCoverage-IntentPOS	6	4	2.00

Table 5.11: Duplicated Affixes Across Categories

From Table 5.11, I confirmed that larger datasets, particularly FullSet-ManualPOS, contained the highest number of incorrect affix duplications, which require the most manual revision. Grammars generated from the IntentPOS group exhibited fewer duplications overall. While this might suggest that automated tagging helps mitigate affix misclassification, a more likely explanation is that INTENT-generated tags lead to fewer words being treated as noun or verb candidates during grammar induction. As a result, AGGREGATION creates fewer affixation rules in the MOM component, reducing opportunities for duplication. Nevertheless, dataset size had a more significant impact than tagging method: samples with fewer IGTs consistently contained fewer affix misclassifications, regardless of POS tagging source.

Case Affix Inconsistencies

The handling of case affixes also revealed inconsistencies across grammar samples. Meitei employs various case markers, including agentive (*-nə*), patient (*-pu*), locative (*-tə*), ablative (*-təgi*), genitive (*-ki*), associative (*-kə*), and instrumental (*-nə*). Since these markers all appear in the same position with respect to nominal stems, an ideal grammar would iden-

tify all seven case markers and place them in one position class. While AGGREGATION successfully identified some of these cases, it often placed them into different position classes.

Table 5.12 summarizes the correctness of case affix classification across different grammar samples. The table presents the number of correctly classified cases out of the total identified cases, along with their respective position class distributions.

Grammar Sample	**Correct / Total Identified Cases**	**Position Class Distribution**
FullSet-ManualPOS	4 / 7	Different position classes
BestCoverage-ManualPOS	3 / 3	Different position classes
SmallMaxCoverage-ManualPOS	3 / 3	Two in class 16, one in class 3
FullSet-IntentPOS	2 / 4	Different position classes
BestCoverage-IntentPOS	3 / 3	Different position classes
SmallMaxCoverage-IntentPOS	3 / 3	Different position classes

Table 5.12: Correctness of Case Affix Classification

In FullSet-ManualPOS, four out of seven cases were correctly assigned, each within a different position class. Smaller datasets, particularly SmallMaxCoverage-ManualPOS, displayed better generalization by placing two cases within the same position class. These results indicate that excessive training data may introduce noise rather than improving classification accuracy. Although INTENT-generated POS tags did not provide equivalent generalization to linguist-provided tags, they still enabled AGGREGATION to correctly identify three case affixes.

Determiner Misidentification

Meitei has two attested determiners, *čhí* "proximate" and *tú* "distal" (Chelliah, 1997), which function prenominally to mark deictic distance:

- *mí čhí kə́p-e* 'The (proximate) man cries.'
- *mí tú kə́p-e* 'The (distal) man cries.'

However, AGGREGATION consistently failed to classify these forms as determiners. Although both *čhí* and *tú* appear in the generated lexicons, they are not assigned to the determiner section. Instead, *čhí* is categorized as a noun and the predicate `_pdet_n_rel`, and *tú* is treated as a verb with predicate `_ddet_v_rel`. This misrepresentation complicates their use in syntactic structures that require determiners. In particular, FullSet-ManualPOS and FullSet-IntentPOS contained the greatest number of incorrect non-determiner entries for these forms, while smaller samples exhibited fewer such errors.

Meanwhile, the determiner sections across samples often contain forms that are not determiners. For instance, AGGREGATION incorrectly classified the following forms as determiners:

- *púm-nə-mək*, which functions as a quantificational adverb meaning "each" or "all";
- *ná*, which is an agentive case marker

These errors likely result from AGGREGATION's reliance on these information correspondences derived from English glosses, where quantifiers and case markers may resemble determiners. Since quantifiers in Meitei behave differently from their English counterparts, this heuristic leads to the misclassification of functionally unrelated forms.

Person-Number-Gender (PNG) Representation

The representation of PNG distinctions in pronouns was another phenomenon examined to evaluate the extent of manual refinement required. Meitei pronouns differentiate between first, second, and third person, as well as singular, plural, and dual number categories. However, I observed inconsistencies in pronoun categorization across samples, where some lexical entries were omitted, and PNG features were not properly assigned. While Meitei does not employ grammatical gender, AGGREGATION introduced unnecessary distinctions in some cases.

In the inferred grammar specifications, the number distinction was incomplete. The current grammar structure for all of the samples only included:

(6)

```
section=number
  number1_name=sg
```

To correct this, I needed to modify the specification to include all number categories:

```
section=number
  number1_name=sg
  number2_name=pl
  number3_name=dual
```

Additionally, the system misclassified all pronouns into only third and non-third categories, rather than distinguishing between first, second, and third persons. The system required a revision from `person=3-noun-3` to `person=1-2-3`.

Across all samples, AGGREGATION failed to capture the full range of pronouns. A total of nine pronouns were expected, but multiple errors were found. Table 5.13 summarizes the number of correctly identified, incorrectly assigned, and missing pronoun lexical entries across each grammar sample.

Dataset	**Correct Pronoun Entries**	**Incorrect Pronoun Entries**	**Missing Pronoun Entries**
FullSet-ManualPOS	2	2	7
BestCoverage-ManualPOS	2	0	7
SmallMaxCoverage-ManualPOS	2	1	7
FullSet-IntentPOS	2	0	7
BestCoverage-IntentPOS	2	0	7
SmallMaxCoverage-IntentPOS	2	0	7

Table 5.13: Pronoun lexical entries identified across samples

Although some smaller datasets performed slightly better, none of the grammars fully captured the range of pronouns with proper feature specifications. Manual correction was required to assign person and number features, as well as to complete the missing pronoun entries while also removing the misclassified ones.

Morphological Misclassification

In generated grammar, morphological misclassifications can lead to parses that resemble phrase structure errors. An issue identified in this case study was the misclassification of verbs as nouns, especially when certain affixes were incorrectly treated as nominalizers. These errors result in grammars that generate misleading parses, such as interpreting a verb phrase (VP) as a noun phrase (NP), which complicates post-editing and reduces overall grammar usability.

To evaluate this issue, I focused on the following IGT:

(7) *yáw-ti* *čiŋ-lə* *čə́t-khi-lə-e*
drag-DLMT drag-PERF go-STILL-PERF-ASRT

'Dragging that, he left.'

The parsing results for this sentence varied across samples:

Sample ID	**A Readings**
FullSet-ManualPOS	36
BestCoverage-ManualPOS	2
SmallMaxCoverage-ManualPOS	6
FullSet-IntentPOS	0
BestCoverage-IntentPOS	0
SmallMaxCoverage-IntentPOS	0

Table 5.14: Parsing Results for Example 7

The example was the verb *yáw* ("to participate"), which AGGREGATION incorrectly classified as a noun and led to the inference that the suffix *-ti* attaches to a noun rather than a verb. According to (Chelliah, 1997), *yáw* can only be a verb, and the nominalization affix *-pə* is needed to form an NP, as in *yáw-pə=ti*.

The impact of these misclassifications varied across datasets. The largest dataset, FullSet-ManualPOS, produced 36 possible readings, but only four were reasonable. BestCoverage-ManualPOS generated two readings, all of which were incorrect due to misinterpreting *yáw-ti* as an NP rather than a VP. In contrast, SmallMaxCoverage-ManualPOS produced six readings, all of which were valid, which reinforced the observation that a smaller but well-structured dataset can lead to better parsing accuracy.

The grammars from INTENT-tagged datasets failed to parse this sentence and produced zero readings. This failure is due to the absence of lexical rules for *-ti* and prevented the formation of the expected VP structure. While INTENT-generated POS tags impose stricter constraints and reduce ambiguity, they also reduce coverage.

Overall Cleaning Effort

The final aspect of the qualitative assessment involved evaluating the overall cleaning effort required to refine these samples. Larger datasets consistently required more extensive corrections, with FullSet-ManualPOS and FullSet-IntentPOS containing more redundant lexical entries, incorrectly categorized affixes, and fragmented morphological structures. Some entries required cross-referencing linguistic literature to determine their proper classification and added to the post-processing workload. Additionally, AGGREGATION frequently placed affixes in morphology sections without associating them with specific syntactic phenomena.

Smaller datasets required less extensive revisions, though all samples needed manual inspection of position classes and lexical entries. However, these additional cleaning efforts were notably higher in FullSet-ManualPOS and FullSet-IntentPOS compared to the samples trained with fewer IGTs. This finding suggests that increasing the dataset size does not

necessarily lead to better grammar quality, as excessive data introduces noise. These insights inform my recommendations in the next section.

5.7.4 Case Study: Recommendations

Based on my findings in the case study, I provide the following recommendations for linguists and researchers seeking to reducing post-processing efforts.

First, I recommend prioritizing smaller, high-quality datasets over larger, noisier ones. The evaluation of SmallMaxCoverage-ManualPOS demonstrated that a smaller but cleaner-structured dataset with consistent annotations can achieve comparable coverage with reduced ambiguity compared to larger datasets such as FullSet-ManualPOS. Larger and noisier datasets introduced more noise and required more manual correction and post-processing.

Second, the choice between linguist-provided and INTENT-generated POS tags depends on the specific goal of grammar development. Linguist-provided POS tags offer higher coverage by successfully parsing more IGTs. However, they also introduce greater ambiguity due to broader lexical categories and multiple interpretations. Conversely, INTENT-generated POS tags produce cleaner, more constrained grammars that reduce ambiguity but at the cost of lower coverage. From the case study samples, I found that INTENT-generated POS tags can capture linguistic phenomena similarly to linguist-provided ones. Therefore, if a linguist has a collection of IGTs but lacks POS tags, they can use INTENT-generated tags as a starting point to produce a comparable grammar.

Third, lexicon and morphological assignments should be carefully reviewed. In the case study, errors such as misclassifying the verb *yáw* as a noun led to incorrect phrase structures like interpreting *yáw-ti* as a noun phrase. These issues stemmed from incorrect lexical type assignments. Linguists should also examine affix assignments, especially where AGGREGATION assigns the same affix form to multiple lexical types.

Finally, the overall post-processing effort required to refine AGGREGATION-generated grammars is non-trivial, particularly for larger datasets. My findings indicate that grammars trained on smaller datasets required less manual intervention and produced more coherent

morphological structures. A practical strategy is to begin by running AGGREGATION on the full dataset to assess maximum achievable coverage, and then iteratively reduce the training set size (monitoring for sharp drops in coverage) to identify the smallest subset that still performs well. I will discuss more heuristics and inspirations for subset selection in the final chapter.

5.8 Summary

In this chapter, I reported a comprehensive set of findings based on 75,000 grammar generation and parsing runs of 25 datasets. I began by identifying structurally meaningful output metrics, then modeled their relationships with input features using both linear and non-linear approaches. A paired analysis of POS tag sources and a focused case study on the Meitei dataset further clarified how annotation quality and training data selection shape grammar usability. Below, I summarize the key findings.

I evaluated 15 output metrics for reliability and susceptibility to test set artifacts. Metrics such as *Coverage Count*, while informative, were rejected due to their dependence on test set size. Instead, I selected four representative metrics with low rank consistency and minimal correlation to evaluation inputs: *Coverage* (coverage), *LIAR* (ambiguity), *Morphological Ambiguity* (grammar complexity), and *Inference Time* (efficiency).

I used scatterplot grids to visualize how structural features correlate with each output metric across the 25 datasets. These plots reveal consistent, interpretable trends that reinforce the univariate modeling results. For example, parsing *Coverage* improves with higher grammatical diversity, as evidenced by positive slopes for *Number of Grams* and *Distinct Affix Types*. Similarly, *LIAR* and *Morphological Ambiguity* decline and rise respectively in line with *Allomorph Ratio*, *Type Stems Ratio*, and related features, confirming that increased morphological complexity simultaneously enhances coverage and increases ambiguity.

The scatterplot grids also highlight typological outliers. For instance, datasets like mni and tsz exhibit steeper trends due to their dense morphological inventories, while datasets such as wbl and wmb compress complexity more efficiently, leading to atypical slopes or

even reversals. In addition, marginal effect plots from the linear models help isolate the contribution of each predictor while controlling for dataset-level variation, visualizing subtle effects such as diminishing returns and performance plateaus.

To capture the joint influence of structural predictors, I fit multivariate models with interaction terms. Stepwise AIC selection consistently gave models that significantly outperformed univariate baselines, confirming the importance of feature combinations. The interaction for *Coverage* between *Distinct Affix Types* and *Affix Ambiguity Ratio* demonstrates diminishing returns: affix diversity improves coverage only when ambiguity remains low.

For *LIAR*, two interactions emerged. First, the combination of *Allomorph Ratio* and *Type Stems Ratio* sharply increased ambiguity, pointing to a form-function overload effect. Second, a similar pattern with *Number of Grams* revealed that ambiguity intensifies in grammatically rich environments when allomorphy is high. *Morphological Ambiguity* also showed a strong interaction: as both *Number of Grams* and *Type Stems Ratio* rise, lexical rule redundancy compounds, increasing ambiguity.

A substantial interaction was observed for *Inference Time*: parsing cost scales quickly with both *Number of IGTs* and *Affix Ambiguity Ratio*. This indicates that training data volume and morphological complexity jointly expand the search space.

A paired analysis comparing manual and INTENT-generated POS tags revealed that manual tags increased both structural ambiguity and test-time disambiguation. Coverage improved by +0.0078, morphological ambiguity by +0.0901, and *LIAR* by +0.214. These effects were statistically significant, suggesting that manually assigned syntactic categories better support generalization despite introducing more rule overlap. However, exception in Meitei dataset showed that POS quality interacts with dataset-specific structure.

The Meitei analysis highlighted how dataset size and annotation quality interact to affect grammar usability. SmallMaxCoverage-ManualPOS, trained on only 295 IGTs, achieved full coverage with lower ambiguity and minimal cleaning effort. Larger datasets like FullSet-ManualPOS generated more affix duplication, phrase structure errors, and annotation-driven

overgeneration. INTENT-generated POS tags produced cleaner grammars with fewer errors but reduced parse success. Specific findings include:

- **Affix misclassification**: Large datasets (especially manual) exhibited frequent redundant affix reuse across categories, requiring manual correction.
- **Case affix inconsistencies**: AGGREGATION often split attested case markers across multiple position classes; smaller datasets performed better.
- **Determiner misidentification**: Forms like *čhí* and *tú* were consistently misclassified or omitted, replaced by unrelated items.
- **PNG representation**: No sample captured the full pronoun system; number and person distinctions were underspecified or absent.
- **Phrase structure errors**: Misclassification of verbs as nouns led to incorrect VP->NP structures, especially in FullSet-ManualPOS.

Based on these findings, I recommend (1) prioritizing small, clean datasets over large, noisy ones; (2) selecting POS tag sources based on intended grammar use: manual tags for generalization, automatic tags for clarity; (3) reviewing lexicon and affix categories to prevent structural noise; and (4) treating AGGREGATION-generated grammars as draft grammars that benefit from systematic refinement.

Taken together, the results of this chapter show that grammar generation quality is not solely a function of data size or modeling complexity. Instead, it is shaped by a complex interplay of input structure, annotation quality, and system constraints, which must be jointly considered in linguistic applications of AGGREGATION.

Chapter 6
DISCUSSION AND CONCLUSION

In this chapter, I reflect on the implications and limitations of the results presented in the previous chapter, with a focus on how they inform practical decisions in data preparation for grammar generation with AGGREGATION. I revisit the original research question from a broader perspective: what structural and annotation-related properties of IGT corpora are most conducive to producing effective, usable grammars, particularly under real-world constraints of limited time and annotation resources?

I organize the discussion into five parts. In Section 6.1, I revisit the research question and summarize the key takeaways to offer concrete guidance for linguists working with either existing corpora or newly collected IGTs. Section 6.2 addresses a central analytical limitation of the study: the distinction between predictive but uncontrolled proxy variables, and those structural features of the data that can be directly influenced during corpus preparation. In Section 6.3, I discuss the methodological constraints of the modeling framework. Finally, Section 6.4 outlines directions for future work, including typological stratification and system-level improvements.

6.1 Revisiting the Research Question

In this thesis, I set out to answer a practical question faced by linguists who work with IGT corpora under resource constraints: how best to prepare data for use with the AGGREGATION system. I focused on identifying which characteristics of input datasets most directly impact the system's ability to produce grammars that are usable, interpretable, and efficient to refine. The question assumes a realistic scenario in which data are originally collected for descriptive or typological research, but later repurposed for grammar generation.

The results suggest a clear answer: annotation consistency is more important than dataset size. While it might be assumed that larger corpora would lead to better grammars, I found that this is not necessarily the case. In fact, grammars trained on smaller, more internally coherent datasets (those with moderate sentence complexity, repeated stems, and low *Affix Ambiguity*) often outperformed those trained on larger and noisier corpora. For example, in the Meitei case study, a dataset of just 295 IGTs (SmallMaxCoverage) produced better coverage and required less manual correction than its full-dataset counterpart, although it was strategically selected to maximize training set coverage while maintaining minimal size.

This finding highlights a broader principle: AGGREGATION benefits more from *learnable structure* than from raw data volume. As shown in Sections 5.5 and 5.5.3-5.5.3, features such as the *Type Stems Ratio* and *Affix Ambiguity* were more important of grammar quality than the total *Number of IGTs*. Based on this, I argue that linguists preparing data for grammar generation should focus on maximizing morphological regularity, lexical repetition, and structural clarity, even if that means working with a smaller subset of their available data. And perhaps there is room for developing tools that could help identify and select such high-quality subsets semi-automatically, using structural metrics as guidance.

I also examined the role of POS annotation by comparing grammars trained on manually assigned tags with those trained on tags generated by the INTENT system. The results showed a consistent trade-off: manual tags improved both coverage and disambiguation, but at the cost of increased morphological ambiguity. Grammars generated from INTENT-generated tags were more compact and less ambiguous at the rule level, but they parsed fewer test sentences successfully. This suggests that manual POS tags, while introducing greater category overlap, provide more reliable cues for generalization. In contrast, automatic tags support cleaner rule sets but are less effective for coverage-oriented grammar generation. These findings point to a functional distinction: if the user's goal is to produce a grammar that performs well across a wide range of sentence and unseen data, manually provided tags are preferable; if the goal is to produce a minimal, low-disambiguation grammar for targeted use or demonstration, automatic tags may suffice. However, this pattern

does not hold universally. In the Meitei case study, manually tagged data introduced more structural inconsistencies and required greater post-editing effort, suggesting that the benefits of manual POS annotation depend not only on annotation granularity but also on the dataset's morphological profile and the consistency of tag use.

6.2 Proxy Variables and Structural Predictors

One limitation of the modeling approach used in this study is that not all predictive features provide equally actionable insight for users preparing IGT data. Some variables, while strongly associated with grammar performance, primarily reflect global properties of the corpus, such as size or lexical diversity, without pointing clearly to what aspects of the data can or should be modified. I refer to these as *diagnostic variables* or *proxy variables*, since they may help explain model behavior but do not offer concrete guidance for data intervention. For example, features such as *Number of Distinct Word Types*, and *Number of Distinct Affix Types* showed consistent associations with output metrics like *LIAR* and *Inference Time*, but these effects are largely driven by corpus size or descriptive richness.

In contrast, other features, such as the *Type-Stems Ratio* and *Affix Ambiguity*, encode structural patterns in the data that linguists can influence when preparing or selecting training IGTs. These are what I refer to as *structurally actionable predictors*. They are interpretable in linguistic terms (e.g., lexical redundancy, affix distinctiveness, morphosyntactic complexity), and they offer practical value to linguists seeking to refine training corpora. For example, a high *Type-Stems Ratio* indicates excessive lexical sparsity, which undermines generalization; a high *Affix Ambiguity* may result from overlapping grammatical functions or from inconsistent glossing, which can be mitigated through more consistent annotation practices.

It is worth clarifying that corpus-level variables like size or token counts are, in principle, controllable. For example, by selecting a smaller training subset from a larger dataset. However, their effects on system's performance often stem from interactions with deeper structural properties, rather than being directly interpretable or optimizable on their own.

The distinction here is therefore not about control per se, but about whether a variable supports targeted, linguistically motivated decisions during corpus preparation.

Overall, while proxy variables help explain why certain datasets lead to better results, it is the structurally actionable predictors that offer a path forward for practical intervention. Users aiming to optimize AGGREGATION performance should therefore focus on modifying variables that reflect internal structural coherence (such as reducing *Affix Ambiguity* that arises from inconsistent glossing) rather than simply increasing dataset size or lexical diversity. Recognizing which variables are diagnostic and which are actionable is important for making informed, effective decisions during data preparation.

6.3 Modeling Limitations and Interpretive Boundaries

The multivariate models developed in this study demonstrate that combinations of structural features meaningfully shape grammar performance across a wide range of low-resource datasets. However, these models are not exhaustive. A key limitation lies in the restricted scope of interaction terms explored: only a number of two-way interactions were tested, each selected based on linguistic plausibility and strong marginal effects observed in earlier analyses. While this targeted approach improves interpretability, it likely omits other interactions that may be statistically and linguistically significant.

Indeed, stepwise feature selection revealed that five-variable models outperformed simpler baselines for all four output metrics, with substantial AIC reductions. This suggests that the explanatory utility of the predictors extends beyond the specific pairs selected for interaction modeling. In other words, multiple structural features likely interact in shaping grammar performance: not just in isolated pairs, but potentially in more complex, higher-order combinations.

Nevertheless, the current analysis is limited to second-order (two-way) interactions. This decision reflects practical tradeoffs: higher-order interaction models are difficult to interpret, require more data to estimate reliably, and risk overfitting in a relatively small and heterogeneous dataset. From a linguistic standpoint, many phenomena are inherently mul-

tidimensional. For instance, the effect of *Affix Ambiguity* might not only depend on the number of affixes, but also on sentence length, stem diversity, or morphological alignment all at once. Such third- or fourth-order interactions are beyond the scope of the present framework.

Future work could address these limitations by systematically exploring higher-order interactions using techniques such as hierarchical modeling (Gelman and Hill, 2006), tensor factorization (Kolda and Bader, 2009), or even nonparametric approaches like GAMs (Generalized Additive Models; Hastie 2017). These methods may help uncover potential dependencies among structural variables that shape grammar usability in subtle but important ways.

Initially, I conducted SHAP (SHapley Additive exPlanations; Lundberg and Lee 2017) analysis on the trained XGBoost model to explore potential feature dependencies and interactions. While this approach offered some insight into variable importance and local behavior, I ultimately chose to abandon it for the purposes of this study. The primary reason is that SHAP values reflect a model-specific decomposition of predictions (based on decision trees) rather than statistically interpretable correlations or causal effects. As a result, the observed patterns, while suggestive, do not generalize beyond the trained model and may mislead interpretation if treated as standalone evidence.

Further research should take this risk into account when integrating machine learning-based interpretation methods. While these techniques can complement statistical modeling, they should be used with caution, especially in contexts where interpretability and generalizability across these small size datasets are essential.

Beyond modeling strategy, there are also interpretive limits in the output metrics themselves. Although I selected four metrics (*Coverage*, *LIAR*, *Morphological Ambiguity*, and *Inference Time*) to balance generalization, ambiguity, complexity, and efficiency, these measures are not orthogonal. In particular, I observed an inherent tension between coverage and ambiguity: grammars that achieved high coverage often did so by generalizing more aggressively, at the cost of introducing morphosyntactic ambiguity and rule redundancy. This

was especially evident in the comparison of `SmallMaxCoverage` versus `FullSet` grammars, where the latter covered more structures but required substantially more post-editing.

This tension highlights a broader point: grammar quality cannot be reduced to a single metric. Different users will weigh coverage, ambiguity, interpretability, and computational cost differently, depending on their goals. A field linguist working on a lesser-described language may prioritize coverage above all, using AGGREGATION to assess how much of their corpus can be accounted for by a draft grammar, even if the result is structurally ambiguous. A computational linguist preparing a demonstration may prefer grammars with low ambiguity and minimal rule complexity. A typologist comparing syntactic structures across languages may prioritize rule interpretability, seeking grammars that expose meaningful structural contrasts rather than those that maximize parse coverage. The models presented here offer insight into how structural features shape these tradeoffs, but they cannot determine the "best" grammar in absolute terms. Ultimately, grammar generation must be evaluated not just on how much it covers, but on how usefully it structures and limits that coverage.

6.4 Future Work

A key limitation of this study is that it treats all IGT datasets as structurally comparable datapoints, implicitly assuming that structural predictors function similarly across datasets. In reality, linguistic structure is shaped by typological variation, differences in morphological complexity, word order, alignment systems, etc., that are likely to interact with the performance of grammar generation systems in meaningful ways.

For example, *Affix Ambiguity* may have very different implications in agglutinative languages, where affix boundaries are regular and affix stacking is common, than in fusional languages, where morpheme segmentation is less transparent. Likewise, average morpheme count or sentence length may indicate syntactic complexity in one language but merely reflect orthographic conventions or glossing practices in another. Without accounting for these distinctions, the models presented here risk conflating actual structural effects with

language-specific artifacts.

Typological stratification offers a promising direction for future work. By grouping languages according to features, such as morphological type (e.g., agglutinative, isolating, etc.), or inflectional richness, it would be possible to examine whether the effects of structural predictors are consistent within types and diverge across them. This could be implemented by introducing typological groupings as fixed effects or interaction terms in mixed-effects models. Doing so would clarify whether certain data preparation strategies are more effective for specific language types, and it would improve the interpretability of predictors that currently behave differently across datasets.

In parallel, as mentioned, future studies could investigate higher-order interaction effects among structural features, especially within typologically homogeneous subsets.

In terms of the AGGREGATION system development, future versions could incorporate automated support for subset selection and multi-version grammar generation. Rather than relying on users to manually experiment with training data size and composition, the system could analyze structural predictors to identify promising training subsets. It could then train multiple grammars in parallel and present users with output metrics for each version, including coverage, ambiguity, and rule complexity. This would allow linguists to compare trade-offs directly and select the grammar best suited to their specific needs, whether for broad analysis, targeted parsing, or teaching purposes. Such a feature would translate the findings of this thesis into practical tooling.

Additionally, the system could support semi-automatic training set assistant by identifying high-quality subsets based on actionable structural metrics, such as low affix ambiguity, high lexical redundancy, or moderate sentence length. By surfacing these subsets with minimal user input, AGGREGATION could guide linguists toward more learnable data configurations, accelerating the path to usable grammars even in resource-constrained settings.

BIBLIOGRAPHY

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Bardagil Mas, B. (2019). Panara corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Bender, E., Drellishak, S., Fokkens, A., Poulson, L., and Saleem, S. (2010). Grammar customization. *Research on Language and Computation*, 8:23–72.

Bender, E. M. (2023). LING 567.

Bender, E. M., Crowgey, J., Goodman, M. W., and Xia, F. (2014). Learning grammar specifications from IGT: A case study of chintang. In Good, J., Hirschberg, J., and Rambow, O., editors, *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.

Bender, E. M., Flickinger, D., and Oepen, S. (2002). The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In Carroll, J., Oostdijk, N., and Sutcliffe, R., editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Bender, E. M., Goodman, M. W., Crowgey, J., and Xia, F. (2013). Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In Lendvai, P. and Zervanou, K., editors, *Proceedings of the 7th Workshop on Language*

Technology for Cultural Heritage, Social Sciences, and Humanities, pages 74–83, Sofia, Bulgaria. Association for Computational Linguistics.

Bender, E. M., Wax, D., and Goodman, M. W. (2012). From IGT to precision grammar: French verbal morphology. In *LSA Annual Meeting Extended Abstracts*.

Bickel, B., Gaenszle, M., Rai, A., Rai, S. K., Rai, V. S., and Gautam (Sharma), N. P. (2011). Audiovisual corpus of the Puma language, including paradigm sets, grammar sketches, ethnographic descriptions, and photographs. DOBES Archive.

Bickel, B., Stoll, S., Gaenszle, M., Rai, N. K., Lieven, E., Banjade, G., Bhatta, T. N., Paudyal, N. P., Pettigrew, J., Rai, I. P., Rai, M., Zakharko, T., and Schikowski, R. (2016). Audiovisual corpus of the Chintang language, including a longitudinal corpus of language acquisition by six children, paradigm sets, grammar sketches, ethnographic descriptions, and photographs.

Chelliah, S. L. (1997). *A Grammar of Meithei*. De Gruyter Mouton, Berlin, New York.

Chelliah, S. L. (2011). *A Grammar of Meithei*, volume 17. Walter de Gruyter.

Chelliah, S. L. (2019). Meithei texts. Manipur Digital Resources in UNT Digital Library. University of North Texas Libraries. (Accessed September 2023).

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Comrie, B., Haspelmath, M., and Bickel, B. (2008). *The Leipzig Glossing Rules: Conventions for Interlinear Morpheme-by-Morpheme Glosses*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Conrad, E. (2021). Tracing and reducing lexical ambiguity in automatically inferred grammars. Master's thesis, University of Washington.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. (2005). Minimal recursion semantics: An introduction. *Reseach On Language And Computation*, 3:281–332.

Crysmann, B. and Packard, W. (2012). Towards efficient HPSG generation for German, a non-configurational language. In Kay, M. and Boitet, C., editors, *Proceedings of COLING 2012*, pages 695–710, Mumbai, India. The COLING 2012 Organizing Committee.

Dods, A. (2022). *Automatically Inferring Grammar Specifications for Adnominal Possession from Interlinear Glossed Text*. University of Washington Libraries.

Donet, C. (2019). Lezgi corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Fisher, R. A. (1992). *Statistical Methods for Research Workers*, pages 66–70. Springer New York, New York, NY.

García-Laguía, A. (2022). Northern Alta corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Gelman, A. and Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.

Georgi, R. (2016). *From Aari to Zulu: Massively Multilingual Creation of Language Tools using Interlinear Glossed Text*. PhD thesis, University of Washington.

Goodman, M. W. (2013). Generation of machine-readable morphological rules from human-readable input.

Goodman, M. W., Crowgey, J., Xia, F., and Bender, E. M. (2015). Xigt: extensible interlinear glossed text for natural language processing. *Language Resources and Evaluation*, 49(2):455–485.

Hargus, S. (2019). Yakima sahaptin corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Harley, H. (2019). Haiki text corpus. University of Arizona. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Hastie, T. J. (2017). Generalized additive models. *Statistical models in S*, pages 249–307.

Hauk, B. (2019). Tsova-Tush lexicon and texts. Unpublished FieldWorks (FLEx) project. V2019.08.20. 2016–2019 (collection date). University of Hawai'i at Manoa.

Howell, K. (2020). *Inferring Grammars from Interlinear Glossed Text: Extracting Typological and Lexical Properties for the Automatic Generation of HPSG Grammars*. PhD thesis, University of Washington, Seattle.

Howell, K. and Bender, E. M. (2022a). Building analyses from syntactic inference in local languages: An HPSG grammar inference system. In Derczynski, L., editor, *Northern European Journal of Language Technology, Volume 8*, Copenhagen, Denmark. Northern European Association of Language Technology.

Howell, K. and Bender, E. M. (2022b). Building analyses from syntactic inference in local languages: An HPSG grammar inference system. *The Northern European Journal of Language Technology (NEJLT)*, 8(1).

Inman, D. (2019). Nuuchahnulth texts. University of Washington. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Joseph Lee Rodgers, W. A. N. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66.

Kaufman, D., Khujamyorova, H., and Perlin, R. (2020). Wakhi texts. Digital collection managed by KRATYLOS. Uploaded from www.elalliance.org, Wakhi. In Finkel, R. and Kaufman, D., Kratylos: Unified Linguistic Corpora from Diverse Data Sources. Uploaded

April 28, 2020 and retrieved from https://www.cs.uky.edu/raphael/ela/ on May 20, 2020.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Lahassois, A. (2019). Thulung corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Lewis, W. and Xia, F. (2008). Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008)*, pages 685–690, Hyderabad, India.

Lin, Y.-C. (2023). Automatically inferring grammar specifications for valence-changing verbal morphology from interlinear glossed text. Master's thesis, University of Washington. Thesis (Master's).

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

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

Paz, A., Camacho, L., and Kaufman, D. (2022). P' urhepecha corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Ruxton, G. D. (2006). The unequal variance t-test is an underused alternative to student's t-test and the Mann–Whitney U test. *Behavioral Ecology*, 17(4):688–690.

Schackow, D. (2022). Yakkha corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Schrock, T. (2019). Ik corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Shimelman, A. (2019). Yaoyos Quechua corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Spearman, C. (2010). The proof and measurement of association between two things. *International Journal of Epidemiology*, 39(5):1137–1150.

Stoll, S., Lieven, E., Banjade, G., Bhatta, T. N., Gaenszle, M., Paudyal, N. P., Rai, M., Rai, N. K., Rai, I. P., Zakharko, T., Schikowski, R., and Bickel, B. (2016). Audiovisual corpus on the acquisition of Chintang by six children.

Strunk, L. (2019). Yup' ik corpus. Unpublished FieldWorks (FLEx) project. (Accessed September 2023).

Thieberger, N. (2006a). Dictionary and texts in south efate. Digital collection managed by PARADISEC [Open Access]. (Accessed September 2023).

Thieberger, N. (2006b). *A Grammar of South Efate: An Oceanic Language of Vanuatu*, volume 33. University of Hawaii Press.

Wax, D. (2014). Automated grammar engineering for verbal morphology. Master's thesis, University of Washington, Seattle.

Xia, F. and Lewis, W. (2007). Multilingual structural projection across interlinear text. In Sidner, C., Schultz, T., Stone, M., and Zhai, C., editors, *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 452–459, Rochester, New York. Association for Computational Linguistics.

Xia, F. and Lewis, W. (2009). Applying NLP technologies to the collection and enrichment of language data on the web to aid linguistic research. In Borin, L. and Lendvai, P., editors, *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. Association for Computational Linguistics.

Xia, F., Lewis, W., Goodman, M., Slayden, G., Georgi, R., Crowgey, J., and Bender, E. (2016). Enriching a massively multilingual database of interlinear glossed text. *Language Resources and Evaluation*, 50(2):321–349.

Zamaraeva, O. (2016). Inferring morphotactics from interlinear glossed text: Combining clustering and precision grammars. In Elsner, M. and Kuebler, S., editors, *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 141–150, Berlin, Germany. Association for Computational Linguistics.

Zamaraeva, O., Curtis, C., Emerson, G., Fokkens, A., Goodman, M., Howell, K., Trimble, T., and Bender, E. M. (2022). 20 years of the Grammar Matrix: cross-linguistic hypothesis testing of increasingly complex interactions. *Journal of Language Modelling*, 10(1):49–137.

Zamaraeva, O., Howell, K., and Bender, E. M. (2019a). Handling cross-cutting properties in automatic inference of lexical classes: A case study of Chintang. In Arppe, A., Good, J., Hulden, M., Lachler, J., Palmer, A., Schwartz, L., and Silfverberg, M., editors, *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.

Zamaraeva, O., Howell, K., and Bender, E. M. (2019b). 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, Hawai‘i.

Zamaraeva, O., Kratochvíl, F., Bender, E. M., Xia, F., and Howell, K. (2017). Computational support for finding word classes: A case study of Abui. In Arppe, A., Good, J., Hulden, M., Lachler, J., Palmer, A., and Schwartz, L., editors, *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.

Appendix A

APPENDIX

A.1 Supplementary Scatterplot Grids: Next 5 Predictors

This appendix presents dataset-level scatterplot grids for the next 5 (least important) structural predictors (ranked 6-10) for each output metric. These plots follow the same layout and interpretation conventions as those described in Section 5.4, with each row corresponding to a dataset and each column representing one predictor variable.

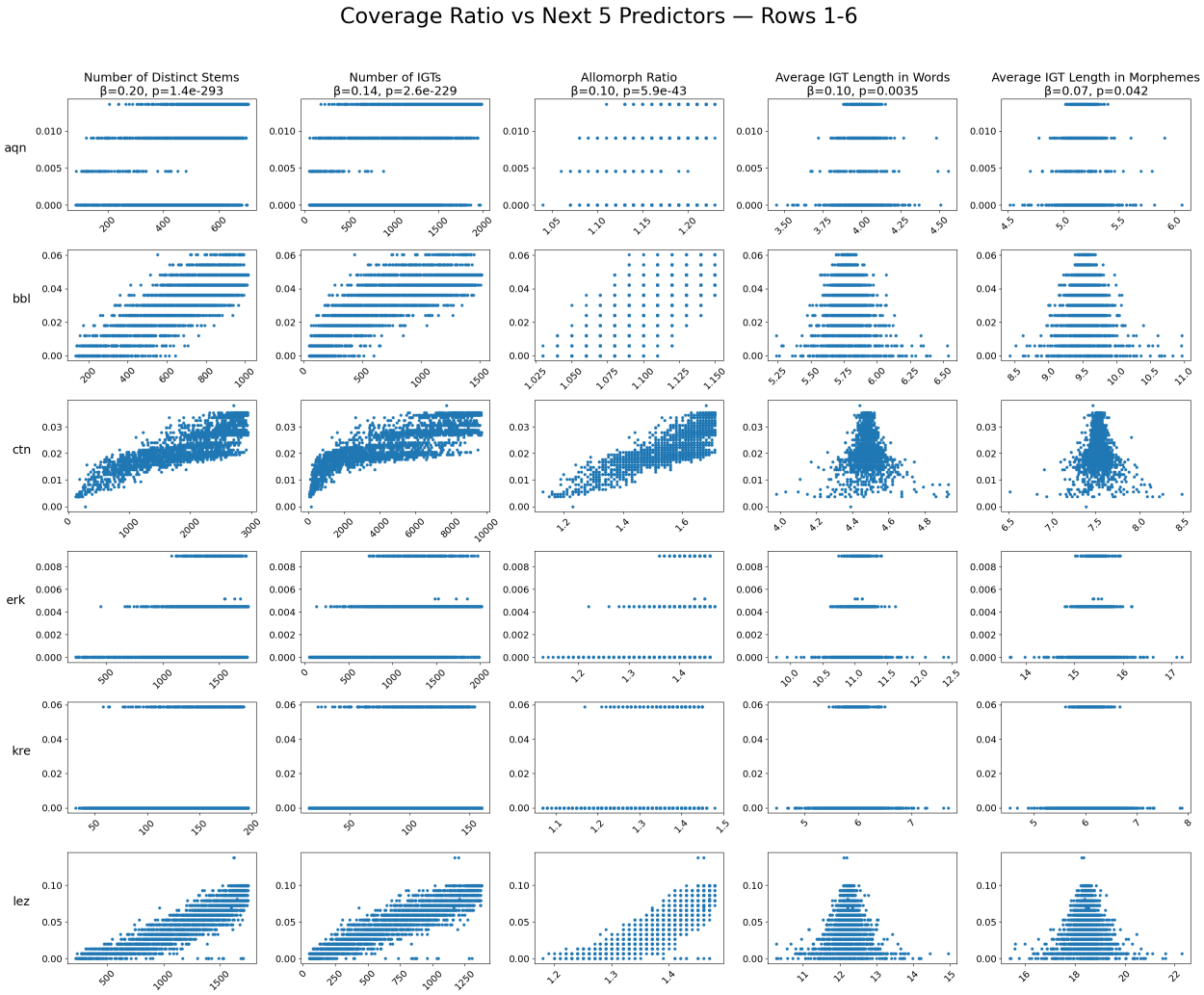


Figure A.1: Scatterplot grid for *Coverage Ratio* vs next 5 structural predictors (Rows 1-6).

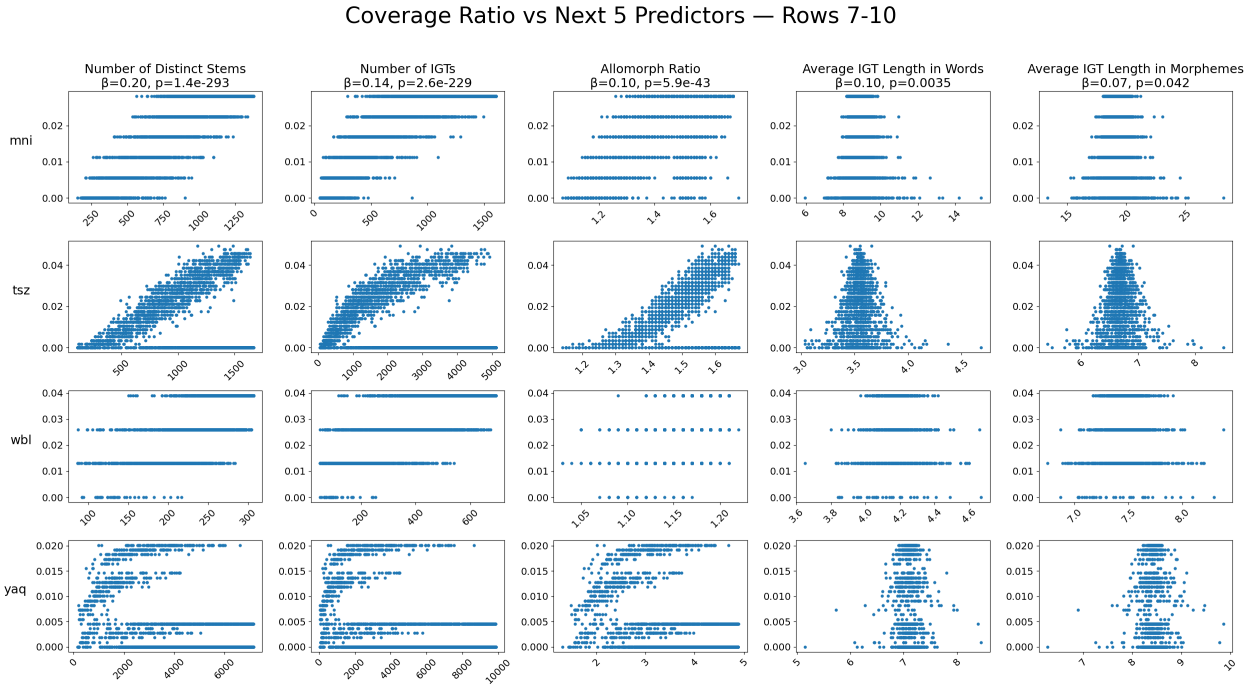


Figure A.2: Scatterplot grid for *Coverage Ratio* vs next 5 structural predictors (Rows 7-12).

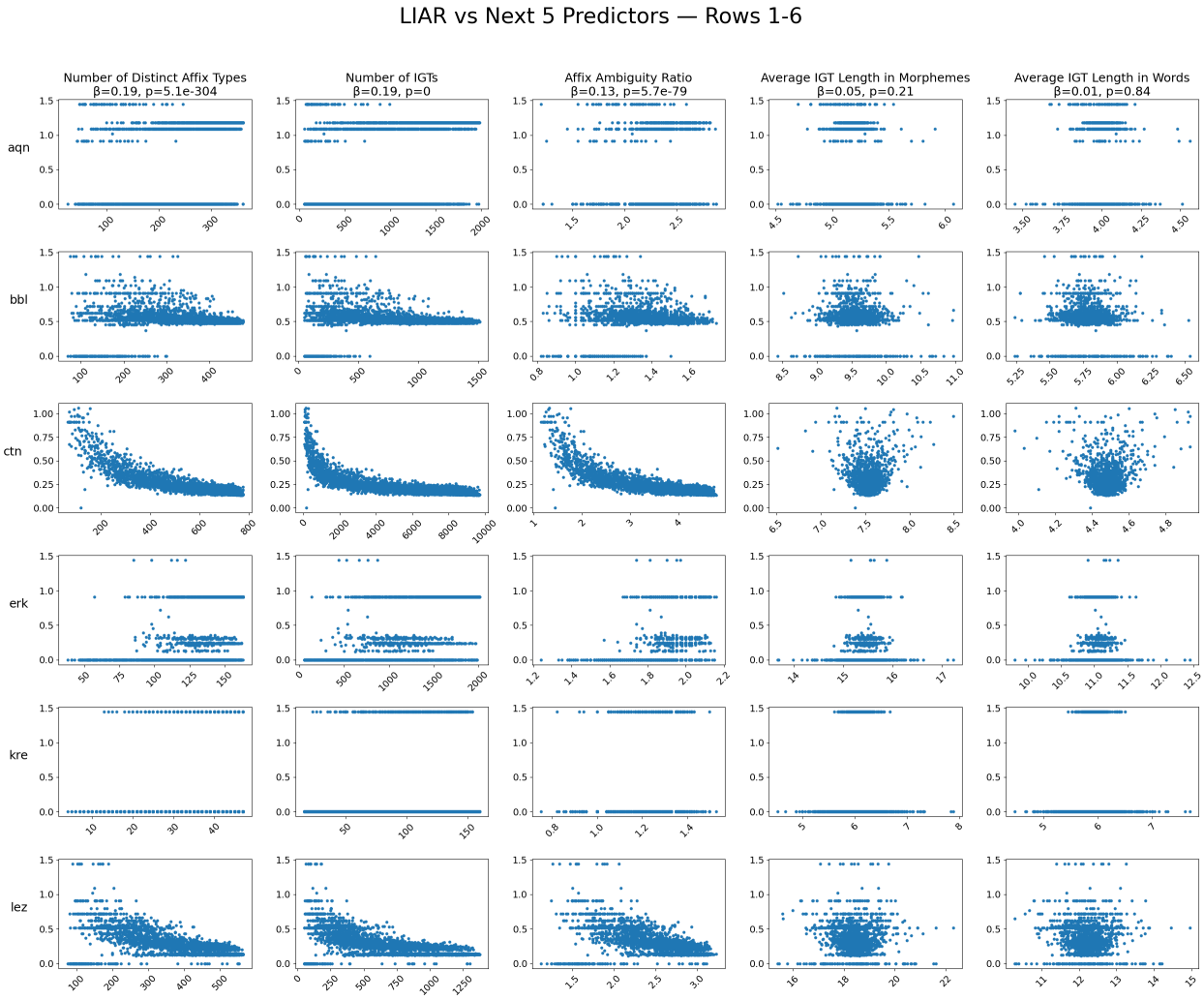


Figure A.3: Scatterplot grid for *LIAR* vs next 5 structural predictors (Rows 1-6).

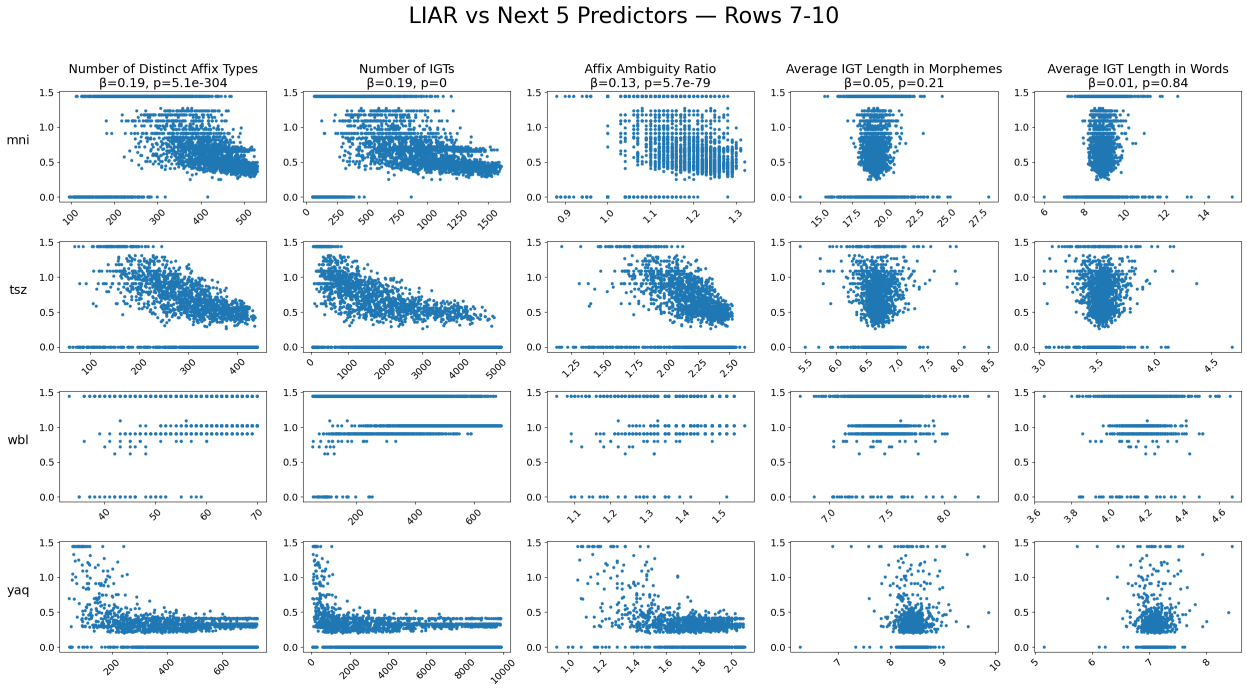


Figure A.4: Scatterplot grid for *LIAR* vs next 5 structural predictors (Rows 7-12).

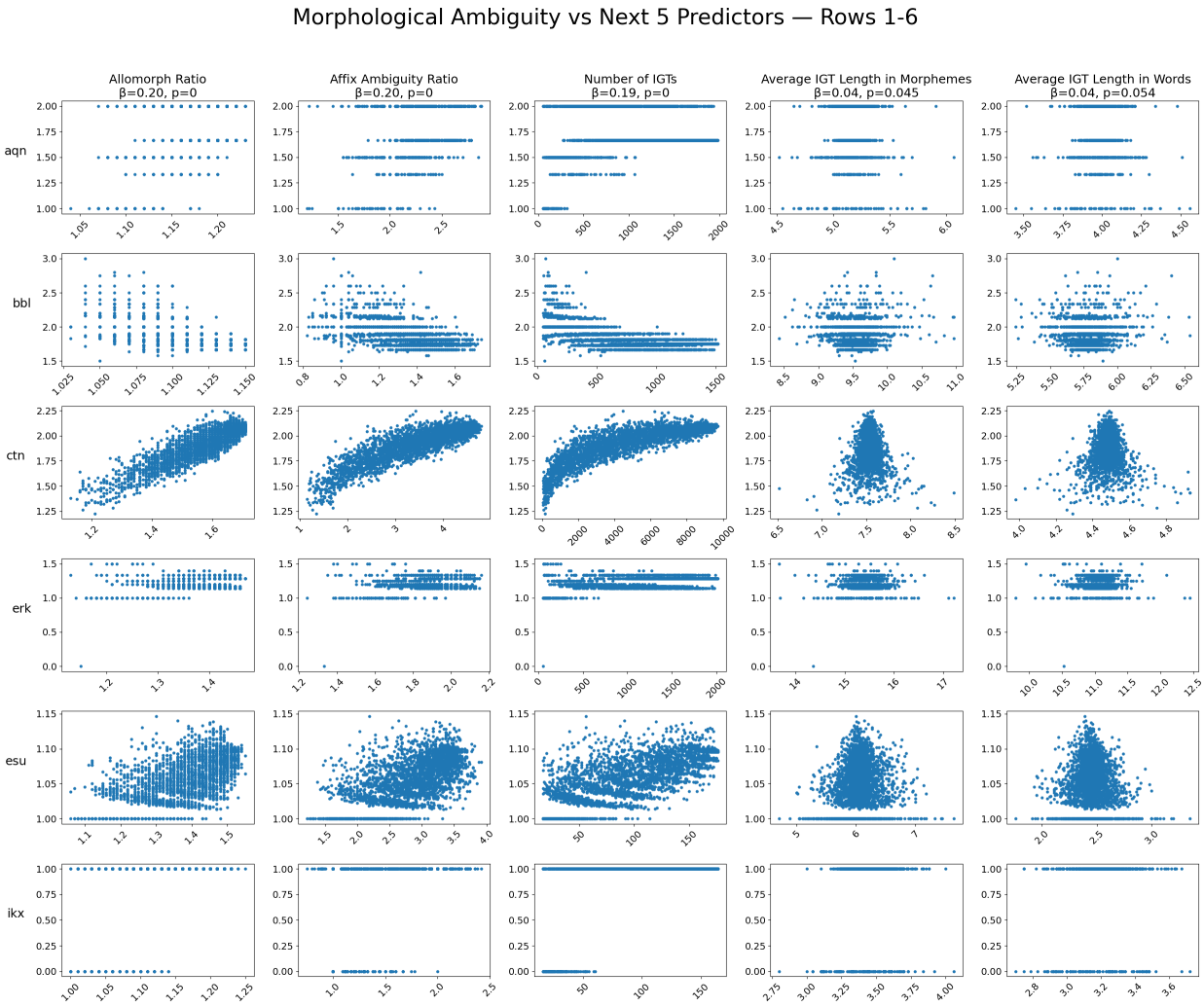


Figure A.5: Scatterplot grid for *Morphological Ambiguity* vs next 5 structural predictors (Rows 1-6).

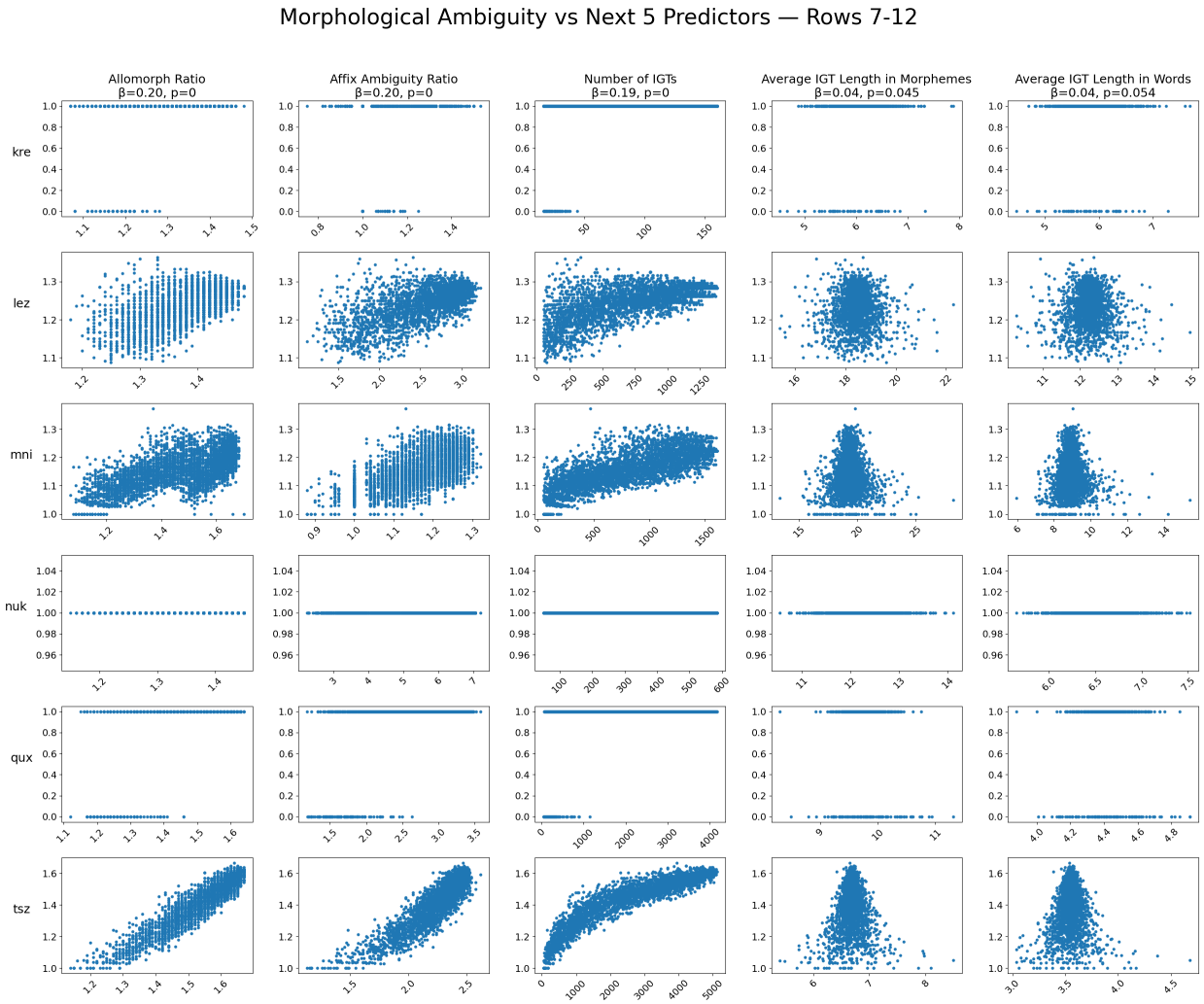


Figure A.6: Scatterplot grid for *Morphological Ambiguity* vs next 5 structural predictors (Rows 7-12).

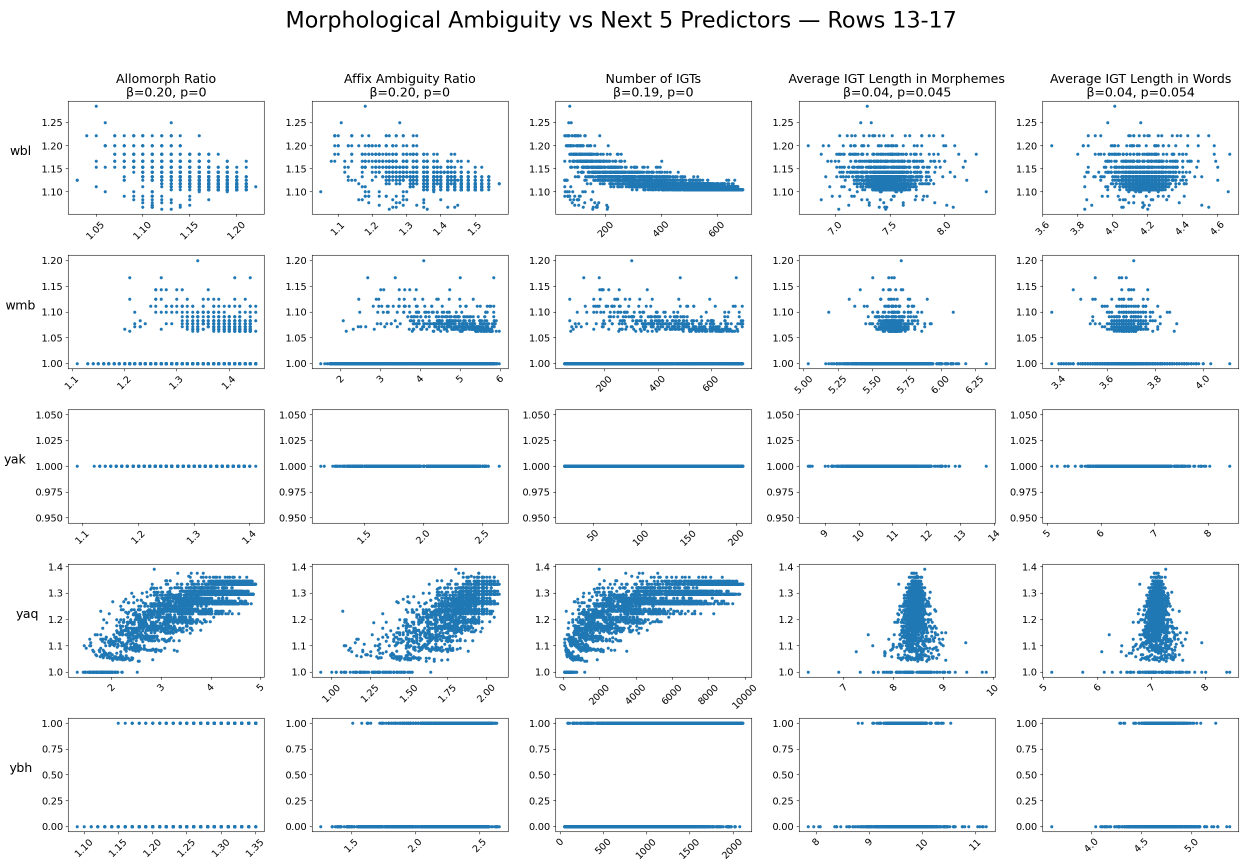


Figure A.7: Scatterplot grid for *Morphological Ambiguity* vs next 5 structural predictors (Rows 13-18).

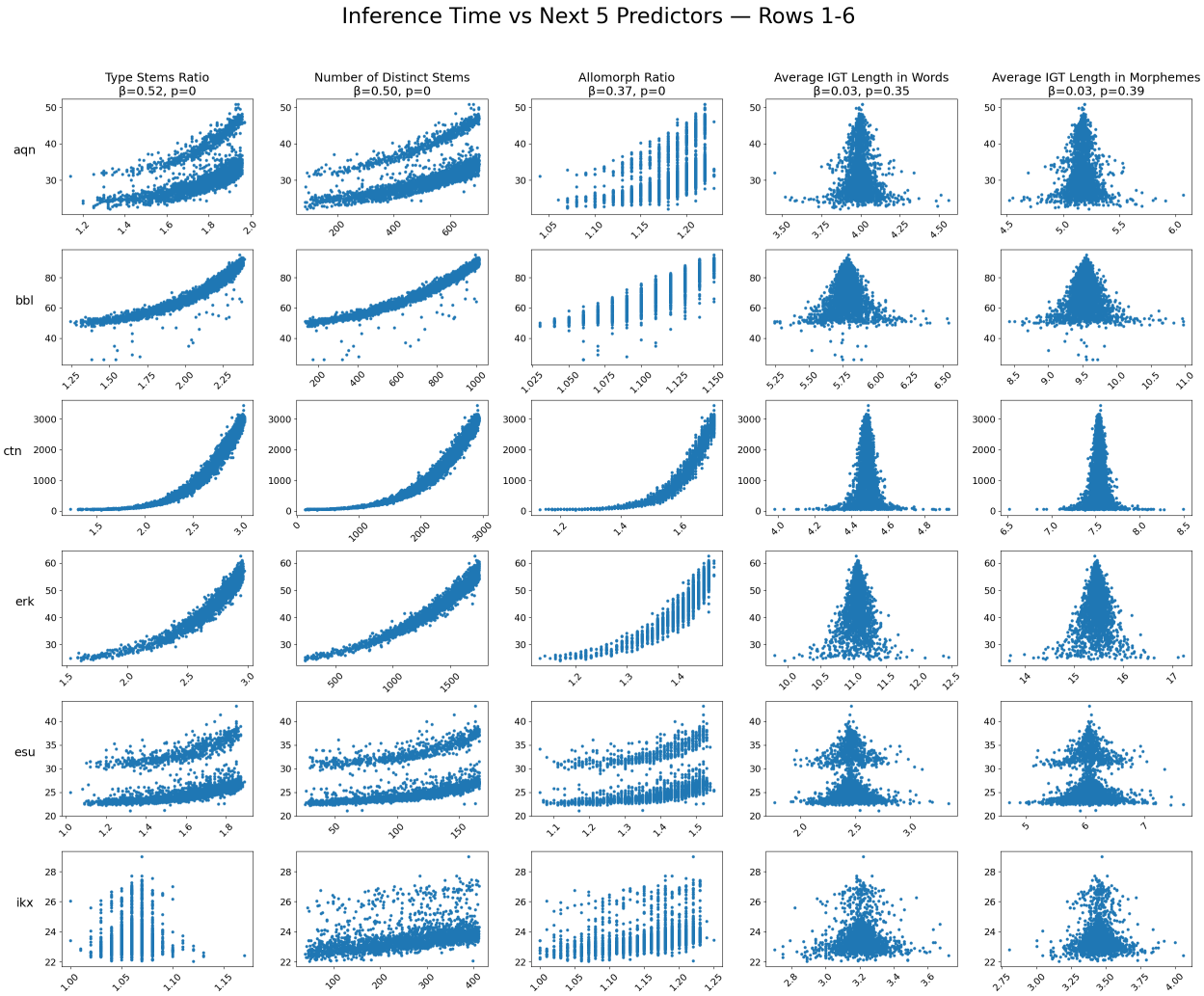


Figure A.8: Scatterplot grid for *Inference Time* vs next 5 structural predictors (Rows 1-6).

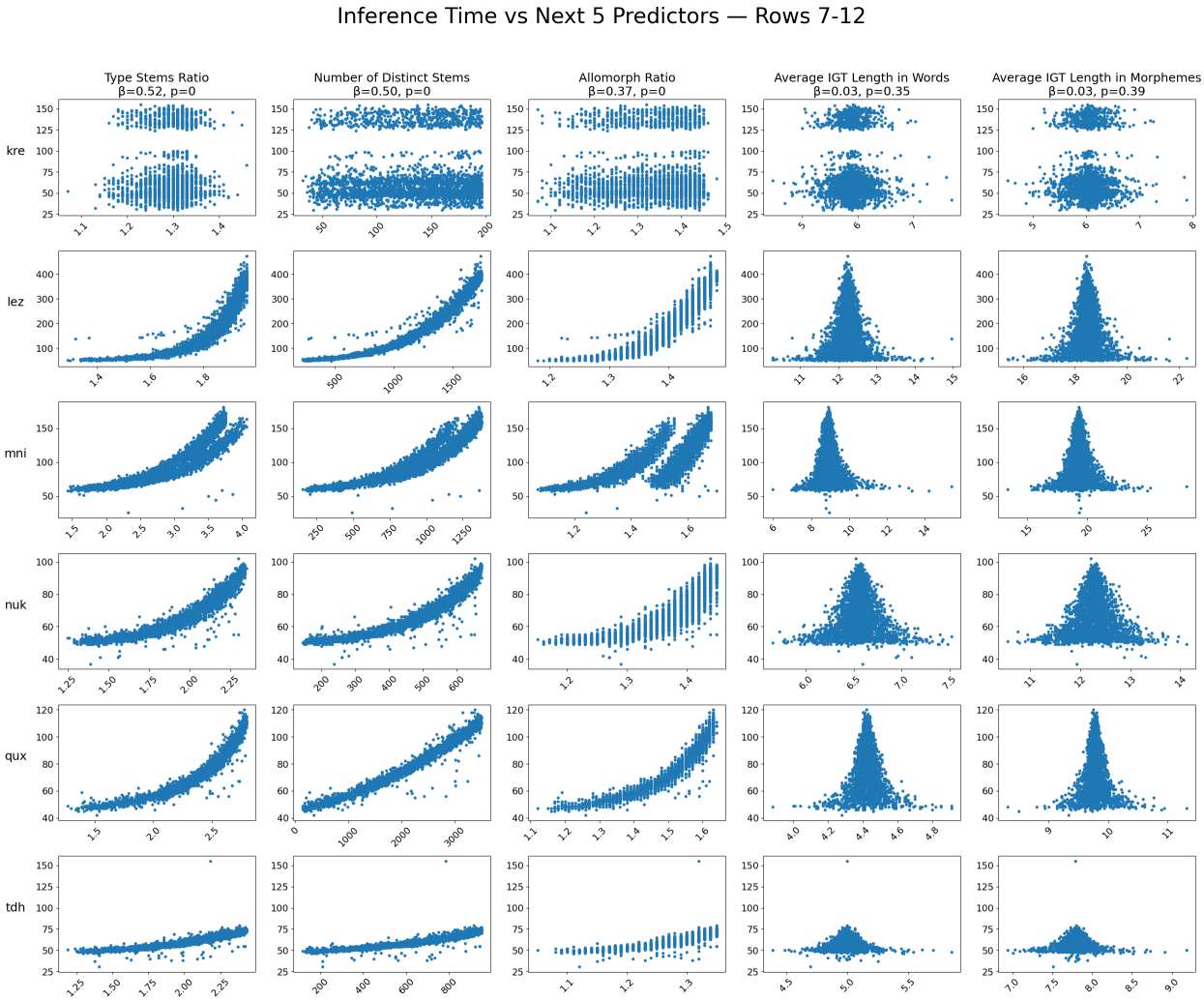


Figure A.9: Scatterplot grid for *Inference Time* vs next 5 structural predictors (Rows 7-12).

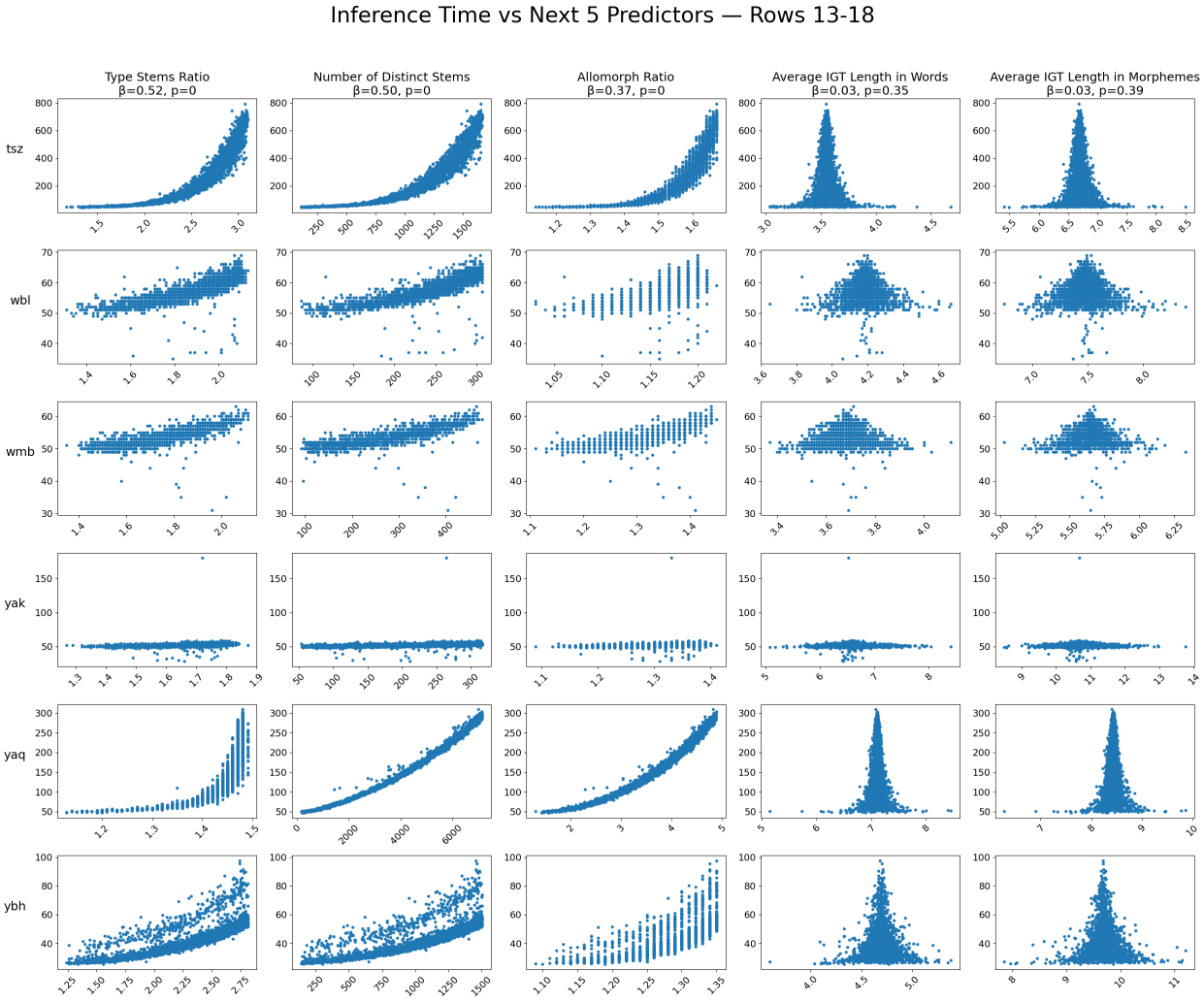


Figure A.10: Scatterplot grid for *Inference Time* vs next 5 structural predictors (Rows 13-18).