Automatically Inferring Grammar Specifications for Adnominal Possession from Interlinear Glossed Text

Allison Dods

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

Master of Science

University of Washington

2022

Reading Committee:
Emily M. Bender, Chair
Fei Xia

Program Authorized to Offer Degree:
Linguistics

University of Washington

Abstract

Automatically Inferring Grammar Specifications for Adnominal Possession from Interlinear Glossed Text

Allison Dods

Chair of the Supervisory Committee:
Emily M. Bender
Department of Linguistics

This thesis presents an update to the AGGREGATION grammar inference project: namely, the ability to automatically infer information about adnominal possession for a given language. Specifically, I contribute code that begins by examining interlinear glossed text (IGT) instances that have been enriched with information about word alignment, part-of-speech tags, and dependency structures; records information about how those IGT handle adnominal possession; consolidates this information; and finally prepares it to be added to the output grammar specification in accordance with the format expected by the adnominal possession library in the LinGO Grammar Matrix, which in turn can be used to create machine-readable grammars. In this thesis, I describe my contribution to the AGGREGATION project in detail, as well as my data-driven development and evaluation process using data from three development languages (Abui, Tsova-Tush, and Nafsan) and three held-out langauges (Matsigenka, Wambaya, and Hiaki). My program successfully infers several different approaches to adnominal possession. Specifically, it adds information to the grammar specification that describes the location of possession-marking and the presence of possessor pronoun affixes, as well as adding morphological information about the person information carried by possessive affixes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am so grateful to have had as an advisor Emily M. Bender, who has not only supported me through this thesis, but whose work has also guided my development as a linguist, from my first syntax class in college through my decision to join CLMS. Thank you, Emily, for the direction and feedback on this work; for all you've taught me about ethics over the years; and for your mentorship and encouragement of my growth.

I also want to thank several others whose support and contributions made this thesis possible: Fei Xia, my second reader, for her thoughtful feedback, particularly her insights on how to organize information efficiently and neatly; Elizabeth Conrad, for creating the beautifully-documented AGGREGATION setup and testing harness tools that saved me countless hours in getting started; Tom Liu, for his work on the adnominal possession library that paved the way for this thesis work to run more smoothly; Olga Zamaraeva, for welcoming me to the AGGREGATION neighborhood (and literal neighborhood) and so often being willing to offer a helping hand; Kristen Howell, for her work on BASIL that served as the foundation for this thesis and for the reference documents and support she offered from afar; and Elizabeth Nielsen, for creating the adnominal possession library and carving up the complicated theoretical space in elegant and digestible ways. Thanks to the current and recent members of the EMB-students and Matrix-AGG dev groups for the community, support, laughs, and food for thought, and to the alumni and wider DELPH-IN network for their guidance and feedback.

Finally, to my family, friends, and partner, who have kept me afloat throughout the experience of writing a master's thesis with a full-time job in a pandemic: thank you!

DEDICATION

To Jenna, Helen, and Alec, who made Seattle home

Chapter 1
INTRODUCTION

The AGGREGATION project seeks to automatically infer information from interlinear glossed text (IGT) about how that language handles various syntactic and morphological phenomena (Howell, 2020). In conjunction with the LinGO Grammar Matrix (Bender et al., 2002, 2010), AGGREGATION can be used to bootstrap the creation of machine-readable grammars, reducing the work a linguist must do by hand. While AGGREGATION is able to infer information about various lexical items, morphological rules, and syntactic phenomena, the system (as developed through Conrad 2021) does not infer information about how a language handles adnominal possession. My thesis aims to expand the AGGREGATION inference system to include this phenomenon.

I contribute a new inference module for the AGGREGATION codebase.[1] This module extends the existing output of the AGGREGATION grammar inference system (namely, a grammar specification to be used with the Grammar Matrix) to include information about how a given language handles adnominal possession. The system as designed and described in this thesis can infer various possessive strategies that a language might employ when the possessor is overt, as well as possessor pronouns that appear as affixes on the possessum. I show how this inference module produces at least one appropriate possessive strategy or possessor pronoun class for five of the six languages I examine in this thesis; while some aspects of adnominal possession remain uninferred, basic aspects of each of these five lan-

[1]The code as it exists at the time of this writing can be found (and used to reproduce these results) at https://git.ling.washington.edu/agg/aggregation/-/tags/aed-thesis.

guages' approach to possession are captured in the grammar specification, where they were not before. I also evaluate the parse coverage, ambiguity, and validated coverage of the inferred grammars on test sentences; while parse coverage and ambiguity do not improve, the inferred grammar for one language does show new validated coverage on two sentences with adnominal possession.

In the chapters that follow, I describe my thesis work. Chapter 2 provides background information about the AGGREGATION project and related tools as well as a description of the adnominal possession library in the Grammar Matrix, which serves as a guide for the design of my project. Chapter 3 walks through the algorithm I created for inferring adnominal possession and explains the data-driven evaluation process I use, and Chapter 4 presents and discusses the results of my project. Finally, Chapter 5 summarizes the results and opportunities for future work.

Chapter 2
BACKGROUND

This thesis extends the AGGREGATION system to infer information about adnominal possession. Section 2.1 of this chapter explains more about what AGGREGATION is, as well as what it means to "infer information" in this context, and Section 2.2 gives a brief description of adnominal possession as it relates to this thesis work and an overview of the way adnominal possession is handled in the Grammar Matrix.

2.1 AGGREGATION

AGGREGATION[1] is an ongoing project, with roots dating back to at least 2012 (Bender et al., 2012, 2013, 2014), that performs a type of automatic grammar generation called *grammar inference*. Howell (2020) defines automatic grammar generation as "the development of systems that automatically create grammars on the basis of data" (p. 6), and grammar inference as one type of approach to automatic grammar generation that is "based on text annotated with partial grammatical information but not full parse trees or logical forms" (p. 6). While there are a variety of ways these systems can be constructed, AGGREGATION takes as input interlinear glossed text (IGT) data from a given language that has been enriched by the INTENT package[2] (Georgi, 2016) and produces a grammar specification, which is a file containing certain lexical, morphological, and syntactic properties of that language.

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

[2] INTENT uses a method called projection to provide information that is not explicitly present in many IGT instances: word-to-word alignment between the language line and translation line, part-of-speech tags, and dependency structures. I give more detail about IGT, INTENT, and projection later in this chapter.

The grammar specification can then be used as input for the LinGO Grammar Matrix, a resource that facilitates the production of machine-readable grammars (Bender et al., 2002, 2010).

At a high level, the pipeline used in the AGGREGATION project consists of the following phases, which are defined and described in subsequent sections of this chapter:

1. Data preprocessing: Converting the input IGT data into an enriched corpus (Section 2.1.1)

2. Grammar inference: Inferring lexical and grammatical information about that language (Section 2.1.2)

3. Output: Producing a grammar specification that can be used with the Grammar Matrix and other tools to create machine-readable grammars and use them to parse sentences (Section 2.1.3)

A more detailed diagram of the AGGREGATION pipeline (from Howell, 2020, p. 16) is shown in Figure 2.1. In this diagram, the box with a bold outline (containing the text "BASIL + MOM") corresponds to the grammar inference phase; the boxes preceding it comprise the preprocessing phase, while the boxes following it cover the production and uses of the output grammar specification.

This thesis contributes improvements to the grammar inference portion of the AGGREGATION pipeline. Section 2.1.4 describes the methods typically used to evaluate changes to that part of the system.

2.1.1 Data preprocessing

The beginning of this process starts with a collection of interlinear glossed text (IGT) instances from some language. For the sake of illustration, consider an IGT instance from

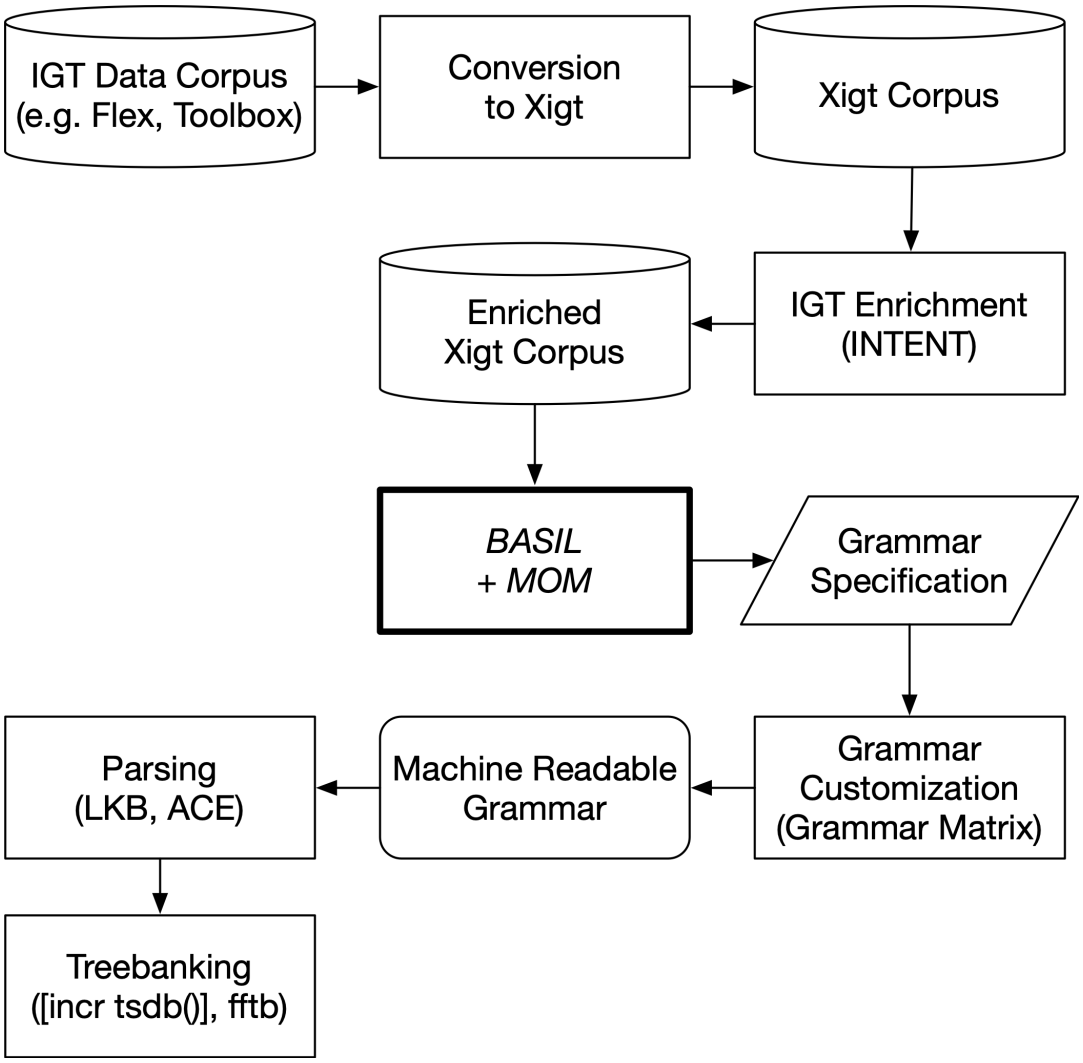

Figure 2.1: AGGREGATION pipeline, from Howell, 2020, p. 16

Abui (abz;[3] Kratochvíl, 2019) in (1).

(1) Di demayool hefalang taa.
Di de-mayool he-fala-ng taa.
3.AGT 3I.AL-wife 3.AL-house-towards sleep.IPFV
"He sleeps in his woman's house." (Kratochvíl, 2019)

A typical IGT instance, like the example above, consists of the following lines:[4]

- Language line: the original sentence
- Segmented line: the original sentence, segmented by individual morphemes
- Gloss line: a morpheme-by-morpheme gloss of the sentence
- Translation line: a translation of the original sentence into another language (which can be any language, but AGGREGATION currently only handles IGT with translation lines in English)

The AGGREGATION inference code expects each collection of IGT to be in a specific XML-based storage format, called Xigt (eXtensible Interlinear Glossed Text), which was developed to support the needs of the AGGREGATION project (Goodman et al., 2015). An instance of Xigt can contain more information than is explicit in a four-line IGT instance like the one above, such as metadata or annotations. Of particular interest for the automatic inference that AGGREGATION performs is Xigt's ability to store alignment information. An alignment is a relationship between two components of an IGT instance; some alignments that are implicit in IGT, such as between the segmented line and the gloss line, are made

[3]In this thesis, all languages are identified by their ISO 639-3 codes, as provided in https://iso639-3.sil.org/code_tables/639/data

[4]Some instances of IGT do not follow this pattern. It's possible, for example, for an IGT instance not to have a phrase line, not to have a segmented line, for the segmented line not to properly reflect the actual segmentation of morphemes or align with the gloss line, etc.

explicit when imported into Xigt. If the Xigt object is further enriched with INTENT (Georgi, 2016), as is the case in the AGGREGATION pipeline, it can store the alignments and other information that INTENT contributes. The Xigt codebase contains tools to convert IGT collections from other formats, like FLeX (Rogers, 2010) or Toolbox (SIL International, 2015), into Xigt and automatically add explicit alignment information.

Example (2) shows an excerpt of the IGT from (1) in the Xigt format. The *tiers* make explicit various aspects of the original IGT, such as the alignment between morphemes and glosses in the **g** tier or between glosses and words in the **gw** tier, and allows one to access individual elements via unique identifiers, such as using "m2" to refer to the morpheme glossed as "3I.AL."

(2)
```
<tier id="p">
  <item id="p1">Di demayool hefalang taa.</item>
</tier>
<tier id="w" segmentation="p">
  <item id="w1" alignment="gw1" segmentation="p1[0:2]">Di</item>
  <item id="w2" alignment="gw2" segmentation="p1[3:11]">demayool</item>
  <item id="w3" alignment="gw3" segmentation="p1[12:20]">hefalang</item>
  <item id="w4" alignment="gw4" segmentation="p1[21:25]">taa.</item>
</tier>
<tier id="m" type="morphemes" segmentation="w">
  <item id="m1" segmentation="w1">di</item>
  <item id="m2" segmentation="w2">de-</item>
  <item id="m3" segmentation="w2">mayool</item>
  <item id="m4" segmentation="w3">he-</item>
  <item id="m5" segmentation="w3">fala</item>
  <item id="m6" segmentation="w3">-ng</item>
  <item id="m7" segmentation="w4">taa</item>
```

```
</tier>
<tier id="g" alignment="m" segmentation="gw">
  <item id="g1" alignment="m1" segmentation="gw1">3.agt</item>
  <item id="g2" alignment="m2" segmentation="gw2">3i.al-</item>
  <item id="g3" alignment="m3" segmentation="gw2">wife</item>
  <item id="g4" alignment="m4" segmentation="gw3">3.al-</item>
  <item id="g5" alignment="m5" segmentation="gw3">house</item>
  <item id="g6" alignment="m6" segmentation="gw3">-towards</item>
  <item id="g7" alignment="m7" segmentation="gw4">sleep.Ipfv</item>
</tier>
<tier id="gw" alignment="w">
  <item id="gw1" alignment="w1">3.agt</item>
  <item id="gw2" alignment="w2">3i.al-wife</item>
  <item id="gw3" alignment="w3">3.al-house-towards</item>
  <item id="gw4" alignment="w4">sleep.Ipfv</item>
</tier>
<tier id="tw" segmentation="t">
  <item id="tw1">He</item>
  <item id="tw2">sleeps</item>
  <item id="tw3">in</item>
  <item id="tw4">his</item>
  <item id="tw5">woman</item>
  <item id="tw6">'s</item>
  <item id="tw7">house</item>
  <item id="tw8">.</item>
</tier>
```

The final preprocessing step in the AGGREGATION pipeline is to enrich the Xigt corpus

before inference begins. This step is performed by INTENT (Georgi, 2016), which can add three broad categories of information to each Xigt object: word alignment, part-of-speech (POS) tags, and syntactic dependencies. Word alignment matches words in the language line to words in the translation line. Once word alignment has taken place, INTENT performs *projection* — using information about the translation line to posit information about the corresponding words in the language line — to provide POS tags and dependency structures for words in the language line. These word alignments and projections are sometimes incomplete, missing, or incorrect, which can cause noise in the data used for grammar inference.

Example (3) shows some of the information that INTENT can add to enrich the IGT shown in (1) and (2), represented as tiers in the same Xigt object. This is not all of the information that INTENT is able to add, but it is the information I rely on most in this thesis work. The **bilingual-alignments_b** tier provides word alignments between the translation line and language line, and the **tw-ds** tier provides dependency relationships between words in the translation line. There is sometimes also a dependency structure tier that shows dependency relationships between words in the language line, but it is often missing or incomplete, so for the purposes of this thesis I rely on the translation dependency structures.[5]

---

[5]One risk that is introduced by my using the translation dependency tier is that when the algorithm considers IGT that have a possessive relationship (described in more detail in Chapter 3), it does so by examining only the translation, which does not always indicate that the original (source language) sentence demonstrates possession. Consider, for example, (i), which demonstrates possession in the English translation but not in the Italian sentence.

(i) Mi lavo i denti.
REFL.1SG wash.1SG the.M.PL teeth.M.PL
"I brush my teeth." (ita)

Currently, this risk does not appear to result in incorrect adnominal possession inference; the possessor in the translation line is not able to align with anything in the language line, so it does not get recognized as a possessive strategy, and unless there is a person-glossed affix on the possessum, no possessive pronoun is inferred. However, future work that expands the system should take care to block examples like these from incorrectly affecting inference. Alternatively, improving INTENT such that the dependency structures in the source language are more robust could also help shift reliance away from the translation dependency structures.

(3)
```
<tier id="bilingual-alignments_b" type="bilingual-alignments"
data-creation-time="2019-03-26 11:38:08" data-provenance=
"INTENT2-2.0a6" source="tw" target="w">
  <item id="tw_w_aln_1" source="tw2" target="w4" />
  <item id="tw_w_aln_2" source="tw7" target="w3" />
</tier>
<tier id="tw-ds" type="dependencies" data-creation-time=
"2019-03-26 11:38:08" data-method="spacy" data-provenance=
"INTENT2-2.0a6" dep="tw" head="tw">
  <item id="tw_ds_dep1" dep="tw1" head="tw2">nsubj</item>
  <item id="tw_ds_dep2" dep="tw2">root</item>
  <item id="tw_ds_dep3" dep="tw3" head="tw2">prep</item>
  <item id="tw_ds_dep4" dep="tw4" head="tw5">poss</item>
  <item id="tw_ds_dep5" dep="tw5" head="tw7">poss</item>
  <item id="tw_ds_dep6" dep="tw6" head="tw5">case</item>
  <item id="tw_ds_dep7" dep="tw7" head="tw3">pobj</item>
  <item id="tw_ds_dep8" dep="tw8" head="tw2">punct</item>
  <item id="tw_ds_dep9" dep="tw9" head="tw10">compound</item>
  <item id="tw_ds_dep10" dep="tw10">root</item>
</tier>
```

2.1.2 Grammar inference

Once a Xigt file has been enriched with INTENT, the core work of inference in AGGREGATION can begin. In Figure 2.1, the part of AGGREGATION that I refer to as "core inference" is designated by the box with the bold outline. It's this core inference section — and in particular BASIL — that my thesis contributes to.

BASIL (Howell, 2020) and MOM (Wax, 2014; Zamaraeva, 2016; Zamaraeva et al., 2017)

are systems that work together to perform grammar inference in AGGREGATION. Briefly, and generally, MOM infers nominal and verbal lexical items, morphotactics, and some of the syntactic and semantic constraints governing lexical rules, while BASIL infers other lexical classes, some morphological information, and various syntactic properties.

In more detail, the inference system worked (prior to my contributions) as follows:

1. The Xigt corpus is ingested, along with a file containing language-specific settings.

2. BASIL infers information about the language's case system and verb transitivity, and passes this information to MOM.

3. MOM infers nouns and verbs (as lexical items) and some affixes (which are grouped into position classes).

4. BASIL separates out pronouns from nouns and auxiliaries from main verbs; infers more lexical items (specifically determiners, adpositions, negation adverbs, and coordinators); infers some more morphological information (specifically person/number/gender and tense/aspect/mood); and infers some syntactic properties (specifically word order, argument optionality, sentential negation, and coordination).

5. The inferred information from MOM and BASIL is stored and printed in the form of a grammar specification.

2.1.3 Inference output

The grammar specification (also called a "choices file") can be used as input to the customization system of the LinGO Grammar Matrix (Bender et al., 2002, 2010). Once the inferred grammar specification is entered as input to the customization system, the Grammar Matrix can produce a grammar for that language, provided that the grammar specification has

enough detail and is internally consistent. Recent precedent in the AGGREGATION literature (e.g. Howell, 2020; Conrad, 2021) is to follow this fully-automated process in order to evaluate the performance of the inference system. However, at this point a linguist can also manually add or edit information either in the grammar specification itself or in the customization system after having uploaded the grammar specification. In this way, the AGGREGATION inference system can bootstrap a linguist's ability to use the Grammar Matrix to create a grammar without having to start from scratch.

The resulting grammar can be used along with the LKB (Copestake, 2002) and ACE (Crysmann and Packard, 2012) to parse sentences. Parsing can be an end in itself — that is, a linguist can use these tools to analyze a given sentence by, for instance, seeing whether it parses using that grammar, or reviewing its syntactic and semantic composition in HPSG (Pollard and Sag, 1994) and MRS (Copestake et al., 2005). A linguist can also use [incr tsdb()] (Oepen and Flickinger, 1998) or FFTB (Packard, 2015) to treebank the parsed sentences, helping to identify correct parses.

2.1.4 Evaluation

Much previous AGGREGATION work (Bender et al., 2014; Zamaraeva et al., 2019; Howell, 2020; Conrad, 2021) has evaluated changes to the inference system by parsing (and, often, treebanking) sets of test sentences and calculating various metrics, using cross-validation; specifically, recent precedent is that the evaluation is run ten times for each language, setting aside a different fold of that language's data to be used as test data each time, while the other nine folds are used for training. Howell (2020) and Conrad (2021) perform this evaluation on both a set of development languages and a set of held-out languages. An AGGREGATION experimenter uses the development data to drive decisions when building the system; for example, they might run the system on the development languages, analyze the results, and then adjust the code based on those results. (The development language data guides the

design of the system, but the system does not involve hyperparameters.) Held-out data, including descriptions of the languages the data is from, on the other hand, is not viewed until the code has been "frozen". In other words, the experimenter builds the system without reference to the held-out languages, and then, only after finishing writing code, runs the system on the held-out data. No further changes are made to the system at this point.

Metrics frequently used in previous AGGREGATION inference evaluations and that are relevant to the current work on syntactic inference include *parse coverage* (how many sentences parse using the inferred grammar), *ambiguity* (of the sentences that parse, how many readings the grammar produces on average), and *validated coverage* (how many of the parsed sentences include a semantically correct parse, according to manual evaluation using treebanking). It can also be helpful to examine the *accuracy of the inferred grammar specification* to evaluate the performance of the inference system independently of the ability of the resulting grammar to parse sentences. I explain my approach to evaluating these metrics in more detail in Chapter 3.

2.1.5 Summary

In this section, I have provided an overview of the AGGREGATION pipeline and related systems, starting from the input IGT and their enhancement via INTENT, continuing through the inference contributed by BASIL and MOM to produce a grammar specification, and ending with possible uses for and evaluation of the grammar specification. This pipeline forms the foundation for the adnominal possession inference code I am adding.

2.2 Adnominal possession in the Grammar Matrix

While my goal, broadly stated, is to expand AGGREGATION to include inference for adnominal possession, what that specifically means is filling out the appropriate adnominal-possession-related values in a grammar specification for a given language. As such, my

approach to inferring adnominal possession is primarily guided by the way the adnominal possession library in the Grammar Matrix has been implemented (Nielsen, 2018).

In the context of the Grammar Matrix, *adnominal possession* denotes the phenomenon in which a noun phrase refers to an entity (the *possessor*) that possesses another entity (the *possessum*), such that "[t]he possessor is a syntactic dependent of the possessum" (Nielsen, 2018, p. 3). Nielsen's review of the typological literature describes various syntactic approaches to adnominal possession in the world's languages, many of which are able to be modeled by the Grammar Matrix. According to this review, many languages express adnominal possession by marking the possessor or possessum (or both) with an affix or word that indicates possession. This marker may be used only to express possession, or it may share a function with some other phenomenon (for example, some languages use the genitive case on the possessor to indicate possession). Additionally, some languages indicate possession by juxtaposing the possessor and possessum in a fixed order with no marking (Nielsen, 2018).

The Grammar Matrix provides support for two broad ways of characterizing adnominal possession in a language: *classes of possessor pronouns* and *possessive strategies*. Possessor pronouns are pronouns that are only used within a language for adnominal possession; English (eng) is an example of a language that has a class of possessor pronouns (e.g. *my*, *your*, *her*). Possessor pronouns can appear as affixes on the possessum, or as separate words (or clitics).[6] A possessive strategy, on the other hand, reflects how a language expresses adnominal possession when the possessor is both overt and does not belong to a specific class of possessor pronouns. In English, constructions such as *Nadia's book* must be modeled in the Grammar Matrix using a possessive strategy.

---

[6]In the Grammar Matrix, "clitic" refers to a word that is syntactically independent but phonologically dependent on some other word. However, because the AGGREGATION project has noticed many field linguists using the = delimiter with affixes with less phonological integration (Emily M. Bender, personal communication), AGGREGATION (including my code) treats morphemes marked in IGT with "=" as affixes. In this thesis, I group clitics with words when discussing the Grammar Matrix and with affixes when discussing AGGREGATION.

When creating a grammar specification, a user-linguist can add zero or more possessive strategies with the following information:

- the relative order of the possessor and possessum (head-initial, head-final, or either)
- whether the possessor behaves like a specifier or a modifier (see Nielsen 2018 for detail about distinguishing between these options; the suggested heuristic is whether the possessum can take both a determiner and a possessor simultaneously)
- which of the constituents (possessor, possessum, both, or neither) is marked for possession, and how the marking is expressed (via an affix, or via a separate word/clitic)
- whether the possession-marking word or affix agrees with the constituent it does not mark (e.g. whether a possessor-marking affix agrees with the possessum)
- in some constructions, whether the possessor is required to have certain features (e.g. be in a certain case)
- in some constructions, whether or not the possessor can be a pronoun

Additionally, the user-linguist can add zero or more classes of possessive pronouns, specifying whether these pronouns:

- appear as affixes or as separate words/clitics
- agree with the possessum
- behave like a specifier or a modifier
- appear before or after the possessum (only in the case of pronouns that are separate words)

In the grammar specification, information about the adnominal possessive strategies and pronoun classes is encoded in the `adnom-poss` section. If the strategy or pronoun involves affixes, further information about those affixes — including the features they carry both inherently and in agreement with other words — is entered in the `morphology` section.

2.3 Summary

In this section, I have provided context for my thesis project inferring adnominal possession by giving an overview of the AGGREGATION project, within which my work is situated, as well as the implementation of adnominal possession in the LinGO Grammar Matrix. The next chapter shows how I build on the existing work to expand the inference system within AGGREGATION.

Chapter 3
METHODOLOGY

In this chapter, I first give an overview of the data I used to develop and test the system. I then describe the algorithm I have designed and implemented to infer adnominal possession information. Finally, I explain the process I used to evaluate these changes.

3.1 Language data

In keeping with the AGGREGATION project's tradition of data-driven development (e.g. Howell, 2020; Conrad, 2021), I tested my changes on two sets of language data: first, data from three *development languages*, which I referred to directly throughout the programming and evaluation phases, and second, data from three *held-out languages*, which I did not view until after I was finished programming. In this way, I could guide and incrementally refine my design and programming choices based on the development languages' approaches to adnominal possession and then evaluate the system on the set of held-out languages to understand how well my system might generalize.

The development languages I consulted for this project were Abui (abz; Papuan; Kratochvíl, 2019),[1] Tsova-Tush (bbl; Northeast Caucasian; Hauk, 2016-2019), and Nafsan (South Efate) (erk; Oceanic; Thieberger, 2006b). I chose to work with the data from Abui based on previous familiarity with that language, and randomly selected the others from a list of data to which the AGGREGATION project has access, making sure as I did so that no two development languages came from the same language family. I then chose three held-out

[1]For each language in this paragraph, the information in parentheses refers to the ISO 639-3 code for the language, its language family, and the source of the data I used, respectively.

languages at random, discarding any languages from the same language family as one of the development languages; the held-out languages I used were Matsigenka (mcb; Arawak; Michael et al., 2013), Wambaya (wmb; non-Pama-Nyungan; Nordlinger, 1998), and Hiaki (yaq; Uto-Aztecan; Harley, 2019).

Table 3.1 gives a sense for how much training data my module was able to work with. Specifically, it shows the mean number of sentences across training folds,[2] as well as the mean number of instances in each training fold that indicate possession. To arrive at the number of instances of possession in a training fold, I count the number of relationships labeled "poss" in the translation dependency tier populated by INTENT (Georgi, 2016). This number does not describe how many *sentences* have possession, as some sentences may have several instances of possession; it also is more accurately described as the number of instances of possession in the *translations* of sentences in the training data. (See Footnote 5 in Chapter 2 for more context.)

Language	Sentences per training fold		Instances of possession in training data	
	mean	std. deviation	mean	std. deviation
Abui (abz)	1441.2	0.6	241.7	5.9
Tsova-Tush (bbl)	1516.8	2.4	231.9	4.0
Nafsan (erk)	1687.5	1.5	464.4	6.7
Matsigenka (mcb)	469.1	0.3	282.2	2.7
Wambaya (wmb)	782.2	0.6	144.5	4.0
Hiaki (yaq)	2519.5	1.5	172.0	4.2

Table 3.1: Training dataset size and number of instances of possession: mean and standard deviation

[2]See Section 2.1.4 in Chapter 2 for more context on the ten-fold cross-validation process used in the AGGREGATION project.

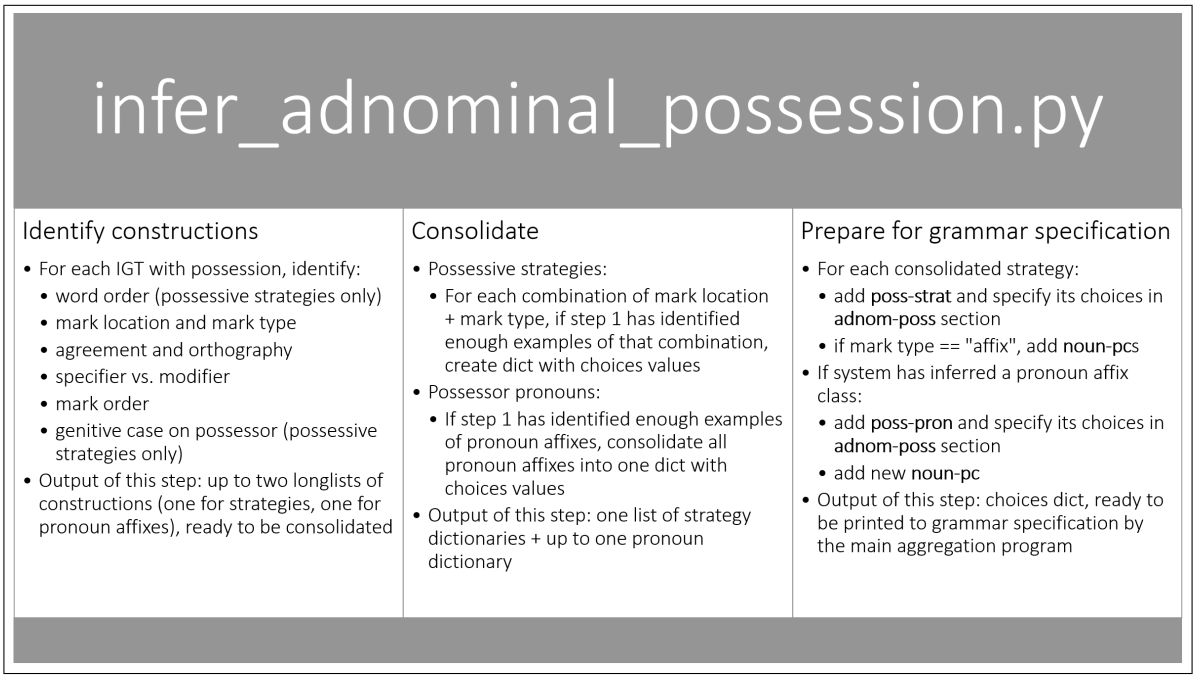

Figure 3.1: Summary of the three main steps in the adnominal possession inference module

3.2 Implementation

The algorithm for inferring adnominal possession consists of three broad steps: first, it loops through likely examples of adnominal possession in the dataset and **identifies** constructions marking possession (potential possessive strategies as well as potential possessor pronoun affix classes); second, it **consolidates** the strategies and pronoun classes to arrive at a de-duplicated and higher-confidence list of each; and finally, it **prepares** this information to be written to the grammar specification by the main aggregation program. These steps, and their substeps, are visualized in Figure 3.1.

The `infer_adnominal_possession` module is passed a Xigt corpus of IGT in the language and the grammar specification (referred to in the code as the "choices file") as constructed up to that point, and is called from the main AGGREGATION program after all other

inference modules have run. `infer_adnominal_possession` is positioned after the inference of features (such as person and case) so it can refer to the list of possible (inferred) features for the language when inferring the agreement behavior of possession markers.

To illustrate, I refer to example (4), from Nafsan (Thieberger, 2006b), throughout this section, commenting on how the system is designed to infer information based on this IGT instance. This Nafsan phrase demonstrates a standalone possession-marking word, *ni*, that does not agree with other entities.

(4) me nlaken too ki1pe lewi-ki kom ni taap̃es
and because chicken 3S.PS1-PF admire-TR comb.BI of swamphen
"But because the children liked the swamphen's comb," (erk, Thieberger (2006b))

3.2.1 Identifying constructions

The first phase of the adnominal possession inference algorithm is to build a list of possessive constructions from the IGT. In the code, this phase is handled by the `identify_strats` function. This function loops through all of the IGT in the training data file and produces two "longlists": one for potential possessive strategies (`strat_longlist`) and one for potential possessive pronominal affixes (`poss_pron_affix_longlist`).[3] Each longlist is a list of dictionaries, and each dictionary corresponds to an instance of adnominal possession from the training data. (It is possible for a single IGT item to have more than one instance of adnominal possession, in which case each instance is represented in a different dictionary.)

While looping through all IGT, the `identify_strats` function only considers those IGT for which INTENT has provided a translation dependency tier, i.e. a tier that provides a label for the syntactic dependency relationship between the words in the translation line. If one of these relationships has the value "poss", the algorithm uses that dependency relationship to identify the possessor and possessum in the translation language, then checks to see whether

[3]While the adnominal possession library in the Grammar Matrix can handle both affix and non-affix possessive pronoun classes, I have currently only implemented inference for the affixes.

INTENT has provided bilingual alignments for both the possessor and the possessum. If both words have bilingual alignments, the algorithm begins building a possessive strategy dictionary to add to the `strat_longlist`; if only the possessum has a bilingual alignment, the algorithm checks whether the possessum appears to have a possession-marking affix, and if so, it begins building an entry in the `poss_pron_affix_longlist`.

The translation word tier (**tw**), dependency tier (**tw-ds**), and bilingual alignments tier (**bilingual-alignments_b**) provided by INTENT for the Nafsan example in (4) are shown in (5). The translation dependency tier indicates a possessive relationship between **tw7**, the possessor, and **tw9**, the possessum.[4] Both **tw7** and **tw9** are associated with Nafsan words via the bilingual alignments tier; therefore, the algorithm will treat this example as a possessive strategy.

(5)
```
<tier id="tw" type="words" segmentation="t">
  <item id="tw1">But</item>
  <item id="tw2">because</item>
  <item id="tw3">the</item>
  <item id="tw4">chicken</item>
  <item id="tw5">liked</item>
  <item id="tw6">the</item>
  <item id="tw7">swamphen</item>
  <item id="tw8">s</item>
  <item id="tw9">comb</item>
</tier>
<tier id="tw-ds" type="dependencies" data-creation-time=
"2020-03-09 16:56:52" data-method="spacy" data-provenance=
"INTENT2-2.0a6" dep="tw" head="tw">
```

[4]The possessum is analyzed as the head in the Grammar Matrix adnominal possession library as well as in my code.

```
  <item id="tw-ds-dep1" dep="tw1">root</item>
  <item id="tw-ds-dep2" dep="tw2" head="tw5">mark</item>
  <item id="tw-ds-dep3" dep="tw3" head="tw4">det</item>
  <item id="tw-ds-dep4" dep="tw4" head="tw5">nsubj</item>
  <item id="tw-ds-dep5" dep="tw5" head="tw1">advcl</item>
  <item id="tw-ds-dep6" dep="tw6" head="tw7">det</item>
  <item id="tw-ds-dep7" dep="tw7" head="tw9">poss</item>
  <item id="tw-ds-dep8" dep="tw8" head="tw7">case</item>
  <item id="tw-ds-dep9" dep="tw9" head="tw5">dobj</item>
</tier>
<tier id="bilingual-alignments_b" type="bilingual-alignments"
data-creation-time="2020-03-09 16:56:52" data-provenance=
"INTENT2-2.0a6" source="tw" target="w">
  <item id="tw_w_aln_1" source="tw2" target="w2" />
  <item id="tw_w_aln_2" source="tw4" target="w3" />
  <item id="tw_w_aln_3" source="tw7" target="w8" />
  <item id="tw_w_aln_4" source="tw9" target="w6" />
</tier>
```

In the following subsections, I list the functions involved in inferring the possessive strategy dictionaries, as that is currently the more complicated and robust portion of the algorithm. After I describe these functions, I describe the process for inferring possessive pronoun affixes, which relies on a modified subset of the functions used for strategy inference.

Word order

The function `infer_poss_order` takes the indices of two words[5] within the current sentence and returns a value to indicate their relative order. It returns "head-final" if the possessor precedes the possessum, and "head-initial" otherwise. Note that in some languages, the relative order of these words may not be fixed, but at this stage the algorithm only considers a single sentence in isolation. Later in the process, when the program is consolidating strategies, if there is evidence of multiple possible word orders for a strategy, that fact is then reflected in the final grammar specification.

Combining the information given by the translation dependency and bilingual alignments tiers for the Nafsan IGT in example (5) suggests (correctly) that the possessor is **w8** and the possessum is **w6**. `infer_poss_order` checks these identifiers and, since six is less than eight and the possessum thus precedes the possessor, returns "head-final."

Mark location and type

The function `infer_mark_loc_and_type` updates the dictionary with information about whether the possessor or possessum (or both, or neither) is marked for possession, and if so, whether the marking morpheme is an affix or a non-affix (separate word or clitic). It also records the identifier (either the morpheme or word ID from the Xigt object, depending on whether the possession marker is an affix or non-affix, respectively).

To determine whether a given word (possessor or possessum) contains a possession-marking affix, I pass the word and IGT to the function `has_possession_marking_affix`, which examines the gloss of the word for strings that I think are likely to indicate possession: "POSS", "of", "POS", "1", "2", or "3". I constructed this list based on the assumption that these glosses indicate morphemes that are *only* used for possession, rather than those that

[5]This function is called only for possessive strategies, i.e. when the previous code has identified a word that corresponds to each of the possessor and possessum.

have non-possession-related uses.[6] Numbers are included in this list because I assume nouns do not have reasons other than possession to be marked for person. Note that this list may not cover all possible glosses that indicate possession.

To determine whether a given word (possessor or possessum) is marked with a possession-marking word or clitic, the word, IGT, and word order information are passed to the function `has_possession_marking_clitic_or_word`. Briefly, what this function does is to examine the possible word orders of a single-word possessor, single-word possessum, and separate word (called "POSS" here as shorthand) to determine whether the current sentence matches any of these patterns:

1. POSS possessor possessum — possessor is marked by a prepositional word/clitic
2. possessor POSS possessum — possessor is marked by a postpositional word/clitic
3. possessor possessum POSS — possessum is marked by a postpositional word/clitic
4. POSS possessum possessor — possessum is marked by a prepositional word/clitic
5. possessum POSS possessor — possessor is marked by a prepositional word/clitic
6. possessum possessor POSS — possessor is marked by a postpositional word/clitic

I've chosen to analyze patterns in which the possession-marking-word (POSS) appears between the possessor and possessum as marking the possessor for simplicity and based on the assertion in Nielsen 2018 that possessor marking is the most common adnominal possession strategy typologically.

To illustrate using the running Nafsan example: the function would first check the possessor (**w8**) to see if it appears to have a possession-marking affix or word. No glosses on the

[6]The distinction between morphemes that are only used for possession and general (non-possession-specific) morphemes is important in the Grammar Matrix's adnominal possession library. Possessors that contain general features that may indicate possession — such as genitive case — are handled later in the process as part of the `infer_psr_feats` function.

possessor itself indicate possession, so it is determined not to have a possession-marking affix. It does, however, match one of the patterns that indicates a possession-marking separate word; namely, it matches the "possessum POSS possessor" pattern. The function determines this by examining the gloss-word tier shown in example (6), which indicates that the word preceding the possessor has a possession-related gloss (that is, "of"). It enters into the strategy dictionary the information that the possessor is marked by a separate word, and that **w7** is the possession marker. No marking is found on the possessum, so the overall mark location for this example is "possessor."

(6)
```
<tier id="gw" type="glosses" alignment="w">
  <item id="gw1" alignment="w1">and</item>
  <item id="gw2" alignment="w2">because</item>
  <item id="gw3" alignment="w3">chicken</item>
  <item id="gw4" alignment="w4">3S.PS1-PF</item>
  <item id="gw5" alignment="w5">admire-TR</item>
  <item id="gw6" alignment="w6">comb.BI</item>
  <item id="gw7" alignment="w7">of</item>
  <item id="gw8" alignment="w8">swamphen</item>
</tier>
```

Specifier-like versus modifier-like possessors

The Grammar Matrix distinguishes between possessors that behave like specifiers and those that behave like modifiers. While Nielsen (2018) lists three heuristic tests for determining which of these behaviors is present in a given language, I have implemented one test: whether or not the possessum takes both a determiner and a possessor simultaneously.[7] The function

[7]Future work may consider implementing other tests mentioned in Nielsen 2018: whether the possessor and a (different) modifier on the possessum can appear in variable order, and whether anything in the gloss or translation suggests definiteness.

`infer_spec_vs_mod` examines the word dependency tier in the Xigt object, if present, and checks whether dependencies both have the value "det" (the value that indicates a relationship between a word and its determiner) and have the possessum as the head. If so, the value of `mod_spec` for that example is "mod". If not, the default value of "spec" recommended by the adnominal possession library documentation is chosen.

Example (7) shows the word dependencies tier provided by INTENT for the running Nafsan example. None of the word dependencies have the value "det", so the default value of "spec" is chosen:

(7)
```
<tier id="w-ds" type="dependencies" data-creation-time=
"2020-03-09 16:56:52" data-method="project" data-provenance=
"INTENT2-2.0a6" dep="w" head="w">
  <item id="w-ds-dep1" dep="w1" head="w2" />
  <item id="w-ds-dep2" dep="w2">mark</item>
  <item id="w-ds-dep3" dep="w3">nsubj</item>
  <item id="w-ds-dep4" dep="w6">dobj</item>
  <item id="w-ds-dep5" dep="w8" head="w6">poss</item>
</tier>
```

Agreement and orthography

If the strategy indicates some form of possession-specific marking (i.e. the value of ["mark loc"] is not *neither*), the program calls `infer_agreement_and_orthography`. This function examines the possession-marking entity for signs of agreement and records this information and its spelling in the dictionary.[8] First, the function records the spelling of the possession-

[8]The agreement and orthography inference are technically independent at this stage, so they may make sense to separate into individual functions in future work. I chose to group agreement and orthography inference into the same function because the Grammar Matrix questionnaire triggers a question about the orthography of the possession marker just in case the user has indicated that it does not demonstrate agreement, and because there are other points in my algorithm (namely, in the consolidation and grammar

marking affix or word into the dictionary. It then checks whether the possession marker has a gloss indicating agreement for person,[9] and if so, records that information in the dictionary as well.

The Grammar Matrix questionnaire asks whether a possessor-marking morpheme agrees with the possessum, and whether a possessum-marking morpheme agrees with the possessor. However, I found it difficult to infer based only on glosses whether the two entities agree in either of these situations, at least with respect to person. For example, a possessum may have an affix whose gloss contains the gram "3", suggesting that the affix might be agreeing with the person value of the possessor. But the possessor may be some word that is assumed to be third-person (e.g. "teacher"), and as such may not itself be glossed with the gram "3". With this constraint in mind, I followed a similar approach to what I described in Section 3.2.1: I implemented the agreement function such that if the gloss of the possession-marking entity contains grams that commonly mark person information ("1", '2", "3"), the morpheme is assumed to agree with respect to person with the word that it does not mark (the target word). (That is, if a possessum-marking morpheme is glossed with a person-related gram, the target word that it's assumed to agree with is the possessor, and vice versa for possessor-marking morphemes, with the added constraint that possessors are not expected to show 1st or 2nd person agreement with possessums.) The function does not assume that any other grams indicate agreement.

In the running Nafsan example, (6) indicates that the possession marker **w7** is not glossed with any number, so the function records in the dictionary for this construction that it is non-agreeing. The orthography "ni" is also entered into the dictionary.

---

specification preparation steps) in which they are more tightly coupled.

[9]I chose the heuristic of considering just person information for simplicity, based on the assumption that a noun that is marked for person is likely marked as such for agreement reasons. Expanding the list of grams to include other features that might indicate agreement, such as gender or number, is left as future work.

Mark order

Inference of mark order (that is, the order of the possession marker relative to the word it marks) is handled differently based on whether the marker is a separate word or an affix. For markers that are separate words, the algorithm simply considers the indices of the marker and the word it marks. In keeping with the convention used by the adnominal possession library of the Grammar Matrix, constructions in which the marker precedes the marked word are entered in the dictionary as "head-initial", and those in which it follows the marked word as "head-final". Affixes are determined to be either prefixes or suffixes by examining the orthography of the affix; if it ends in a hyphen, it is assumed to be a prefix, and otherwise assumed to be a suffix, and entered into the dictionary as such.

Example (6) indicates that the possession marker, **w7**, precedes the possessor (which is the word it marks), **w8**. Therefore, the entry "psr mark order" in the dictionary receives the value "head-initial."

Identifying potential possessor features

This function first collects all morphemes that align to the possessor (that is, the possessor word can be segmented into these morphemes). It then adds to the dictionary an entry called "psr glosses", containing the glosses corresponding to each of these morphemes. Later, during the consolidation step, the algorithm will check to see whether any of these glosses appear to indicate genitive case, and use that information as evidence for whether the possessor is *required* to be in genitive case in this possessive strategy.[10] This function is called as long as the strategy is not one where the possessor is marked with an affix, in accordance with the logic used in the Grammar Matrix.

In the running Nafsan example, there is only one morpheme (**m10**) associated with

[10] Although currently the consolidation step only checks for genitive case, I designed this particular function more broadly so that any future work that aims to expand beyond just genitive case can do so during the consolidation step, without having to updating multiple points in the code.

the possessor word **w8**, as can be seen in example (8). The gloss associated with **m10**, "swamphen", is entered into the "psr glosses" dictionary.

(8)
```
<tier id="m" type="morphemes" segmentation="w">
  <item id="m1" segmentation="w1">me</item>
  <item id="m2" segmentation="w2">nlaken</item>
  <item id="m3" segmentation="w3">too</item>
  <item id="m4" segmentation="w4">ki1</item>
  <item id="m5" segmentation="w4">pe</item>
  <item id="m6" segmentation="w5">lewi</item>
  <item id="m7" segmentation="w5">-ki</item>
  <item id="m8" segmentation="w6">kom</item>
  <item id="m9" segmentation="w7">ni</item>
  <item id="m10" segmentation="w8">taapes</item>
</tier>
<tier id="g" type="glosses" alignment="m" segmentation="gw">
  <item id="g1" alignment="m1" segmentation="gw1">and</item>
  <item id="g2" alignment="m2" segmentation="gw2">because</item>
  <item id="g3" alignment="m3" segmentation="gw3">chicken</item>
  <item id="g4" alignment="m4" segmentation="gw4">3S.PS1</item>
  <item id="g5" alignment="m5" segmentation="gw4">PF</item>
  <item id="g6" alignment="m6" segmentation="gw5">admire</item>
  <item id="g7" alignment="m7" segmentation="gw5">-TR</item>
  <item id="g8" alignment="m8" segmentation="gw6">comb.BI</item>
  <item id="g9" alignment="m9" segmentation="gw7">of</item>
  <item id="g10" alignment="m10" segmentation="gw8">swamphen</item>
</tier>
```

Inferring classes of possessor pronoun affixes

The adnominal possession library allows for two types of possessor pronoun classes: affixes on the possessum and non-affixes (i.e. separate words or clitics). In this work, I only attempted to infer affixed pronouns. Future work to expand inference to include non-affix possessor pronouns would likely be beneficial; several of the languages I examined during evaluation had such pronouns, and several examples of other such languages are mentioned in the typological survey of adnominal possession reviewed by Nielsen (2018).

The process for inferring classes of possessor pronoun affixes is similar to the process for inferring possessive strategies, but only calls a subset of the functions. Briefly, the process is as follows:

1. As mentioned in Section 3.2.1, the process for inferring possessor pronouns begins when a bilingual alignment is only provided for the possessum, not the possessor. At this point, the algorithm does not consider the possessive construction to necessarily be an instance of a possessor pronoun affix; it just *could* be.[11]

2. The algorithm checks whether the possessum has an affix that appears marked for possession by calling the `has_possession_marking_affix` function. If so, the word is now considered an example of a possessum with a possessor pronoun affix, and the following steps are carried out.

3. The mark type is recorded in the dictionary as "affix", and the identifier of the affix is recorded for future reference.

4. The functions `infer_spec_vs_mod` and `infer_agreement_and_orthography` are called.

[11] I make this assumption with the expectation that if an overt possessor were present in the sentence, INTENT would provide an alignment for it. Unfortunately, this is not always true currently, and there is a risk — though not one I observed consequences of in my analysis of my results — of classifying a construction as a possessor pronoun when it is really a possessive strategy.

Their behavior is identical in this case to when they are used for inferring possessive strategies; the dictionary likewise records whether the possessor affix appears to behave like a specifier or a modifier, as well as its orthography and any person grams.

5. The heuristic described earlier in this section for deciding whether an affix is a prefix or a suffix based on the position of the hyphen is performed.

The resulting dictionary is added to the longlist of possessor pronoun affix dictionaries. The next step, for both this longlist and the longlist of possessive strategy dictionaries, is consolidation.

3.2.2 Consolidating strategies and pronoun classes

At this point in the process, the system has compiled two lists of possible possessive constructions, one for strategies and one for pronoun affix classes, each represented as a list of dictionaries. Each dictionary corresponds to, and characterizes, a single possessive phrase observed within the training data. Some of these will be duplicates or near-duplicates of each other; these are likely multiple instances of the same possessive strategy or pronoun class. Others will be sufficiently different to be considered completely separate possessive strategies or pronoun classes. The goal of the consolidation stage is to consolidate (or de-duplicate) these lists, producing a shorter, streamlined set of strategies and pronoun classes that will ultimately be entered in the grammar specification.

I chose to consolidate possessive strategy instances according to their mark location, mark type, and whether or not the marker agrees. There are four possible mark locations: possessor-only, possessum-only, neither, and both. Possessor-marked-only instances are consolidated into up to four strategies (possessor marked by agreeing affix, by non-agreeing affix, by agreeing word, and by non-agreeing word), and likewise with possessum-marked-only instances. The system consolidates all neither-marked instances to a single strategy.

There can be up to sixteen both-marked strategies after consolidation, one for each possible combination of the values listed in (9). Therefore, the system can technically infer up to 25 different possessive strategies for a given language in total if the data were to show evidence for that many different types of possessive constructions, although in my experience it tends to infer fewer than five.

(9)
```
possessor marked by: [affix, non-affix]
possessor marker displays agreement: [yes, no]
possessum marked by: [affix, non-affix]
possessum marker displays agreement: [yes, no]
```

I took a simpler approach to consolidating pronoun classes, based on an assumption that individual instances of pronouns might make more intuitive sense to group in a single class than to spread out among multiple classes. The algorithm loops through all pronoun dictionaries in the longlist and consolidates them into a single dictionary. It's possible that some languages may benefit from adding multiple pronoun classes to the grammar specification (for example, languages with different sets of pronouns for alienable and inalienable possession), and future work might consider making this change.

In the interest of reducing spurious possessive strategies and pronoun classes, I check that any possessive strategy type has at least three pieces of evidence (that is, three IGT that demonstrate that possessive construction, represented by three dictionaries in the longlist) before adding the strategy to the consolidated list. I also check that there are at least three pronoun affix dictionaries before consolidating that list. I arrived at the requirement of three pieces of evidence through experimentation with various requirements on the development language data. I found that this requirement reduces ambiguity significantly compared to printing all de-duplicated strategies, since some of those strategies may be noise (often due to mislabeled IGT, incorrect word alignments, or complex phrases or sentences that my inference heuristics are not sophisticated enough to accurately handle — an example is

nested possession, such as in the phrase *my father's book*, in which *father* acts as both a possessor and a possessum). Because some datasets are small enough that they might not have three pieces of evidence, I added a loop such that if no strategies meet this requirement, the algorithm allows strategies with two pieces of evidence to count instead (and so on).

De-duplicating strategies and pronoun classes

When determining which strategies and pronoun classes are unique — versus which ones are variations of the same overall possessive construction — I look to the typological literature for guidance, as well as to the likelihood of noise in the data, and to the overall goal of ultimately increasing good parse coverage of inferred grammars. In this subsection, I describe some of the choices I've made in consolidating strategies that merit further explanation. The process is similar for strategies and pronoun affix classes, with pronoun affix classes requiring fewer steps.

- **Specifier-like versus modifier-like possessors**: If more than half of the instances in a given strategy or pronoun class have a "mod-spec" value of "spec" (that is, their possessor appears to act more like a specifier), I assign "spec" as the value for the entire strategy. In general, I aimed to bias the algorithm towards assigning the value of "spec" in keeping with the adnominal possession library's recommendation that this value be the default.[12] However, the exact threshold I set — 50% — is otherwise arbitrary.

- **Mark order**: Among the instances in a given strategy or pronoun class, whichever mark order (that is, order of the possession marker relative to the word it marks) is more common becomes the overall mark order for the strategy or pronoun class.

[12]See the adnominal possession library documentation page containing this recommendation at https://github.com/delph-in/docs/wiki/MatrixDoc_AdnominalPossession.

- **Feature constraints on the possessor**: I examine the lists of possessor features stored on appropriate strategies according to the Grammar Matrix customization questionnaire (that is, strategies in which neither possessor nor possessum is marked; only the possessum is marked; or the possessor is marked by a non-affix) and check for the presence of specific features that are likely to serve as constraints on the possessor. Currently, the only feature I check for is case, and specifically for the value "genitive".[13] If this feature is present on over half of the examples of a given strategy, I consider it a constraint on the possessor for that strategy. (The threshold was chosen somewhat arbitrarily and may be raised in future work, but noise in the data means that requiring 100 percent of examples to display the constraint will fail to return correct results.)

- **Agreement and orthography**: If a possessive strategy involves a possession marker that demonstrates agreement, its agreement features and their corresponding spellings are all added together to a single dictionary that belongs to the de-duplicated strategy. If the marker does not demonstrate agreement, the most common spelling observed among the instances of that strategy is the one added to the dictionary.[14]

- **Word order**: Because word order can vary within a single possessive strategy — according to both the customization questionnaire and typological evidence — if the algorithm sees two constructions within the same strategy that differ only in word order, it marks that strategy's word order as "either". The exception is that if a

[13]Genitive case is the only example of non-possession-specific possessor feature constraints mentioned in Nielsen 2018, and the only type of feature for which I had seen data while I was coding the system. However, during analysis of the held-out languages, I found a language that uses the dative case on the possessor to mark possession. This language provides motivation for future work to use this function to infer the presence of dative case and other features beyond genitive case.

[14]An implication of this approach to consolidating orthography is that if a language were to have two orthographically-different morphemes that occupy the same mark location and are of the same type (word or affix) but do not demonstrate agreement, only the most common of these spellings would be included in the grammar specification.

strategy marks neither the possessor nor the possessum, *and* there are no feature constraints on the possessor, the algorithm assigns the overall strategy whichever word order is more common among its instances. This exception exists to avoid ending up with a strategy in which there is no way to tell which word is the possessor and which is the possessum.

At the end of this step, the system has produced a list containing between zero and 25 possessive strategy dictionaries, each representing the consolidation of similar instances, as well as a list containing zero or one possessor pronoun affix dictionaries. The next step is to format the information in these lists in a way that can be consumed by the AGGREGATION code for printing to the grammar specification, specifically by defining choices that will appear in the adnominal possession and morphology portions of the grammar specification.

3.2.3 Preparing for the grammar specification

The final phase of the adnominal possession inference algorithm is to take the consolidated lists and print the information within to the grammar specification. Much of this step is formatting the output to match the expectations of the grammar specification, but there are a few steps that are slightly more complex that I explain in this subsection. In this step, possessive strategies and possessor pronoun affix classes are handled similarly; I call out differences in this section where appropriate.

Most of the output is entered in the `adnom-poss` section of the grammar specification; each possessive strategy and each possessor pronoun class gets it own entry in this section. Possessive strategies are named **poss-strat1, poss-strat2, ..., poss-strat**$_n$, where n is the total number of inferred possessive strategies, and possessor pronoun classes are named **poss-pron1, poss-pron2, ..., poss-pron**$_i$, where i is the total number of inferred possessor pronoun classes. The characteristics of each possessive strategy and possessor pronoun class are listed below its name in a format expected by the Grammar Matrix.

Possessive strategies that involve affixes, as well as possessor pronoun affix classes, additionally require setting values in the morphology section. Specifically, I create a noun position class for any distinct combination of possessive strategy or pronoun name plus affix position (prefix or suffix). I made the simplifying choice to hardcode the `input` value for each of these classes to be "noun". (See Chapter 4 for a discussion of why this decision fails to produce the correct output in words with multiple suffixes.) One candidate for future work is to integrate the output of the adnominal possession library into existing position classes rather than always creating new position classes; I expect this change can cut down on ambiguity and avoid the problems that requiring the input to be "noun" currently causes.

Within each position class, the algorithm always creates at least one lexical rule type, in accordance with the adnominal possession documentation in the Grammar Matrix customization questionnaire. This first lexical rule type marks the affix as having a possession-related feature: for possessive strategies, the feature name is the name of the strategy, and the value is either "possessor" or "possessum", depending on which entity the affix marks, while for possessor pronoun affixes, the feature name is the name of the pronoun class, and the value is "plus". The spelling of the affix is entered as a lexical rule instance. If the affix demonstrates agreement with another entity, the agreement features and corresponding spellings appear as subsequent lexical rule types and lexical rule instances. One note is that while the adnominal possession library instructions recommend making these position classes obligatory, and creating a corresponding "nonpossessive" lexical rule type, I had trouble getting that approach to work in inference; during development, it blocked almost all parses. Therefore, I currently do not mark any position classes I create as "obligatory", nor do I create any nonpossessive lexical rule types. It is possible that this decision is the source of some ambiguity.

Besides the representation of the morphology information, the other significant decision I have made in adding information to the grammar specification is to hardcode some values that were complicated to infer. Specifically, the "allow-pronoun" value (which asks whether

a possessor in a given strategy can be a pronoun)[15] is always set to "yes", and the "agree" value for possessor pronoun affixes (which asks whether the affix agrees with the possessum) is always set to "no". Future work may consider determining these values programmatically.

3.2.4 Summary

In this section, I have described the algorithm I implemented to infer adnominal possession, walking through the IGT examination phase, the consolidation phase, and the output phase. I now turn to describing my process for evaluating my changes.

3.3 Evaluation

As described in detail in Chapter 2, the inferred grammar specification can be used as input to the Grammar Matrix to produce a machine-readable grammar for a language. This grammar can then be used to parse test sentences using the LKB (Copestake, 2002) or ACE (Crysmann and Packard, 2012). I evaluate my changes to AGGREGATION both by directly comparing the inferred grammar specifications to expected outputs as well as by testing the ability of these grammar specifications to parse test sentences. The reason for this two-part evaluation process is to isolate problems that arise at various points in the AGGREGATION pipeline; for example, the inference system may produce a grammar specification that is similar to what a human user-linguist would produce manually, but the grammar specification may still fail to parse sentence due to issues with the interactions between different Grammar Matrix libraries or large amounts of ambiguity.

The current section gives an overview of the process I used to evaluate my changes, and the results themselves are given in Chapter 4. Subsection 3.3.1 describes the high-level

[15]Here, I mean a strategy in which the possession marker can mark a pronoun in addition to non-pronominal NPs. This is distinct from a "possessor pronoun", which is a pronoun whose meaning indicates possession. For example, in the English possessive strategy that marks possessors with the clitic *'s*, as in *Nadia's book*, the possessor cannot be a pronoun; **I's book* is ungrammatical.

metrics I aimed to improve, and Subsection 3.3.2 my approach to treebanking to evaluate good (semantically-appropriate) coverage.

The primary metric that I aimed to improve was validated parse coverage, briefly defined as increasing the percentage of test sentences that parse with at least one reading with correct semantics — that is, where the semantics of the sentence align with those in the linguist-annotated IGT. Other metrics I observe are total coverage (the amount of test sentences that parse at all), total ambiguity (the average number of parses per parsed test sentence), and grammar specification accuracy (how many of the inferred adnominal possession-related properties in the grammar specification matched my expectations).

3.3.1 Evaluating overall coverage and ambiguity

To get a quick picture of the impact my changes have on overall coverage and ambiguity, I use a testing harness developed by Elizabeth Conrad[16] to calculate these metrics across all ten folds. **Coverage** is here defined as the number of test sentences that have at least one parse, represented as both an absolute number and as a percentage over the total number of test sentences. **Ambiguity** is defined as the average number of parses per parsed sentence.

3.3.2 Treebanking to validate coverage

The coverage metric alone cannot guarantee that a *correct* parse has been found. The metric of validated parse coverage can help both (1) contextualize what amount of coverage increases are likely correct, and (2) catch improvements to the system that result in a newly-correct parse for a sentence that already parsed (but incorrectly), which would not be caught by the traditional coverage metric. To validate parse coverage, I perform manual treebanking using FFTB (Packard, 2015) and count sentences that have at least one parse that aligns with the semantics as indicated in the linguist-provided IGT.

[16]https://git.ling.washington.edu/agg/aggregation-evaluation

Because treebanking every test sentence would be prohibitively time-consuming, I follow a heuristic described by Howell (2020) and treebank only a pre-selected set of test sentences. Specifically, I create a test profile for each language that consists of all of the IGT items in the first fold that either have a translation dependency with the value "poss" or that have a word indicating possession (*my, your, his, her, its, our, their, 's*) in the translation. This method of selecting the test sentences may miss some sentences with possession in them and occasionally also includes sentences that have possession in the translation but not in the source language; in those cases, I remove the sentences if it is obvious that there is no possession in the source language sentence.

I first treebank every sentence in this profile that parses. If there are ten or more sentences that parse in the profile, I stop after this step. If there are fewer than ten sentences, I select a random set of the sentences that did not parse, such that the total number of sentences I am evaluating is ten or until I have examined all sentences in the profile. I then perform interactive unification using the LKB (Copestake, 2002) to see whether the reason the sentence does not parse is because of something wrong or missing with the adnominal possession inference system, or whether adnominal possession is behaving as expected (and the problem preventing the sentence from parsing is with some other part of the system). Finally, I report the number of sentences with a validated parse divided by the number of total hand-evaluated sentences as the value for validated coverage.

3.3.3 Summary

In this section, I have given an overview of my process for evaluating my changes to the inference system.

3.4 Summary

Having described both the algorithm I created to infer adnominal possession as well as my process for evaluating this work, I hope that these descriptions provide context for the reader against which to consider the results I report and analyze in Chapter 4.

Chapter 4

RESULTS

This chapter presents an evaluation of the adnominal possession inference changes I contributed to AGGREGATION, following the evaluation process described in the previous chapter. I first present a high-level summary and analysis of the overall results, then the results and error analysis per language in more detail.

4.1 Overall results

The primary metrics I am evaluating across each language are as follows:[1]

- **Coverage:** How many of the test sentences parse?
- **Ambiguity:** What is the average number of readings per parsed sentence?
- **Validated coverage:** For a select set of sentences, how many of them parse with appropriate semantics?
- **Grammar specification accuracy:** Do the inferred grammar specifications capture the expected adnominal possession strategies for this language?

I provide the results for grammar specification accuracy within the discussion for each language. The other metrics are summarized in Table 4.1 (for development languages) and Table 4.2 (for held-out languages), along with their performance with the inferred system as a relative difference compared to the coverage and ambiguity using the baseline system (i.e.

[1]Specific definitions for each metric are provided in Chapter 3.

the system without my code). For "Validated coverage", I show the number of sentences that parsed with correct semantics out of all of the sentences in the first test fold that appear to contain possession; for Wambaya specifically, I did not measure validated coverage because the code produced no inference (i.e. printed nothing novel to the grammar specification).[2] The sections in this chapter discussing the results of each language provide more detail about the validated coverage process, results, and analysis.

Language	Coverage			Ambiguity	
	Absolute	Change	Validated	Absolute	Change
Abui (abz)	33.35%	-0.19%	2/14	295.99	134.54%
Tsova-Tush (bbl)	16.25%	-0.37%	0/20	212.11	28.90%
Nafsan (erk)	6.72%	-13.70%	0/43	10990.77	21.69%

Table 4.1: Results for development languages

Language	Coverage			Ambiguity	
	Absolute	Change	Validated	Absolute	Change
Matsigenka (mcb)	2.58%	0%	0/8	390.56	-33.69%
Wambaya (wmb)	1.47%	0%	N/A	2.92	0%
Hiaki (yaq)	8.77%	0%	0/8	37.23	20.56% %

Table 4.2: Results for held-out languages

In general, the system's coverage is the same or worse following my changes. The coverage loss is due to the significant increase in ambiguity that my changes add to the system; sentences with too many potential readings cause the parser to run out of resources, resulting

[2]None of these sentences parsed with correct semantics prior to my changes.

in zero parses for many sentences that had previously parsed. This effect is more pronounced in languages for which the system already produced a high level of ambiguity (such as Nafsan) than for languages that began with a comparatively low level of ambiguity (such as Abui and Tsova-Tush, which each lost one sentence in coverage).

Increased ambiguity is expected,[3] although this large an increase is not desirable due to both the loss of coverage discussed in the previous paragraph as well as inherent complexity resulting from high ambiguity (see Conrad (2021) for discussion). Several of the rules in the adnominal possession library introduce a large amount of ambiguity, and second, my inference changes often add more than one new position class, none of which are marked as obligatory, compounding the ambiguity. The ambiguity could be improved (i.e. decreased) in future work by integrating the adnominal possession inference output with existing position classes and lexical rules inferred by MOM, rather than creating new position classes and lexical rules. Similarly, future contributors to the AGGREGATION project might consider continuing the work of Conrad (2021) in aiming to decrease ambiguity caused by the inference system across the board; such efforts could make it easier to evaluate the system by reducing the risk that the parser runs out of resources.

4.2 Results per language

In this section, I describe how each language whose data I used handles adnominal possession, primarily relying on reference grammars as sources. Based on my understanding of each language's approach to possession, I describe what I would expect the possession-related portions of a grammar specification for the language to look like. For the three held-out languages, I wrote these descriptions before viewing IGT data or the results of the experiment,

[3]The decrease in ambiguity for Matsigenka is due to a similar issue as the loss of coverage I discussed in the previous paragraph. For one sentence in the Matsigenka test dataset, ambiguity rose significantly, causing the parser to run out of resources. However, unlike with Nafsan, Abui, and Tsova-Tush, in this case the parser was able to complete the process of constructing the packed parse forest (hence no loss in coverage); it was just unable to enumerate (unpack) all of the possible parses (hence the loss in ambiguity).

so they can function like predictions against which I can compare the actual grammar specifications that my system inferred. Next, I present the results for that language: first, the relevant portions of the grammar specifications,[4] followed by a summary of the results of the treebanking evaluation, and finally a discussion of the specific issues and errors that I discovered during analysis of the results.

4.2.1 Abui (development)

Abui (abz) is a Papuan language belonging to the Timor-Alor-Pantar family and spoken in Indonesia. In Abui, a possessum is marked with a prefix that carries the person (first, second, and third) and number (only in first and second person) information of the possessor. There is an inclusive/exclusive distinction in first person plural, two third person prefixes whose use depends on whether the possessor is coreferent with the actor argument, and different sets of prefixes depending on whether the possessum is inalienably possessed. A non-pronominal third-person possessor can optionally appear before the possessum, which in this situation is still marked with the same third-person pronominal prefix as when the possessor is not overtly stated. Finally, these prefixes share orthographies with prefixes that can attach to verbs, in which case their meaning is not possessive (Kratochvíl, 2007).

Examples of possessive noun phrases with overt and non-overt third-person possessors are shown in (10) and (11).

(10) maama he-sepeda
father 3II.AL-bike
"father's bike"

[4]By "relevant portions", I mean the entirety of the `adnom-poss` section, which lists the choices characterizing the possessive strategies and possessor pronoun class, as well as any new position classes my inference module added within the `morphology` section, the latter of which is only the case if the possessive strategies or pronoun classes involve affixes. This is because the morphology section of a grammar specification is where the choices that will eventually (when the grammar specification is used to create a fully-fledged grammar) turn into rules governing affix behavior are listed.

(11) ne-fala
1SG.AL-house
"my house" (Kratochvíl, 2007, p. 139)

Based on the description given by Kratochvíl, I would expect a grammar specification for Abui to include one possessive strategy as well as one or two classes of possessor pronouns (depending on whether the linguist wishes to separate alienable from inalienable possessive prefixes, a distinction which is not fully supported by the Grammar Matrix). The possessive strategy should reflect that the possessum is marked with an affix that agrees in person with the possessor, and that the possessor precedes the possessum. Because the prefix demonstrates agreement, this strategy should be accompanied by a noun position class that attaches a prefix that carries a "possessum" feature (in accordance with the expectations of the Grammar Matrix's adnominal possession library), as well as inflecting rules that list the possible third-person prefixes. The possessive pronoun class(es) should be accompanied by one or more lexical rule types that correspond to each person-number combination with the appropriate orthographies.

Results for Abui

The grammar specification that my adnominal possession inference system infers (excerpted in (12)) includes the expected possessive strategy and possessor pronoun class, without generating any spurious rules. Specifically, the `adnom-poss` section contains a possessive strategy, **poss-strat1**, in which the possessum is preceded by the possessor ("poss-strat1_order=head-final") and the possessum is marked by an affix that agrees with the possessor. While those values are correct, **noun-pc24**, which provides the choices for the lexical rule governing this affix, only shows a value for the third-person affix rather than for all possible person values (see the discussion of Abui results for commentary on why this likely happened). The grammar specification also correctly includes a possessor pronoun affix class, **poss-pron1**;

the spellings and associated person values for this affix class are listed in **noun-pc26**. Of course, the system only infers person information, and fails to provide any information about the number distinction in the Abui first- and second-person possessive affixes.[5]

(12)
```
section=adnom-poss
  poss-strat1_order=head-final
  poss-strat1_mod-spec=spec
  poss-strat1_mark-loc=possessum
  poss-strat1_possessum-type=affix
  poss-strat1_possessum-affix-agr=agree
  poss-pron1_type=affix
  poss-pron1_agr=non-agree
  poss-pron1_mod-spec=spec
section=morphology
    noun-pc24_name=noun-pc24
    noun-pc24_order=prefix
    noun-pc24_inputs=noun
    noun-pc24_lrt1_name=noun-pc24_lrt1
      noun-pc24_lrt1_feat1_name=poss-strat1
      noun-pc24_lrt1_feat1_value=possessum
      noun-pc24_lrt1_feat1_head=itself
      noun-pc24_lrt1_feat2_name=person
      noun-pc24_lrt1_feat2_value=3rd
      noun-pc24_lrt1_feat2_head=possessor
      noun-pc24_lrt1_lri1_inflecting=yes
```

[5]In general, I avoid commenting on the correctness (or lack thereof) of the **mod-spec** value throughout the languages in this chapter, as I find it often not obvious from reference grammars whether a possessor should be analyzed as behaving more like a specifier or a modifier. At any rate, my system biases heavily towards assigning the value "spec", in accordance with the suggested default given by the adnominal possession library in the Grammar Matrix, and all results discussed in this chapter have this value.

```
  noun-pc24_lrt1_lri1_orth=he-
noun-pc26_name=noun-pc26
noun-pc26_order=prefix
noun-pc26_inputs=noun
  noun-pc26_lrt1_name=noun-pc26_lrt1
    noun-pc26_lrt1_feat1_name=poss-pron1
    noun-pc26_lrt1_feat1_value=plus
    noun-pc26_lrt1_feat1_head=itself
    noun-pc26_lrt1_feat2_name=person
    noun-pc26_lrt1_feat2_value=2nd
    noun-pc26_lrt1_feat2_head=itself
    noun-pc26_lrt1_lri1_inflecting=yes
    noun-pc26_lrt1_lri1_orth=e-
    noun-pc26_lrt1_lri2_inflecting=yes
    noun-pc26_lrt1_lri2_orth=a-
  noun-pc26_lrt2_name=noun-pc26_lrt2
      noun-pc26_lrt2_feat1_name=poss-pron1
      noun-pc26_lrt2_feat1_value=plus
      noun-pc26_lrt2_feat1_head=itself
      noun-pc26_lrt2_feat2_name=person
      noun-pc26_lrt2_feat2_value=3rd
      noun-pc26_lrt2_feat2_head=itself
      noun-pc26_lrt2_lri1_inflecting=yes
      noun-pc26_lrt2_lri1_orth=ha-
      noun-pc26_lrt2_lri2_inflecting=yes
      noun-pc26_lrt2_lri2_orth=he-
      noun-pc26_lrt2_lri3_inflecting=yes
      noun-pc26_lrt2_lri3_orth=de-
```

```
   noun-pc26_lrt2_lri4_inflecting=yes
   noun-pc26_lrt2_lri4_orth=ho-
noun-pc26_lrt3_name=noun-pc26_lrt3
   noun-pc26_lrt3_feat1_name=poss-pron1
   noun-pc26_lrt3_feat1_value=plus
   noun-pc26_lrt3_feat1_head=itself
   noun-pc26_lrt3_feat2_name=person
   noun-pc26_lrt3_feat2_value=1st
   noun-pc26_lrt3_feat2_head=itself
   noun-pc26_lrt3_lri1_inflecting=yes
   noun-pc26_lrt3_lri1_orth=ne-
   noun-pc26_lrt3_lri2_inflecting=yes
   noun-pc26_lrt3_lri2_orth=na-
```

The possession-specific profile I created from the first fold of Abui test data had fourteen total sentences, five of which parsed. Upon treebanking, I found that two of these sentences had the correct semantics, including for possession. In both of these sentences, the possessor was third-person, and this person information is correctly included in the semantics; in Abui, because there is no number distinction in the third-person possessum-marking affixes, the fact that my system does not yet infer number was not a problem for these sentences. It was, however, a problem for the other three sentences that parsed, all of which had a first-person possessor; for these sentences, the possessive prefix applied as expected and carried the correct person information, but no number information is included.

I also reviewed five random sentences from the list of those that did not parse. Three of these had a gap in lexical coverage unrelated to possession, and the possession syntax and semantics behaved as expected (that is, person information is correctly included in the semantics, but number information is not). The other two exposed an issue with adnominal possession inference, which I discuss in the following subsection.

Discussion of Abui results

The system infers much of the expected values for the Abui grammar specification, and adds the proper possession syntax and semantics to almost all of the hand-reviewed sentences. The only issue I discovered during analysis was that some possessive prefixes were not included in the grammar specification. The second-person-plural-inalienable prefix *ri-* was not inferred; it only occurs once in the training data for this fold, and the system did not pick up on that example (shown in [13]) because the translation line for that IGT does not contain possession.

(13) na ri-areng mia
1SG.AGT 2PL.AL-top be.in
"I am above you." (Kratochvíl, 2019)

Another sentence that does not properly parse with respect to possession is (14).

(14) edo a-ne maa re baai
2SG.FOC 2SG.INAL-name who or also
"What is your name again" (Kratochvíl, 2019)

This sentence contains other phenomena that the inference system cannot yet handle, but the possession-specific issue is that the grammar is unable to parse the combination of the second-person-singular pronominal subject and the second-person-singular-inalienable prefix on the possessum. This is because there are no examples in the training data of an overt second-person possessor coexisting with a possessum, so the grammar specification (see (12)) does not contain any lexical rule types in associated with **poss-strat1** in which the value of *person* is anything but "3rd". However, the reference grammar I used for Abui (Kratochvíl, 2007) only mentions third-person overt possessors, so it is unclear to me exactly how to analyze this construction based on my understanding of this language's approach to possession. It is possible that other phenomena (such as focus, which the first word in this sentence is glossed for) are at hand here.

4.2.2 Tsova-Tush (development)

Tsova-Tush (bbl), also called Batsbi, is a Northeast Caucasian language spoken in the country of Georgia that expresses possession by having the possessor be in the genitive case. This approach applies to both pronominal and non-pronominal possessors (Holisky and Gagua, 1994).

The adnominal possession library in the Grammar Matrix treats this kind of possessive approach (in which the possessor is required to have some feature) as a possessive strategy in which neither the possessor nor the possessum is marked. This is because the feature constraint on the possessor may not be specific to possession and needs to be accessible in other contexts (Nielsen, 2018). I therefore would expect an appropriate grammar specification for Tsova-Tush to have only one possessive strategy, in which "neither" entity is marked, the possessor can be a pronoun, and the strategy includes a feature for genitive case (which the adnominal possession library takes as an indication that the feature is a constraint on the possessor).

Results for Tsova-Tush

The inferred grammar specification for the first fold of the Tsova-Tush test data meets the expectation described in the previous subsection. The relevant excerpt is shown in (15). The grammar specification correctly lists a head-final possessive strategy in which neither the possessor nor the possessum is specifically marked for possession, but in which the possessor is required to be in genitive case and can be a pronoun. Insofar as it completely matches the expected possessive strategy I described above, I consider this output correct. However, as I will describe in the following paragraphs, there are still problems that adversely impact the resulting grammar's ability to parse sentences with adnominal possession.

(15)
```
section=adnom-poss
  poss-strat1_order=head-final
```

```
poss-strat1_mod-spec=spec
poss-strat1_mark-loc=neither
poss-strat1_pronoun-allow=yes
  poss-strat1_feat1_name=case
  poss-strat1_feat1_value=gen
```

The possession-specific profile that I created from the first fold of Tsova-Tush test data contains twenty sentences, none of which parse. I chose ten of these sentences at random to interactively unify and investigate whether possession inference might be involved in their failure to parse. In each of these ten sentences, the possessor is a pronoun. None of these ten sentences successfully applied the possessive rule to unify the possessor and possessum, for a variety of reasons. In the following subsection, I break down the causes I was able to identify for these failures.

Discussion of Tsova-Tush results

I identified two broad categories of errors in the Tsova-Tush sentences that I analyzed:

1. The grammar does not recognize that the possessor is in the genitive case.

2. The possessor is not marked for genitive case.

The first category appears to involve errors successfully applying the lexical rule that marks a word as being marked for genitive case. After some investigation, I suspect that these errors have to do with morphological inference in AGGREGATION mistakenly requiring that these pronouns be marked for ergative or locative case, as well as with some issues around lexical inference of pronouns. When I manually edit the grammar to remove the mistakenly obligatory case rule and change the orthography of the pronoun's lexical entry, the possessive rule applies as expected. Therefore, I expect that if case and pronoun inference are improved

in the future, these sentences will be able to parse with proper possessive semantics without further changes needed to the adnominal possession inference system.

The second category involves words that are translated as possessors but that are not marked for genitive case. Some of these words are glossed "REFL.POSS.NOM", indicating a reflexive possessive pronoun in nominative case. Another is glossed "my own". I am unable to find more information about these words in the reference grammar I am using; however, the reference grammar does state that reflexives have not been well-studied in Tsova-Tush (Holisky and Gagua, 1994). The words do not have lexical entries in the inferred grammar, and while they appear in the training data, they are usually glossed as a reflexive word with no possessive meaning in English (so the adnominal possession inference library does not consider them examples of possession). However, even if the translation were to indicate possession, it is possible that my system would still fail to infer these; they appear to be a class of possessive pronouns that are separate words, and the system currently only infers classes of possession pronouns that are affixes on the possessum. It is possible that expanding the inference system to include non-affix possessive pronoun classes could result in a gain of coverage in this area, but because of the other issues I have mentioned regarding these words, it's not certain that those changes would fully resolve this problem.

4.2.3 Nafsan (South Efate) (development)

According to the website of Nick Thieberger, the author of the reference grammar I am using, Nafsan was declared by some of its speakers in 2015 to be the name of this language, which is also known as South Efate.[6] Nafsan is an Oceanic language spoken in Vanuatu that has several different ways of expressing adnominal possession for indirectly possessed nouns (Thieberger, 2006a):[7]

---

[6] https://nthieberger.net/sefate.html

[7] "Directly/indirectly possessed" is the language used in the reference grammar; I assume this is equivalent to an inalienable/alienable distinction.

1. a class of possessive pronouns that are separate words from the possessum and follow the possessum,

2. the first-person singular possessive word *nakte* ("my"), which precedes the possessum and according to Thieberger is used only rarely,

3. a non-inflecting possessive word *ni* ("of") that appears between the possessum and possessor,

4. a non-inflecting possessive word *knen* ("of it") that indicates some non-overt inanimate third-person singular possessors, and

5. simple juxtaposition of possessum and possessor without marking.

Additionally, directly possessed nouns (such as some kinship terms and body parts) are marked for possession via possession-specific suffixes that indicate the person and number of the possessor. Occasionally, this suffix is preceded by "a synchronically unpredictable vowel" (Thieberger, 2006a, p. 121) that is glossed as "V".

Of these approaches to possession, I consider only the third and fifth strategies for indirectly possessed nouns, as well as the set of possessive suffixes for directly possessed nouns, in scope for this thesis. The first and second strategies for indirectly possessed nouns, the classes of possessor pronouns, should be able to be modeled by the adnominal possession library, and the fourth strategy could potentially be modeled by the adnominal possession library if the possessive word were interpreted as a third-person possessor pronoun. However, I have chosen not to infer classes of possessor pronouns that are separate words for the purposes of this thesis, so I do not expect the grammar specification to reflect these constructions. With that in mind, I would expect the inferred grammar specification to contain two possessive strategies: one in which the possessor is preceded by the marker word *ni*, and

one in which the possessum and possessor are simply juxtaposed in that order. I also expect to see a class of possessor pronouns that corresponds to the possessive suffixes on directly possessed nouns, along with the appropriate noun position class and lexical rule types that match the spellings of these suffixes to the correct agreement features. (An ideal grammar specification would contain information about both person and number on the suffixes; at this time, I have only implemented inference for person agreement).

Results for Nafsan

The grammar specification for Nafsan includes the expected possessive strategies and pronoun class, without inferring spurious strategies. The relevant excerpt is showing in (16). In the `adnom-poss` section, **poss-strat1** corresponds to the "third strategy" discussed in the previous paragraph (i.e. the one in which a non-inflecting possessive word *ni* ("of") appears between the possessum and possessor), while **poss-strat2** corresponds to the "fifth strategy" (i.e. simple juxtaposition of the possessum and possessor). Neither strategy requires any changes to the `morphology` section, as neither involves an affix. However, the possessive pronoun affix class **poss-pron1** is associated with three lexical rule types in the `morphology` section, one for each person value (but lacking distinction by number value). Some of the listed orthographies are correct (for instance, "noun-pc37_lrt1_lri1_orth=-n"), while others are incorrect. I give a possible reason for the incorrect orthographies in the discussion subsection below.

(16)
```
section=adnom-poss
    poss-strat1_order=head-initial
    poss-strat1_mod-spec=spec
    poss-strat1_mark-loc=possessor
    poss-strat1_pronoun-allow=yes
    poss-strat1_possessor-type=non-affix
```

```
  poss-strat1_possessor-mark-order=head-initial
  poss-strat1_possessor-agr=non-agree
  poss-strat1_possessor-orth=ni
  poss-strat2_order=head-initial
  poss-strat2_mod-spec=spec
  poss-strat2_mark-loc=neither
  poss-strat2_pronoun-allow=yes
  poss-pron1_type=affix
  poss-pron1_agr=non-agree
  poss-pron1_mod-spec=spec
section=morphology
  noun-pc37_name=noun-pc37
  noun-pc37_order=suffix
  noun-pc37_inputs=noun
    noun-pc37_lrt1_name=noun-pc37_lrt1
      noun-pc37_lrt1_feat1_name=poss-pron1
      noun-pc37_lrt1_feat1_value=plus
      noun-pc37_lrt1_feat1_head=itself
      noun-pc37_lrt1_feat2_name=person
      noun-pc37_lrt1_feat2_value=3rd
      noun-pc37_lrt1_feat2_head=itself
      noun-pc37_lrt1_lri1_inflecting=yes
      noun-pc37_lrt1_lri1_orth=-n
      noun-pc37_lrt1_lri2_inflecting=yes
      noun-pc37_lrt1_lri2_orth=-r
      noun-pc37_lrt1_lri3_inflecting=yes
      noun-pc37_lrt1_lri3_orth=ga
      noun-pc37_lrt1_lri4_inflecting=yes
```

```
  noun-pc37_lrt1_lri4_orth=i1
noun-pc37_lrt2_name=noun-pc37_lrt2
  noun-pc37_lrt2_feat1_name=poss-pron1
  noun-pc37_lrt2_feat1_value=plus
  noun-pc37_lrt2_feat1_head=itself
  noun-pc37_lrt2_feat2_name=person
  noun-pc37_lrt2_feat2_value=2nd
  noun-pc37_lrt2_feat2_head=itself
  noun-pc37_lrt2_lri1_inflecting=yes
  noun-pc37_lrt2_lri1_orth=-m
  noun-pc37_lrt2_lri2_inflecting=yes
  noun-pc37_lrt2_lri2_orth=pa1
noun-pc37_lrt3_name=noun-pc37_lrt3
  noun-pc37_lrt3_feat1_name=poss-pron1
  noun-pc37_lrt3_feat1_value=plus
  noun-pc37_lrt3_feat1_head=itself
  noun-pc37_lrt3_feat2_name=person
  noun-pc37_lrt3_feat2_value=1st
  noun-pc37_lrt3_feat2_head=itself
  noun-pc37_lrt3_lri1_inflecting=yes
  noun-pc37_lrt3_lri1_orth=-k
  noun-pc37_lrt3_lri2_inflecting=yes
  noun-pc37_lrt3_lri2_orth=-kit
```

The possession-specific profile I created for Nafsan contains forty-three sentences, two of which parsed using the inferred grammar. Neither of these two sentences have the proper semantics. However, one of them does have proper syntax and semantics for adnominal possession; the problem appears to be with non-possession related lexical coverage. In example

(17), possession is indicated via juxtaposition of the possessum and possessor:

(17) go nɲ̃au wit ki1-mer tul
and head octopus 3s.ps1-again sway
"And the octopus s head was swaying." (Thieberger, 2006b)[8]

The other sentence that parses fails to display the expected syntax for several reasons, at least two of which appear to be related to adnominal possession. These reasons are discussed in the following subsection.

I also examined a randomly-selected set of eight sentences from the 41 that did not parse. Of these eight sentences, one of them displayed the correct syntax and semantics for its adnominal possessive phrase: *nam̃er ni Ermag*, glossed as "people of Erakor." The other seven displayed possession-related issues, which I discuss in the following subsection.

Discussion of Nafsan results

The inferred grammar specification for Nafsan displayed the expected possessive strategies and possessor pronoun class, except for some spurious affix orthographies. I believe these may be caused by instances in the training data wherein the possessive phrase is next to a non-possessive word with a affix whose gloss indicates person value. Although I tried to use part-of-speech tags in the code to ensure that the algorithm only considers nouns that have affixes glossed for person (as opposed to verbs, which inflect for person for reasons not related to possession), this constraint was not airtight. INTENT labels some words that appear to be verbs as nouns, so the system may have picked up on those and mistakenly included their affixes.

In my analysis of the test sentences, I discovered three issues with adnominal possession inference:

[8]I have reproduced the translation line exactly as it appears in the data; INTENT is still able to detect that "octopus s" is meant to denote a possessive relationship in English despite the lack of apostrophe.

1. Five sentences included a possessor pronoun that is a separate word following the possessum.

2. Four sentences included a directly-possessed noun that have not only a possessor suffix but also a vowel infix preceding the suffix.

3. One sentence includes a directly-possessed noun that also has a determiner, but Nafsan was inferred to be a language where possessors act like specifiers (that is, they cannot coexist with determiners on the possessum).

The first issue is expected and due to my choosing not to implement inference for possessor pronouns that are separate words from the possessum. The second issue involves the unpredictable vowels that I discussed in the first subsection for Nafsan. This issue might be addressed by integrating the output of the adnominal possession inference module with the existing position classes inferred by MOM, so that the noun could take both the vowel suffix and the possessive suffix. Finally, the third issue may be that the system is heavily biased toward labeling languages as having specifier-like possessors in the absence of strong evidence in the other direction. Future work could explore changing the threshold for determining this label or adding more robust heuristic tests.

Having presented and discussed the results for the development languages, I next do the same for the held-out languages. While I was able to review the development language reference grammars and results while working on the code, I only interacted with held-out language reference grammars or data after having finished coding.

4.2.4 Matsigenka (held-out)

Matsigenka (mcb) is an Arawak language, spoken in Peru, that is closely related to Nanti. Because I am unable to locate a reference grammar for Matsigenka, I am using a description

of possession in Nanti (Michael, 2012) for reference and assuming that this description applies to Matsigenka as well.

Nanti handles possession in the following ways, according to Michael (2012):

- The possessum is marked with a possessive prefix that indicates the person, gender, and (in first person only) number of the possessor.

- The marked possessum can be accompanied by an optional overt possessor. If the overt possessor is a pronoun, it precedes the possessum. If the overt possessor is non-pronominal, it normally follows the possessum, but can precede it to indicate constrastive focus.

- Alienable possessums also have a non-inflecting suffix to indicate their alienability in addition to the possessive prefix. These suffixes have different spellings depending on the syllabic length of the possessum.

Based on this description, I would expect a grammar specification for Matsigenka to include the following information:

- A possessive strategy in which the possessum is marked with a prefix that agrees with the possessor on the basis of person (ideally also gender and number, but I have not implemented those features as part of my thesis). Word order in this possessive strategy should be head-initial.

- A second possessive strategy that is identical to the first except for that word order is head-final, and the possessor is not allowed to be a pronoun. This strategy cannot be fully inferred by the current system, as I currently assume that possessors are always allowed to be pronouns.

- A set of possessive prononominal prefixes that indicate the person (ideally also gender and number) of the possessor.

- Ideally the grammar specification would include inflectional rules that make one of the alienable suffixes obligatory for alienable nouns. My code does not handle this case; changes to AGGREGATION to infer separate noun position classes for alienable and inalienable possessive markers (which may also involve creating separate noun types that each take a different type of possession) may be of interest as future work.

Results for Matsigenka

The inferred grammar specification (18) includes the expected possessum-marking prefixes via a possessive strategy (**poss-strat1**) and a possessor pronoun class (**poss-pron1**), but also unexpectedly suggests that the possessor (when overt) must have a prefix that agrees with the possessum. This suggestion is reflected in the incorrect line "poss-strat1_mark-loc=both" (where the expected value is "possessum"). As such, **lrt1** and **lrt2** under **noun-pc24** in the `morphology` section should not be present in the choices file. I discuss below a reason this incorrect inference might have occurred.

(18)
```
section=adnom-poss
    poss-strat1_order=head-initial
    poss-strat1_mod-spec=spec
    poss-strat1_mark-loc=both
    poss-strat1_possessor-type=affix
    poss-strat1_possessor-affix-agr=agree
    poss-strat1_possessum-type=affix
    poss-strat1_possessum-affix-agr=agree
    poss-pron1_type=affix
    poss-pron1_agr=non-agree
```

```
  poss-pron1_mod-spec=spec
section=morphology
    noun-pc24_name=noun-pc24
    noun-pc24_order=prefix
    noun-pc24_inputs=noun
        noun-pc24_lrt1_name=noun-pc24_lrt1
            noun-pc24_lrt1_feat1_name=poss-strat1
            noun-pc24_lrt1_feat1_value=possessor
            noun-pc24_lrt1_feat1_head=itself
            noun-pc24_lrt1_feat2_name=person
            noun-pc24_lrt1_feat2_value=3rd
            noun-pc24_lrt1_feat2_head=possessum
            noun-pc24_lrt1_lri1_inflecting=yes
            noun-pc24_lrt1_lri1_orth=i-
        noun-pc24_lrt2_name=noun-pc24_lrt2
            noun-pc24_lrt2_feat1_name=poss-strat1
            noun-pc24_lrt2_feat1_value=possessor
            noun-pc24_lrt2_feat1_head=itself
            noun-pc24_lrt2_feat2_name=person
            noun-pc24_lrt2_feat2_value=3rd
            noun-pc24_lrt2_feat2_head=possessum
            noun-pc24_lrt2_lri1_inflecting=yes
            noun-pc24_lrt2_lri1_orth=o-
        noun-pc24_lrt3_name=noun-pc24_lrt3
            noun-pc24_lrt3_feat1_name=poss-strat1
            noun-pc24_lrt3_feat1_value=possessum
            noun-pc24_lrt3_feat1_head=itself
            noun-pc24_lrt3_feat2_name=person
```

```
        noun-pc24_lrt3_feat2_value=3rd
        noun-pc24_lrt3_feat2_head=possessor
        noun-pc24_lrt3_lri1_inflecting=yes
        noun-pc24_lrt3_lri1_orth=o-
    noun-pc24_lrt4_name=noun-pc24_lrt4
        noun-pc24_lrt4_feat1_name=poss-strat1
        noun-pc24_lrt4_feat1_value=possessum
        noun-pc24_lrt4_feat1_head=itself
        noun-pc24_lrt4_feat2_name=person
        noun-pc24_lrt4_feat2_value=3rd
        noun-pc24_lrt4_feat2_head=possessor
        noun-pc24_lrt4_lri1_inflecting=yes
        noun-pc24_lrt4_lri1_orth=i-
noun-pc26_name=noun-pc26
noun-pc26_order=prefix
noun-pc26_inputs=noun
    noun-pc26_lrt1_name=noun-pc26_lrt1
        noun-pc26_lrt1_feat1_name=poss-pron1
        noun-pc26_lrt1_feat1_value=plus
        noun-pc26_lrt1_feat1_head=itself
        noun-pc26_lrt1_feat2_name=person
        noun-pc26_lrt1_feat2_value=3rd
        noun-pc26_lrt1_feat2_head=itself
        noun-pc26_lrt1_lri1_inflecting=yes
        noun-pc26_lrt1_lri1_orth=o-
        noun-pc26_lrt1_lri2_inflecting=yes
        noun-pc26_lrt1_lri2_orth=i-
        noun-pc26_lrt1_lri3_inflecting=yes
```

```
    noun-pc26_lrt1_lri3_orth==ri
    noun-pc26_lrt1_lri4_inflecting=yes
    noun-pc26_lrt1_lri4_orth==ro
noun-pc26_lrt2_name=noun-pc26_lrt2
    noun-pc26_lrt2_feat1_name=poss-pron1
    noun-pc26_lrt2_feat1_value=plus
    noun-pc26_lrt2_feat1_head=itself
    noun-pc26_lrt2_feat2_name=person
    noun-pc26_lrt2_feat2_value=2nd
    noun-pc26_lrt2_feat2_head=itself
    noun-pc26_lrt2_lri1_inflecting=yes
    noun-pc26_lrt2_lri1_orth=pi-
noun-pc26_lrt3_name=noun-pc26_lrt3
    noun-pc26_lrt3_feat1_name=poss-pron1
    noun-pc26_lrt3_feat1_value=plus
    noun-pc26_lrt3_feat1_head=itself
    noun-pc26_lrt3_feat2_name=person
    noun-pc26_lrt3_feat2_value=1st
    noun-pc26_lrt3_feat2_head=itself
    noun-pc26_lrt3_lri1_inflecting=yes
    noun-pc26_lrt3_lri1_orth=no-
    noun-pc26_lrt3_lri2_inflecting=yes
    noun-pc26_lrt3_lri2_orth=ige
    noun-pc26_lrt3_lri3_inflecting=yes
    noun-pc26_lrt3_lri3_orth=icha
    noun-pc26_lrt3_lri4_inflecting=yes
    noun-pc26_lrt3_lri4_orth=incho
```

The possession-specific profile I created for Matsigenka contains the eight sentences from

the first fold of the test data that indicate possession. None of these sentences parse with the inferred grammar. I manually inspected all eight in the LKB and found that one sentence displays the expected syntax and semantics for its possessive phrase. The phrase in question is in (19). This phrase is correctly analyzed as a possessum that is marked by a possessor pronoun. The other seven sentences fail to parse for a variety of reasons, which I discuss in the next subsection.

(19) ijina
i-*jina
3MP-wife
"his wife" (Michael et al., 2013)

Discussion of Matsigenka results

The inferred grammar specification for Matsigenka matches my expectations except for its declaration that the possessor must be marked when it is overt, which does not match with my understanding of adnominal possession as described in the Nanti text (Michael, 2012). It is possible that Matsigenka and Nanti differ on this point. However, I looked through the Matsigenka data and found several examples of possessors that are not marked, so I suspect the inference system is producing an incorrect result. My hypothesis for the cause of the issue is that many examples of possession in the training data include a possessor that is itself possessed, and therefore is marked in its capacity as a possessor but not as a possessum. At one point, I had a function in my algorithm that discarded possessors that were themselves possessed, but I removed the function in favor of instead requiring more evidence that a possessor was marked before entering that as a possessive strategy. While my decision was acceptable for my development languages, it seems not to be for Matsigenka, so a potential fix might be to bring that function back or to create a similar one (with the basic goal of more thoroughly whether morphemes on a possessor are serving some other purpose before inferring that their function is to mark possession).

In analyzing the sentences that did not parse, I identified three issues:

1. 2 sentences have an overt possessor that is not marked.

2. 4 sentences have suffixes on the possessum in addition to the possessive prefix.

3. 1 sentence has a possessum that does not have a lexical entry.

The third issue is a lexical coverage gap that does not have to do with adnominal possession inference, but the first two issues do point to issues with my inference system. The first issue is a consequence of the incorrectly-identified possessive strategy discussed earlier in this section; the grammar expects that the possessor has a prefix. I verified that if I give the possessor a prefix, the phrase is able to parse. The second issue seems to be that for all noun position classes I create as part of adnominal possession inference, I set the input of those position classes to "noun" (i.e. the lexical entry of a noun). Once the stem has gotten a suffix, it no longer counts as "noun" and is not able to be an input to the possession-related position classes. Future work might explore integrating the output of adnominal possession inference with existing position classes inferred by MOM; I expect that many sentences across languages would benefit from this fix.

4.2.5 Wambaya (held-out)

Wambaya (wmb) is a non-Pama-Nyungan language spoken in Australia. My description of its approach to adnominal possession is from Nordlinger 1998.

Wambaya expresses alienable possession when the possessor is overt primarily by requiring that the possessor (which precedes the possessum) is in the dative case, although occasionally it is in the genitive case instead. The genitive suffix has a different form for some kinship nouns, and also inflects to agree with the gender of the possessum.

Possessor pronouns are modifier-like separate words that precede the possessum and agree with it in gender, number, and case. Occasionally, the possessor pronoun is substituted by an oblique pronoun. These pronouns are used with alienable possessums.

Inalienable possession is expressed through juxtaposition. The examples in the reference grammar suggest (although it is not explicitly stated) that the possessor and possessum need not be immediate neighbors, and it appears that the word order may not be fixed. However, both entities are marked for the same case (Nordlinger, 1998). I am not sure whether there is currently a way to cleanly model this strategy in the adnominal possession library; when neither entity is marked for possession, the user can specify that the possessor have a certain case, but not that the possessor's case match that of the possessum. With that in mind, I consider this strategy out of scope for the current discussion.

Given this description of Wambaya's approach to adnominal possession, I would expect a grammar specification to contain the following information:

1. A possessive strategy in which neither entity is marked for possession, but the possessor is required to take the dative case. My inference system does not look for dative case currently, so this strategy cannot be inferred today.

2. A possessive strategy in which neither entry is marked for the possession, but the possessor is required to be in the genitive case and precede the possessum.

3. A set of modifier-like possessor pronouns that precede the possessum and agree with it in gender, number, and case. The inference system will not infer these currently, as I have not implemented inference for possessor pronouns that are separate words.

Results for Wambaya

The inferred grammar specification does not contain any information about adnominal possession. This is the case across all ten folds of Wambaya data. As such, my changes con-

tributed no new coverage or readings to the Wambaya test data, so I did not treebank or analyze any specific test sentences for this language. Instead, I investigated why no results were produced for Wambaya; the results of this investigation are in the next subsection.

Discussion of Wambaya results

Of the three approaches to possession that I described in my expectations for a Wambaya grammar specification, it is unsurprising that the first and third (i.e. the dative case marker and the possessor pronouns that are standalone words) are not inferred by my system, as I did not implement either a search for dative case nor for separate possessor pronouns. I initially expected that the second approach (genitive case marker on the possessor) would have been inferred, given that there were over a dozen examples of this kind of construction in the training data. However, I realized that my algorithm, in searching for evidence of genitive case on the possessor, does not split glosses into parts, such that a gloss that contains information other than the genitive case will not be caught. Example (20) is an example from the Wambaya training data in which the gloss for the genitive morpheme contains multiple grams (GEN for genitive case and IV for the noun's gender/noun class).

(20) Mirra ngi alangi-niganka-ni jalyu-nil
sit 1.SG.S.PR boy.I-GEN.IV-LOC bed.IV-LOC
"I'm sitting on the boy's bed." (Nordlinger, 1998)

Wambaya provides motivation for inferring feature constraints other than genitive case on the possessor, particularly dative. Future work that adds this inference, inference for standalone possessive pronouns, and splits complex glosses may hopefully be sufficient to add parses to adnominal possession sentences in the Wambaya test data.

4.2.6 Hiaki (held-out)

Hiaki is an Uto-Aztecan language spoken in Mexico and the United States in which possessors are marked with the genitive case and precede the possessum (Harley, 2013). It also has a class of standalone possessor pronouns that precede the possessum and that inflect for person and number. One of the two third-person singular possessor pronouns, *aa* appears to be accompanied by a non-inflecting possessive suffix on the possessum, while the other possessor pronouns accompany unmarked possessums (Sanchez et al., 2015).

I would expect a grammar specification for Hiaki to include:

1. a possessive strategy wherein the word order is head-final, neither entity is marked, and the possessum is required to be in genitive case,

2. a class of possessor pronouns that inflect for person and number, and

3. (possibly) a possessive strategy wherein both entities are marked, the possessor with the third-person pronoun *aa* and the possessum with the possessive suffix *wa*, neither of which carries agreement features.

Of these options, only the first is able to be inferred by the current system. The other two options rely on the as-yet unimplemented inference of standalone possessor pronouns.

Results for Hiaki

The inference system correctly infers the genitive-marked possessive strategy, as shown in the grammar specification excerpt (21). It also infers a set of possessor pronouns as affixes on the possessum. The orthography for these affixes, which include the = character, indicates that they may be analyzed as clitics rather than affixes (see Footnote 6 in Chapter 2 for more discussion on the treatment of clitics in this thesis). However, even if these possessor

pronouns are accidentally included in the output grammar specification, their inclusion will not necessarily lead to correct parse coverage; in particular, one of these pronouns is the third-person singular pronoun *aa*, which must also be accompanied by a suffix on the possessum. The grammar specification does not make reference to any such suffix.

(21)
```
section=adnom-poss
  poss-strat1_order=head-final
  poss-strat1_mod-spec=spec
  poss-strat1_mark-loc=neither
  poss-strat1_pronoun-allow=yes
    poss-strat1_feat1_name=case
    poss-strat1_feat1_value=gen
  poss-pron1_type=affix
  poss-pron1_agr=non-agree
  poss-pron1_mod-spec=spec
section=morphology
 noun-pc24_name=noun-pc24
 noun-pc24_order=suffix
 noun-pc24_inputs=noun
    noun-pc24_lrt1_name=noun-pc24_lrt1
      noun-pc24_lrt1_feat1_name=poss-pron1
      noun-pc24_lrt1_feat1_value=plus
      noun-pc24_lrt1_feat1_head=itself
      noun-pc24_lrt1_feat2_name=person
      noun-pc24_lrt1_feat2_value=2nd
      noun-pc24_lrt1_feat2_head=itself
      noun-pc24_lrt1_lri1_inflecting=yes
      noun-pc24_lrt1_lri1_orth=em=
    noun-pc24_lrt2_name=noun-pc24_lrt2
```

```
noun-pc24_lrt2_feat1_name=poss-pron1
noun-pc24_lrt2_feat1_value=plus
noun-pc24_lrt2_feat1_head=itself
noun-pc24_lrt2_feat2_name=person
noun-pc24_lrt2_feat2_value=3rd
noun-pc24_lrt2_feat2_head=itself
noun-pc24_lrt2_lri1_inflecting=yes
noun-pc24_lrt2_lri1_orth=aa=
```

Of the eight IGT with possession in the first fold of the Hiaki test dataset, none parsed using the inferred grammar. I examined all eight via the LKB and found that most of the possessive phrases were unable to parse, although one phrase unexpectedly unified. I discuss the reasons for the lack of unification as well as the unexpected unification in the next subsection.

Discussion of Hiaki results

Six of the eight test sentences contained a possessum that lacked lexical coverage. Two were unable to parse in general for reasons not having to do with possession, but were able to unify the possessor and possessum. For example, the grammar licenses the phrase in (22) under the possessive rule.[9] Note that the possessor in this example, *vem* is not glossed as having genitive case, but the possessive rule (which requires the possessor to be in genitive case) licenses the construction because the word is underspecified for case.

(22) vem familia
3PL.POSS family
"their family" (Harley, 2019)

[9]The pronoun "vem" was not inferred by my possession algorithm but instead by the lexical inference portion of the AGGREGATION inference program.

I noticed two other issues with the test sentences: first, that the parser did not recognize the morpheme *aa=*, so the possessive pronoun lexical rule that licenses possessums with this prefix did not apply; and second, that several of the possessums contained suffixes (often case endings), and, once, the *-wa* possessive suffix that appears on possessums possessed by the third-person pronoun *aa*. The inferred grammar does not support possessums that include other affixes; as with other results discussed in this chapter, it's possible that this problem could be improved by future work integrating the adnominal possession inference output with existing position classes in the grammar specification rather than creating new ones.

4.2.7 Summary

In this section, I have presented the results, discussion, and error analysis for each of my three development and three held-out languages.

4.3 Summary

Overall, I found that the system largely behaved as intended for the development languages, especially with respect to the grammar specification output. Several of the gains seen in the development results carry over to the held-out results as well, although in those cases there were more examples of failures to parse due to gaps in adnominal possession coverage. Some of these gaps (e.g. standalone possessor pronouns) were expected, while others (e.g. dative case marking on the possessor) were new information I had not considered in designing the system.

Chapter 5

CONCLUSION

In this thesis, I have presented a working algorithm for automatically inferring adnominal possession information — specifically possessive strategies for overt possessors, as well as possessor pronoun affixes — about a given language. I have shown that the grammars created from this system produce new correct readings for some sentences in one language (Abui) that were not possible prior to this work. Additionally, the system makes visible progress (as demonstrated by the correct grammar specification information as well as by my per-sentence analysis) towards increased validated coverage in several languages.

In analyzing these results, I identified a number of items that I think would be valuable to consider for future work. These items range from simple updates (for example, inferring dative case marking on the possessor by building on the current process for inferring genitive case, or tweaking the threshold used to determine whether a possessor is specifier-like or modifier-like) to more complicated work (such as integrating adnominal possession inference with the position classes inferred by MOM, adding inference for standalone possessor pronouns, or inferring agreement for features other than person). I believe each of these items would be worthwhile to explore and would likely significantly contribute to the coverage of the adnominal possession library. Integration with MOM might also reduce the ambiguity by removing duplicate position classes for orthographically identical affixes. Other assumptions and hard-coded decisions I made in designing this system are that the possessor can always be a pronoun (future work may consider inferring this rather than hard-coding it) and that possessor pronoun affixes are always labeled as not agreeing with the possessum. My analysis has also highlighted an opportunity for future work to re-examine morphological inference

to understand why in Tsova-Tush the genitive case marker is often not getting applied when it should, which in turn prevents possession rules from applying.

I would like to briefly mention two takeaways from this project that I hope can be valuable to future researchers exploring grammar inference, especially as part of the AGGREGATION project. First, it was surprisingly possible to infer accurate and detailed information from what often seemed to me like a very small amount of training data. I suspect this ease is largely due to the wealth of extra data added by INTENT (Georgi, 2016) to already rich IGT that display varied possessive constructions, as well as the tools within the AGGREGATION pipeline, such as the foundational work of BASIL (Howell, 2020) and MOM (Wax, 2014; Zamaraeva, 2016; Zamaraeva et al., 2017) and the clear structure of the adnominal possession library in the Grammar Matrix (Nielsen, 2018), all of which enabled me to target specific, well-defined values with my inference code. Second, I believe my analysis highlights the importance in AGGREGATION inference work of evaluating the grammar specification on its own, rather than only examining the performance of the resulting grammar. Not only does the grammar specification sometimes reflect the correct choices even if there are other reasons that sentences don't parse (suggesting that the sentences will begin parsing when those other parts of the system are fixed), but even a partially-complete set of inferred information in the grammar specification can — by allowing a linguist to edit a robust starting point rather than having to create the specification from scratch — help bootstrap the creation of machine-readable grammars.

BIBLIOGRAPHY

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation*, pages 23–72.

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

Emily M. Bender, Michael Wayne Goodman, Joshua Crowgey, and Fei Xia. 2013. Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. *Proceedings of the ACL 2013 workshop on Language Technology for Cultural Heritage, Social Sciences and Humanities*.

Emily M. Bender, Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. 2014. Learning grammar specifications from IGT: A case study of Chintang. *Proceedings of the Workshop on the Use of Computational Methods in the Study of Endangered Languages, ACL 2014*.

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

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

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

Berthold Crysmann and Woodley Packard. 2012. Towards efficient HPSG generation for German, a non-configurational language. *Proceedings of the 24th International Conference on Computational Linguistics*, pages 695–710.

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

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

Heidi Harley. 2013. Squib: Feature-matching and case and number dissociation in Hiaki. *Revista linguística*, 9(1):1–9.

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

Bryn Hauk. 2016-2019. Tsova-tush lexicon and texts. University of Hawai'i at Mānoa. Unpublished FieldWorks (FLEx) project. V2019.08.20.

Dee Ann Holisky and Rusudan Gagua. 1994. Tsova-tush (Batsbi). In Rieks Smeets, editor, *The Indigenous Languages of the Caucasus. Volume 4: The North East Caucasian Languages*. Caravan Books.

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

František Kratochvíl. 2007. *A grammar of Abui*. Utretch: LOT.

František Kratochvíl. 2019. Abui corpus. Electronic Database: Unpublished toolbox project (accessed March 2019).

Lev Michael. 2012. Possession in Nanti. In Alexandra Y. Aikhenvald and R. M. W. Dixon, editors, *Possession and Ownership*. Oxford University Press.

Lev Michael, Christine Beier, Zachary O'Hagan, Haroldo Vargas, and José Vargas. 2013. Matsigenka text corpus (version June 2013; FLEx database and LaTeX interlinear output).

Elizabeth Nielsen. 2018. Modeling adnominal possession in the LinGO Grammar Matrix. Master's thesis, University of Washington.

Rachel Nordlinger. 1998. *A Grammar of Wambaya, Northern Australia*. Pacific Linguistics.

Stephan Oepen and Dan Flickinger. 1998. Towards systematic grammar profiling. Test suite technology ten years after. *Journal of Computer Speech and Language*, 12(4):411–436.

Woodley Packard. 2015. Full forest treebanking. Master's thesis, University of Washington.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Chris Rogers. 2010. Fieldworks language explorer (FLEx). *Language Documentation & Conservation*.

Jose Sanchez, Alex Trueman, Maria Florez Leyva, Santos Leyva Alvarez, Mercedes Tubino Blanco, Hyun-Kyoung Jung, Louise St. Amour, and Heidi Harley. 2015. *An Introduction to Hiaki Grammar*. University of Arizona Press.

SIL International. 2015. Field linguist's toolbox. Lexicon and corpus management system with a parser and concordancer. http://www-01.sil.org/computing/toolbox/documentation.htm.

Nick Thieberger. 2006a. *A grammar of South Efate: An Oceanic language of Vanuatu*. University of Hawaii Press.

Nick Thieberger. 2006b. Dictionary and texts in South Efate. Digital collection managed by PARADISEC [Open Access]. (Accessed March 2019).

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

Olga Zamaraeva. 2016. Inferring morphotactics from interlinear glossed text: Combining clustering and precision grammars. *Proceedings of the 14th SIGMORPHON Workshop on Computatoinal Research in Phonetics, Phonology, and Morphology*.

Olga Zamaraeva, František Kratochvíl, Emily M. Bender, Fei Xia, and Kristen Howell. 2017. Computational support for finding word classes: A case study of Abui. *Proceedings of ComputEL-2: 2nd Workshop on Computational Methods for Endangered Langauges*.

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