Tracing and Reducing Lexical Ambiguity in Automatically Inferred Grammars

Elizabeth Conrad

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

Master of Science

University of Washington

2021

Committee:

Emily M. Bender

Fei Xia

Program Authorized to Offer Degree:
Linguistics

University of Washington

Abstract

Tracing and Reducing Lexical Ambiguity
in Automatically Inferred Grammars

Elizabeth Conrad

Chair of the Supervisory Committee:

While the automated creation of machine-readable grammars is a valuable resource for linguists who wish to work with these grammars for linguistic hypothesis testing, the complexity of developing a system capable of creating such grammars presents a number of obstacles. One obstacle faced by the system this work belongs to is the excessive amount of ambiguity that the output grammars license. Because the system takes input from a diverse collection of resources, the glossing practices between datasets can vary, making it necessary to employ generalized approaches to determining the syntactic function of certain glosses. Such generalized approaches expand the kinds of data that can be used, but at the risk of introducing spurious ambiguity. This study investigates linguistically informed ways to reduce the ambiguity of the inferred lexicons and morphological systems in these grammars. By decreasing the presence of non-inflecting lexical rules and imposing stricter requirements on which roots may be entered into the lexicon, the lexical and morphological ambiguity of the system was reduced without an overwhelming loss of coverage.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF LISTINGS

LIST OF TABLES

ACKNOWLEDGMENTS

The work of this thesis stands on an incredibly strong foundation of support. In the months it took to complete this thesis, my advisor Emily M. Bender supported this work intellectually by providing necessary academic background, situating me within a fantastic network of colleagues, and assisting in finding avenues to complete the work itself. More importantly than the academic support, however, was her continuous encouragement, without which I would have been completely lost.

In addition to her, there are a number of people I had contact with that also ensured the completion of this work. I am grateful to Fei Xia for agreeing to be a reader on this work and providing her feedback. I would like to thank Olga Zamaraeva for being an incredibly helpful resource when it came to navigating the technology used in this project. Additionally, I'd like to thank Kristen Howell for always being available to answer my questions about her work and for helping me springboard from where she left off to continue making improvements in the same system. Similarly, I must thank David Wax for conceiving and developing the original MOM system that this work expands. I also got to interact with the extremely welcoming community of researchers within DEPLH-IN, and in particular I want to express my gratitude to Woodley Packard for helping me when I encountered problems using ACE. Furthermore in terms of software, without the work of Ann Copestake on the LKB and Stephan Open on [incr tsdb()], this thesis would not have been possible.

It is also imperative to thank the linguists that provide data to this project and enable this work to continue, for this thesis I must specifically thank František Kratochvíl, Heidi Harley, Nick Thieberger, Shobhana Chelliah, Bryn Hauk, and Daniel Kauffman.

Every one of the people mentioned above played a key role in ensuring this work was

completed, and I am incredibly grateful for both the support of these wonderful people and the opportunity to get to work on a project such as this.

DEDICATION

to Rick & Elizabeth

Chapter 1

INTRODUCTION

The AGGREGATION project and prior research efforts that it builds on, such as the LinGO Grammar Matrix (Bender et al., 2002, 2010), the RiPLeS project (Xia et al., 2016), and the interlinear text enrichment toolkit known as INTENT (Georgi, 2016), have worked to ease the ability to produce machine-readable grammars. As a part of the AGGREGATION project, additional tools have been built to further work toward this end, such as the lexical and morphological inference system known as MOM (Wax, 2014) and the syntactic inference system called BASIL (Howell, 2020). The grammars produced by these tools can be used by field linguists for linguistic hypothesis testing (Bender et al., 2013; Howell, 2020) without extensive effort on the part of the linguists themselves. However, as precision grammars are highly complex artifacts that, when developed by hand, can take upwards of decades of person-years of work to develop, it is no simple task to engineer a system that can output such grammars programmatically.

One of the issues with the grammars constructed from the automatic inference performed by MOM and BASIL is that they license an excessive amount of ambiguity, sometimes finding thousands of parse trees for a simple sentence. The question this thesis attempts to answer is how can careful investigation of the lexicon and the morphological rules of an output grammar assist in finding ways to reduce the lexical and morphological ambiguity of said grammar?

Chapter 2 of this thesis provides the background necessary to understand the overall project this work is a part of and contextualize where this work sits in that project. Chapter 3 covers the evaluation metrics that will be used to test the efficacy of the improvements, and Chapter 4 describes those improvements, such as decreasing the presence of non-inflecting

lexical rules and imposing stricter requirements on when roots may be entered into the lexicon. Chapter 5 goes over the results of the experiments, which overall found that these improvements made beneficial reductions to ambiguity without excessive loss of coverage. Lastly, Chapter 6 provides some closing thoughts on this work and potential ways it can be expanded.

Chapter 2

BACKGROUND

The purpose of this chapter is to outline the pipeline of grammar inference that this project sits within and detail the role of this project in said pipeline. It also will describe particular pieces of software in this pipeline that were most pertinent to completing this project.

2.1 The AGGREGATION Project

The AGGREGATION project aims to simplify the production of machine-readable Head-Driven Phrase Structure (HPSG) grammars (Pollard and Sag, 1994) that can be used by field linguists for linguistic hypothesis testing (Bender et al., 2013; Howell, 2020). The stages of the pipeline can be broadly enumerated as follows:

1. Accept Interlinear Glossed Text (IGT) input in a given language
2. Infer the lexicon and a set of morphological rules from this input
3. Infer a set of syntactic properties from the same input
4. Using these sets, produce a machine-readable grammar
5. Use this grammar to parse sentences in the given language using a parsing software
6. Optionally perform treebanking to select acceptable parses produced by the grammar for given sentences

Each of these steps will be covered in more detail below. The pipeline is also illustrated in Figure 2.1.

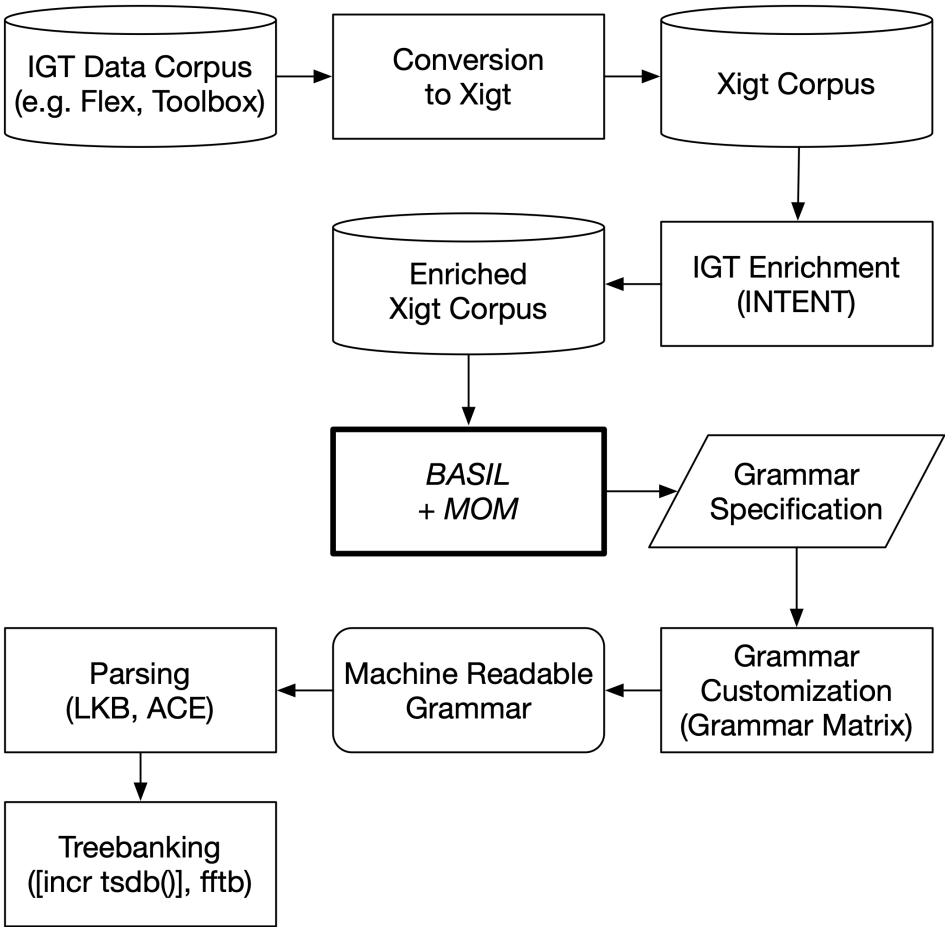


Figure 2.1: AGGREGATION Pipeline, recreated from (Howell, 2020, p. 16)

2.2 Interlinear Glossed Text

The initial input to the AGGREGATION pipeline is an IGT corpus, usually in Toolbox or FLeX format (Rogers, 2010; SIL International, 2015). An example line of IGT from Hiaki is shown below in (1).

(1) Juan vaaso-ta hamta-k
Juan glass-ACC break(tr.)-PERF
'Juan broke the glass' [yaq] (Harley, 2019)

The first line, called the language line, presents the text in the source language, and in this case is morpheme segmented. IGT input to the AGGREGATION pipeline must be morpheme segmented in order to enable morphological rule inference. If (1) were not morpheme segmented, and instead appeared as *Juan vaasota hamtak*, the system in its current state would be unable to deduce that *-ta* and *-k* are affixes. The second line, the gloss line, is aligned with the language line and contains a gloss for each morpheme in the language line. The last line is a translation line which translates the original sentence to a language of broader communication, in this case English. Because IGT is a format rich with linguistic information, is in wide use by documentary linguists, and is largely standardized via the Leipzig Glossing Rules (Bickel et al., 2008), it serves as an excellent source of data for computationally driven linguistic projects such as the AGGREGATION project.

While data in IGT format is standardized for human use, it is important to have the data in a format that is machine readable and promotes ease of storage and automated processing. This is the goal of the XML-based format called Xigt (Goodman et al., 2015). Therefore, all IGT data used in the AGGREGATION project is converted to Xigt. An additional benefit of Xigt is that it is an extensible format that easily allows for further enrichment of the data by adding more linguistic information that would not ordinarily be found in a linguist-provided IGT item. Such information can include part-of-speech (POS) tags and syntactic dependencies. The AGGREGATION project takes advantage of this and, after conversion to Xigt, the corpus gets enriched with POS tags and syntactic dependencies using INTENT

(Georgi, 2016). Because there exist more robust solutions to the problems of POS and dependency tagging in English than in low resource languages, INTENT uses the English translation of each sentence to deduce such information and projects this information onto the source language via the alignment present in an IGT item.[1] In (1), because an English POS tagger can tell that *break* is a verb, INTENT will note this, then figure out that it is aligned with *hamta* in the language line and thus the Xigt entry can be enriched with the information that *hamta* is a verb in Hiaki, providing an excellent starting point for lexical inference. The enriched Xigt corpus is then fed into MOM (section 2.4), which performs the lexical and morphological rule inference.

2.3 Grammar Specifications

The output of the morphological inference system (section 2.4) and the syntactic inference system (section 2.5) are combined into a single grammar specification that is then input into the LinGO Grammar Matrix (Bender et al., 2002, 2010) (section 2.6). This section explains what is included in these grammar specifications, as understanding the content is integral to understanding how it is populated.

The grammar specification contains information concerning lexical, morphological, and syntactic properties of the language. It is sometimes called a "choices file," as it reflects the "choices" made by linguist in filling out the web-based questionnaire for the LinGO Grammar Matrix. One goal of the AGGREGATION project is to automate this by deducing what "choices" would be selected by making inferences about the data. Because this thesis focuses on the lexical and morphological aspect of the AGGREGATION pipeline, the sections of the grammar specification concerning those will be explained in detail. As far as syntax goes, an example is shown in Listing 2.1 of some of the specifications relevant to syntax for an automatically generated grammar of Hiaki.

```
1 section=person
```

[1]Though sometimes the IGT corpora already have linguist-provided POS tags, and in that case these are used for their higher accuracy.

```
person=1-2-3
first-person=none

section=gender

section=case
case-marking=nom-acc
nom-acc-nom-case-name=nom
nom-acc-acc-case-name=acc
  case1_name=gen
  case2_name=all
  case3_name=ins
```

Listing 2.1: Syntax specification snippet

Each `section` pertains to a library of the Grammar Matrix which uses the information in a given section to constrain properties in the final output grammar. In Listing 2.1 the `section` dealing with `case` (Drellishak, 2009) specifies that Hiaki has a nominative-accusative case system, and also inferred three additional cases outside of that based on glosses found in the data: `gen`, `all`, and `ins`.

Below is a snippet of the `section` pertaining to the `lexicon`. The `lexicon` is composed of a number of inferred **word classes** (section 2.4.1).

```
section=lexicon
  noun7_name=noun7
    noun7_feat1_name=person
    noun7_feat1_value=3rd
  noun7_det=opt
    noun7_stem1_orth=aso'ola
    noun7_stem1_pred=_baby_n_rel
    noun7_stem2_orth=nooka
    noun7_stem2_pred=_speak_n_rel
    noun7_stem3_orth=yo'o
    noun7_stem3_pred=_adult_n_rel
```

```
    noun7_stem4_orth=yoemia
    noun7_stem4_pred=_family_n_rel

. . .

  noun121_name=noun121
  noun121_pron=on
    noun121_feat1_name=person
    noun121_feat1_value=1st
    noun121_feat2_name=number
    noun121_feat2_value=sg
    noun121_feat3_name=case
    noun121_feat3_value=nom
  noun121_det=opt
    noun121_stem1_orth=inepo
    noun121_stem1_pred=_pron_n_rel
```

Listing 2.2: Lexicon specification snippet

Listing 2.2 shows two sample noun classes shown: `noun7` and `noun121`. The class `noun7` specifies that the `person` feature on every noun in the class has the value `3rd` and that for these nouns determiners are optional (`noun7_det=opt`). Underneath all of the class-wide feature specifications is the list of the noun stems that are members of the class. Each stem has listed its orthography and predication value.[2] The structure of the noun class `noun121` is the same in that properties that apply to every noun are listed first followed by member stems, however for this class the actual features that are specified are different. This is a class for 1st person singular nominative pronouns, which in this language applies to only one stem. There also exist analogous word classes for verbs with features related to verbs, such as argument structure and case frame, being specified for each class.

[2]For the convenience of the contributors to the AGGREGATION project, these predication values follow a convention of using English lemmas, but this is not intended to suggest that this is a "correct" way of representing lexical semantics, as English lemmas likely will not have a one-to-one relationship with lemmas of other languages.

The next `section` of interest in the choices file is called `morphology` (Goodman, 2013), which is composed of a number of inferred **position classes**. This will be covered in more detail in section 2.4.2, but for now think of a position class as being a "slot" that may be filled by an affix. An example of a verb position class is shown in Listing 2.3.

```
1 verb-pc10_name=verb-pc10
2 verb-pc10_order=suffix
3 verb-pc10_inputs=verb-pc4, verb-pc8, verb-pc11, verb-pc14, verb-pc15, verb
     -pc20, verb20, verb-pc19, verb-pc27, verb-pc31, verb-pc33, verb-pc29,
     verb-pc48, verb-pc49, verb131, verb220
4     verb-pc10_lrt1_name=verb-pc10_lrt1
5       verb-pc10_lrt1_feat1_name=aspect
6       verb-pc10_lrt1_feat1_value=pfv
7       verb-pc10_lrt1_feat1_head=verb
8       verb-pc10_lrt1_lri1_inflecting=yes
9       verb-pc10_lrt1_lri1_orth=-k
```

Listing 2.3: Verb position class example

The first thing specified in a position class after its name is the `order` of the position class, that is whether affixes belonging to that position class are prefixes or suffixes. The next thing specified in a position class is its `inputs`. In the code itself, the `inputs` appear as a list of other position classes and/or word classes. This means that any **lexical structure** resulting from these position and/or word classes is eligible to take any affix in this position class. The term **lexical structure** is used here to refer to any feature structure that may be the daughter of a lexical rule instance in HPSG, which can either be a feature structure from a **lexical entry** or the feature structure corresponding to the OUTPUT of a **lexical rule instance**. In this implementation, a lexical entry is represented as a member of a word class, and the OUTPUT of a lexical rule instance is simply the programmatic output of a lexical rule instance being applied from a particular position class. So in Listing 2.3, any verb stem that is a member of either verb class `verb131` or `verb220` can take an affix belonging to position class `verb-pc10`, and the resulting feature structure would be eligible to take affixes which

have `verb-pc10` in their `inputs` list. Similarly, the result of the application of any lexical rule instance from the position classes listed in the `inputs` list would also be eligible to take an affix from `verb-pc10` (e.g. the resulting lexical structure from a lexical rule instance being applied from `verb-pc4` could take an affix from `verb-pc10`). Figure 2.2 illustrates how the `inputs` of a position class result in affixes being concatenated to a stem in a particular order.

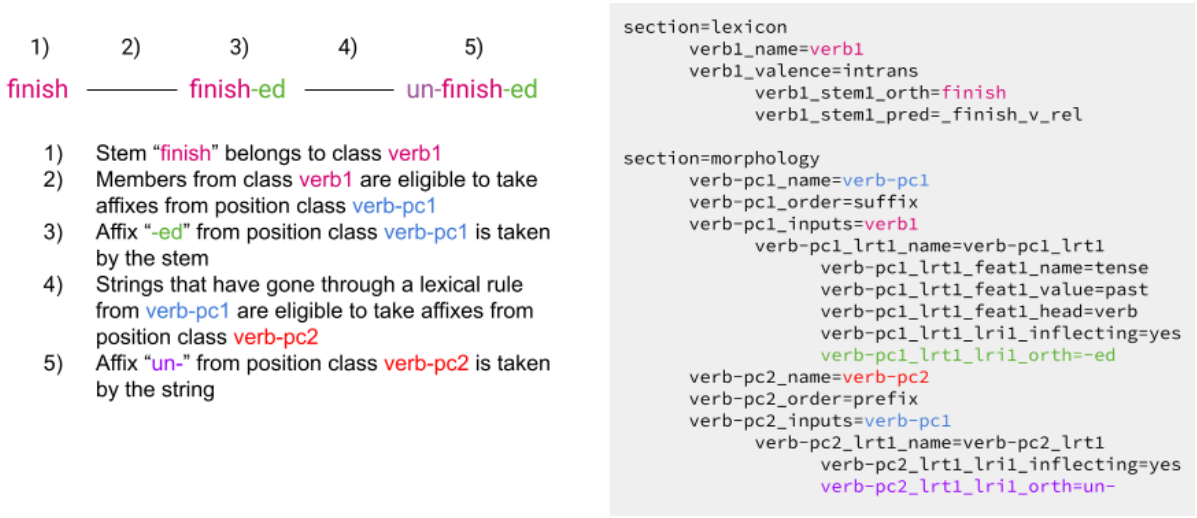


Figure 2.2: Diagram illustrating how a position class' inputs determine affix concatenation

After the `inputs`, a position class will have a list of **lexical rule types** followed by features that are defined by that lexical rule type. In Listing 2.3 the only lexical rule type in `verb-pc10` specifies that strings going through lexical rule instances of this type will carry a value of `pfv` for the feature `aspect`. Then, for every lexical rule type, there is a list of **lexical rule instances**. In the current example, there is one lexical rule instance which is inflecting and is realized as a suffix with the orthography *-k*. It is possible for there to be more than one affix that would be associated with the same feature value pairs, and in that case there would be an additional lexical rule instance with the appropriate orthography in the same lexical rule type. It is also possible for features to be realized without an explicit affix, these are called zero morpheme rules, and in that case there would be a lexical rule

instance where `inflecting` was marked as `no` and there would be no associated orthography. Now that the relevant content of the grammar specifications has been explained, the next section will cover how this information is populated into the specification.

2.4 MOM

The Matrix-Odin Morphology system (Wax, 2014) is the piece of the pipeline that infers the lexicon and a set of morphological rules for the output grammar. It is named such due to its association with the LinGO Grammar Matrix (Bender et al., 2002, 2010) and because upon its conception much of the data used for development and testing was retrieved from the Online Database for Interlinear Glossed Text (ODIN) (Lewis and Xia, 2010).

When inferring the lexicon, the system groups words into a number of word classes, as described in Section 2.4.1. Along with this, it infers what affixes exist in the language and groups them according to position classes, as described in Section 2.4.2.

2.4.1 Word classes

When inferring the lexicon, MOM will use the enriched IGT information to group the lexicon into parts of speech. Beyond that, it will also group words within each POS[3] into classes. The linguistic motivation behind this is that different classes of words will take affixes specific to that class. One example of this is Spanish verbs. Verbs in Spanish are routinely conceptualized as belonging to one of three groups: *-ar* verbs, *-er* verbs, and *-ir* verbs, as indicated by the last two characters of the verb in its infinitive form. The affixes that *-ar* verbs take for particular realizations of tense, aspect, and mood are distinct from those of the *-er* verbs, and also distinct from those of the *-ir* verbs.

When MOM begins grouping words in a particular POS into classes, say verbs, it will first determine the root of the verb and what features are specified on this root (Zamaraeva et al., 2019). For example, the English word *ran* would be recognized as a verbal root with

[3]Currently, the system only creates classes for nouns and verbs

a value of PAST for TENSE and an intransitive valence frame, as opposed to *run* which would simply be recognized as intransitive with no specification on the TENSE feature (unless it was specifically glossed as 'run.PRES').

In the initial pass-through of the data, for every eligible word (i.e. verb or noun) MOM will determine the root and feature value pairs for that word and place it in its own word class (Zamaraeva et al., 2019). These are later compressed into more general classes containing many stems, as explained in section 2.4.3.

2.4.2 Position classes

The position class conceptualization of morphology, as implemented in the Grammar Matrix morphotactics library in Goodman 2013, is the foundation of the MOM system. A position class can be thought of as a "slot" that an affix may fill. Each "slot" can be filled by any affix from a given set, but it may only be filled by one affix from the set in any particular realization of a word. Finnish is a language with complex inflectional morphology; nouns and adjectives in the language have four position classes that may be filled following the root (Karlsson, 2017). The order and general meaning of these position classes can be expressed as follows: ROOT-NUMBER-CASE-POSSESSION-CLITIC, where the NUMBER slot may be filled by a plural marker (singular is unmarked), the CASE slot may be filled by one of the numerous case markers, the POSSESSION slot may be filled by a suffix that indicates number and person of a possessor, and lastly the CLITIC slot may be filled by one of a handful of suffixes that contribute additional meaning such as the suffix *-kin* which means 'also' or 'too' (Karlsson, 2017). The example below shows a Finnish word with one suffix filling each of the four position classes:

(2) pullo-i-ssa-nne-kin
bottle-PL-INESS-POSS.2PL-also
'in your bottles too' [fin] (Karlsson, 2017)

MOM takes advantage of this notion of position classes to infer morphological rules in a language. When processing affixes on a given root, MOM starts with prefixes, then moves

onto suffixes; both sets are processed starting with the one closest to the root first and then moving outward (Wax, 2014). Initially, every affix is given its own position class, these are later compressed as explained in the following section.

2.4.3 Graph Compression

The morphology produced by MOM can be conceptualized as a graph, where word classes and position classes are the nodes and the inputs are directed edges. A node's outgoing edges point toward position classes that may accept lexical structures resulting from that node, and conversely, a node's incoming edges represent the nodes that position class may accept lexical structures from.

After the first pass-through of the data, there is a word class for every unique stem and a position class for every unique affix. In order to achieve more general classes, merging operations are performed on both the word and position classes.

First, the position classes are merged. The general idea is that if two position classes share a certain percentage of incoming edges from the same nodes, they are good candidates for merging (Wax, 2014). So in the most minimal example, if two position classes each have one incoming edge that comes from the same node, then instead of being two separate classes with one lexical rule instance each, they can be merged into one class with two lexical rule instances. This is shown in Figure 2.3.

In the figure, what was originally `verb-pc1` and `verb-pc2` are merged, and the lexical rule type from `verb-pc2` is added to `verb-pc1`, so now any stem belonging to the word class `verb1` may have a lexical rule instance from the new `verb-pc1` applied to it, and this is a more sensible analysis of the way suffixes work in English, as both of these fill the same "slot" following a root, and only one may appear in any word. Furthermore, if a lexeme takes one affix from a class, it most likely can take the others. (The primary exception to this is irregular forms; however MOM targets a morpheme-segmented representation, which usually has already abstracted away from morphophonological processes.)

The decision of whether to merge any two classes is based on a parameter called the

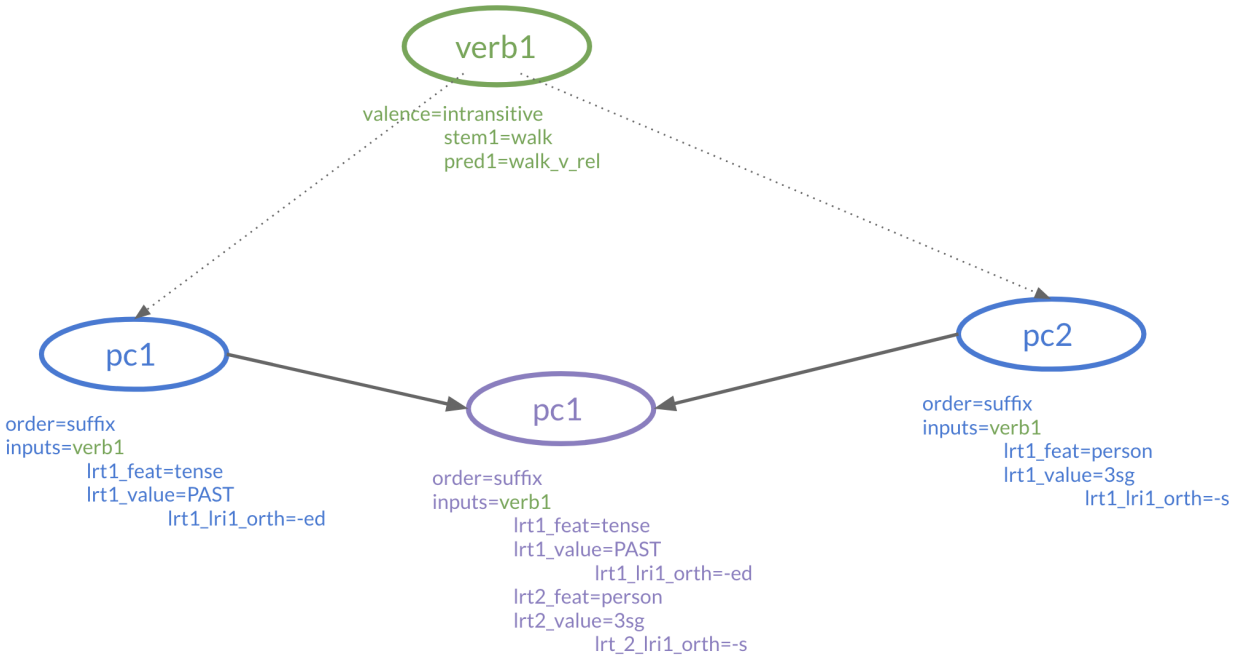


Figure 2.3: Example of position class merging

minimum overlap. If the percentage of shared incoming edges (i.e. members from their `inputs` list) from two position classes exceeds the minimum overlap parameter, then the two classes are merged (Wax, 2014).

After merging all combinations of position classes that have sufficient overlap, an analogous merge operation is performed on the word classes, except that rather than merging based on overlap of *incoming edges* the merge is performed on *outgoing edges* (Zamaraeva et al., 2017). In the case of word class nodes, there is no list of `inputs`, so these nodes never have incoming edges. However, the system keeps track of their outgoing edges and performs the merge on these, the idea being that if two word classes share a significant amount of outgoing edges, then stems from these word classes are found to take the same affixes and therefore can be assumed to be one class (Zamaraeva et al., 2017). This merging is performed with the same restriction on overlap as the position class merge operation and continues until all combinations with sufficient overlap have been merged.

Once this merging operation is completed, another merge operation is performed again on the position classes, since with reduced word classes there are likely more combinations of position classes that exceeded the minimum overlap parameter. In the default system, the merging stops at this point, but there is a compression round parameter that the user may enter into their configuration file that will perform more rounds of compression (where one round is one merge operation of word classes followed by one merge operation on position classes) until the number of rounds specified has finished.

2.5 BASIL

The next component of the AGGREGATION pipeline is BASIL (Howell, 2020), which performs syntactic inference on the provided IGT data. BASIL greatly reduces the manual work needed to get a working machine-readable grammar, as it automatically infers many of the specifications that previously would need to be filled out by a linguist using the LinGO Grammar Matrix web-based questionnaire. As reducing syntactic ambiguity is not the focus of this thesis, the details of how BASIL performs this inference will not be covered, but some of the phenomena that BASIL can infer include how the language realizes person, tense, word order, case, and coordination, among others (Howell, 2020).

2.6 LinGO Grammar Matrix

Once morphological and syntactic inference has been performed, it is the job of the LinGO Grammar Matrix (Bender et al., 2002, 2010) to use the inferred grammar specification to construct a machine-readable HPSG grammar. The Grammar Matrix makes use of a number of libraries developed specifically for the Matrix that contain information about various ways a particular phenomenon is realized across many languages. For example, the library that handles case (Drellishak, 2009) will output certain constraints in a grammar for a specification that states that the language has a nominative-accusative case system and different constraints if the language is stated to have an ergative-absolutive case system. Currently, there are libraries for adnominal possession (Nielsen and Bender, 2018), argument

optionality (Saleem, 2010), case (Drellishak, 2009), coordination (Drellishak and Bender, 2005), TAM (tense, aspect, and mood; Poulson, 2011), word order (Fokkens, 2010), and others.

2.7 Parsing software

Once a grammar is output from the grammar matrix, it can be used in tandem with a parsing software to parse test sentences. The two pieces of parsing software that were used during this thesis were the LKB (Linguistic Knowledge Building) system (Copestake, 2002) and ACE (Answer Constraint Engine) (Crysmann and Packard, 2012). These pieces of software can accept a sentence in the relevant language as input and produce readings for that sentence in the form of parse trees using the constraints from the grammar being tested. They also produce semantic representations of the readings in the form of Minimal Recursion Semantics (MRS; Copestake et al., 2005). Figure 2.4 shows a sample parse tree from the LKB for the sentence in (3). Figure 2.5 shows the MRS for the same tree.

(3) Juan vaaso-ta hamta-k
Juan glass-ACC break(tr.)-PERF
'Juan broke the glass' [yaq] (Harley, 2019)

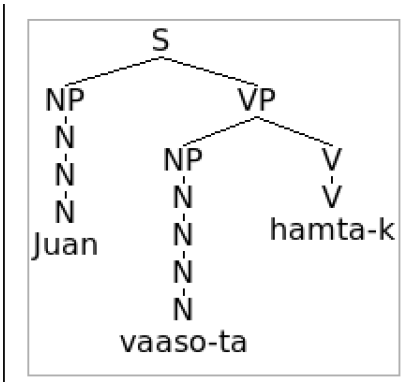


Figure 2.4: Parse tree of Hiaki (yaq) sentence in (3)

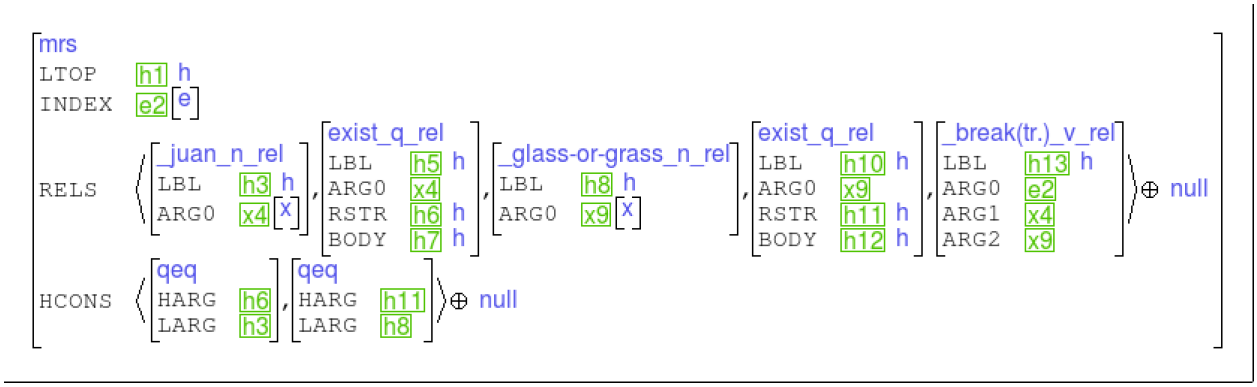


Figure 2.5: MRS of Hiaki (yaq) sentence in (3)

2.8 Managing Test Suites

In many cases, it is more efficient to work with large collections of test sentences, rather than inputting one sentence at a time to a parsing software. For this thesis, two pieces of software were used for this purpose: [incr tsdb()] (Oepen, 2001) and ART[4]. Both pieces of software can find readings for every sentence in a test suite and save the results.

2.9 Treebanking

Optionally, it is possible to perform treebanking on the analyses output by a particular grammar. When treebanking a sentence using a treebanking software (Oepen et al., 2004), all of the parse trees that the grammar found for that sentence are shown along with the minimal discriminants (Carter, 1997) (in the form of phrase structure rules, lexical rules, and lexical entries) between those trees. The user then will make decisions about which rules and entries seem the most appropriate and narrow down the analyses to one final "correct" analysis. To confirm that the analysis is correct, the user would look at the MRS shown for that particular analysis and ensure that it is sensible for the given sentence. For this thesis, no new treebanking work was performed, however part of the evaluation of results was done

[4] `http://sweaglesw.org/linguistics/libtsdb/art.html`

by making use of data that was treebanked in Howell 2020.

2.10 Summary

This chapter covered the structure and relevant software of the AGGREGATION project, paying special attention to those pieces which pertain to morphological inference, as that is the most pertinent to this thesis. The following chapter will explain the metrics for evaluating the results of the work, doing which will provide better understanding for just what the work is intending to improve.

Chapter 3
EVALUATION METHODOLOGY

This chapter will explain the evaluation metrics used to test the effectiveness of changes made in the AGGREGATION pipeline that aimed to reduce morphological and lexical ambiguity. These metrics were devised prior to the actual ambiguity reduction work in order to ascribe concrete meaning to the concept of "reducing ambiguity" before determining how this reduction could be achieved. Four evaluation metrics were used: (1) average number of readings per sentence, (2) coverage of the output grammar, (3) persistence of treebanked readings, and (4) "good loss" of coverage. After discussing the metrics used, the structure of the training and evaluation data will be explained.

3.1 Defining Ambiguity

A sentence is considered ambiguous according to a grammar if the grammar licenses more than one structure for said sentence. It is entirely possible for a sentence to exhibit genuine ambiguity, such as in the sentence "The kid saw the astronomer with the telescope." The prepositional phrase *with the telescope* could either modify the verb (i.e. the kid is using the telescope to see the astronomer), or the astronomer (i.e. the astronomer has a telescope). However, one of the current issues with automatically generated grammars whose constraints are inferred by MOM and BASIL is that they permit an excessive level of ambiguity that is not indicative of true ambiguity. The hypothesis I'm investigating in this thesis is that one of the major sources of excess ambiguity in the grammars produced in the AGGREGATION pipeline is due to the way lexical and morphological inference is performed. With this as a starting point, four evaluation metrics were devised to help determine if making changes in the pipeline did in fact reduce the ambiguity that these grammars purported. Table 3.1

Metric	Value
Number of sentences in test suite	1568
(1) Average readings per sentence	919.07
(2) Coverage by count	657/1568
(2) Coverage by %	41.90%
(3) Treebanked sentences with valid analysis	12
(4) Treebanked sentences with only invalid analyses	52
(4) 'Good loss'	–

Table 3.1: Baseline evaluation statistics for Abui (abz)

shows the values for these metrics for Abui in the baseline run of the system. The numbers to the left of the row titles indicate which metric they are associated with.

3.2 Evaluation Metric 1: average number of readings per sentence

The first metric is to see if a given change reduced the average number of readings per sentence in a language. So very simply, if, after making a change to the pipeline, the average number of readings per sentence decreases then the goal of reducing ambiguity was achieved. In Table 3.1, the baseline ambiguity is 919.07 readings per sentence, so any number lower than this after a change would improve this metric. However, it is important to keep in mind the potential loss of coverage that occurs when readings are eliminated.

3.3 Evaluation Metric 2: coverage

Coverage is a measure of how many sentences in a corpus a grammar is able to provide an analysis for, and if a particular change comes with a moderate reduction in ambiguity but a large loss in coverage, then the change may have done more harm than good. It is possible for there to be a loss of coverage that was helpful though, and this is what the fourth metric

measures, as discussed in Section 3.5. Looking at these metrics in tandem will provide a clearer picture of what a loss in coverage represents.

3.4 Evaluation Metric 3: persistence of treebanked readings

The third and fourth evaluation metrics provide more robust measures of the benefit of an implemented change, and are calculated with respect to available treebanked data. Treebanking a sentence involves sifting through candidate readings for that sentence and selecting one whose semantic representation, syntactic rules, and lexical rules are sensible (see section 2.9). In this way, when treebanking a sentence, a "correct" reading is chosen. In the development of BASIL, Howell (2020) performed treebanking on a selection of sentences for many of the languages she worked with. This existing treebanked data is the same data I use in my thesis for the third and fourth evaluation metrics.

The third metric is a measurement of the persistence of readings of treebanked sentences after a change is made. It is a calculation of how many treebanked sentences that had a valid reading in the baseline system still have a reading whose MRS matches the MRS of the exemplar reading[1] in the baseline after a change, even if the exact syntactic structures between the readings differ. The ideal would be that not a single valid reading is lost, thus showing that only invalid readings were eliminated, while keeping valid ones.

3.5 Evaluation Metric 4: "good loss" of coverage

The fourth evaluation metric is a calculation of how many treebanked sentences that had no valid reading in the baseline system do not have any readings at all in the new system. If a treebanked sentence had 10 possible analyses in the baseline, but none of them were selected to be correct, then losing coverage on this sentence would actually demonstrate a loss of coverage that was beneficial. This metric is represented in Table 3.1 across two rows.

[1]Here, finding a matching MRS means finding an MRS that is isomorphic to the exemplar. Readings may have isomorphic MRSes even if the ordering of the elements in the RELS list or the particular variable names differ between the readings. Whether two readings were isomorphic was tested using a function in PyDelphin (Goodman, 2019).

The first relevant row, "treebanked sentences with only invalid readings," is a count of the number of treebanked sentences that were determined to have no legitimate readings during treebanking, but had parses in the baseline system, in this case 52. The second relevant row, "Good loss" has no value in the baseline, but if one of the 52 sentences with only invalid readings is not covered by any analysis after a change is made, then the "Good loss" row would have a value of 1.

3.6 Training and Testing Data

In the development of her dissertation, Howell organized all of the data she used into folds for training and testing. There were 10 training and testing folds created for each language used in Howell 2020. In any given fold, 90% of the available corpus data would serve as the training data and the remaining 10% would serve as test data.

Say for a given language there are 1000 available sentences in the corpus. This would mean that for the first fold sentences 1-100 (10% of the data) could be set aside to test the resulting output grammar, and the remaining 900 sentences (90% of the data) would serve as the training input to the system. Then, for the second fold, sentences 101-200 could be set aside as testing data, while sentences 1-100 plus sentences 201-1000 would make up the 90% of the data serving as training input. This sort of division would continue until 10 unique folds were created such that the union of all of the testing sets is the full original corpus. By dividing the data in this way, it effectively allows the system to be trained and tested 10 separate times, producing 10 output grammars, despite having a limited amount of data.

When running my experiments, I used three development languages: Abui [abz] (Kratochvíl, 2019), South Efate [erk] (Thieberger, 2006), and Hiaki [yaq] (Harley, 2019). Abui is an Alor-Pantar language in the Trans-New Guinea family, South Efate is a language of Central Vanuatu in the Austronesian family, and Hiaki is a Cahitan language in the Uto-Aztecan family (Hammarström et al., 2020). I also used three test languages, whose data was not viewed during development at all: Tsova-Tush [bbl] (Hauk, 2016–2019), Meitei [mni] (Chelliah, 2019), and Wakhi [wbl] (Kaufman et al., 2020). Tsova-Tush is a Nakh language in

the Nakh-Daghestanian language family, Meitei is a Kuki-Chin-Naga language in the Sino-Tibetan language family, and Wakhi is an Iranian language in the Indo-European language family (Hammarström et al., 2020). Within the AGGREGATION project, experiments are routinely run on test *languages*, as opposed to just test data within the same development language. This helps to test whether particular changes are beneficial cross-linguistically, since the AGGREGATION project aims to aid in automatic grammar production for a wide variety of languages. When running the experiments on both the development and test langauges, I used the same folds that were already created in Howell 2020. Note that tree-banking was only performed on one or a few of the ten folds in some of the datasets. In results reporting, the values for each metric will be cumulative across the 10 folds for the language being reported.

3.7 Summary

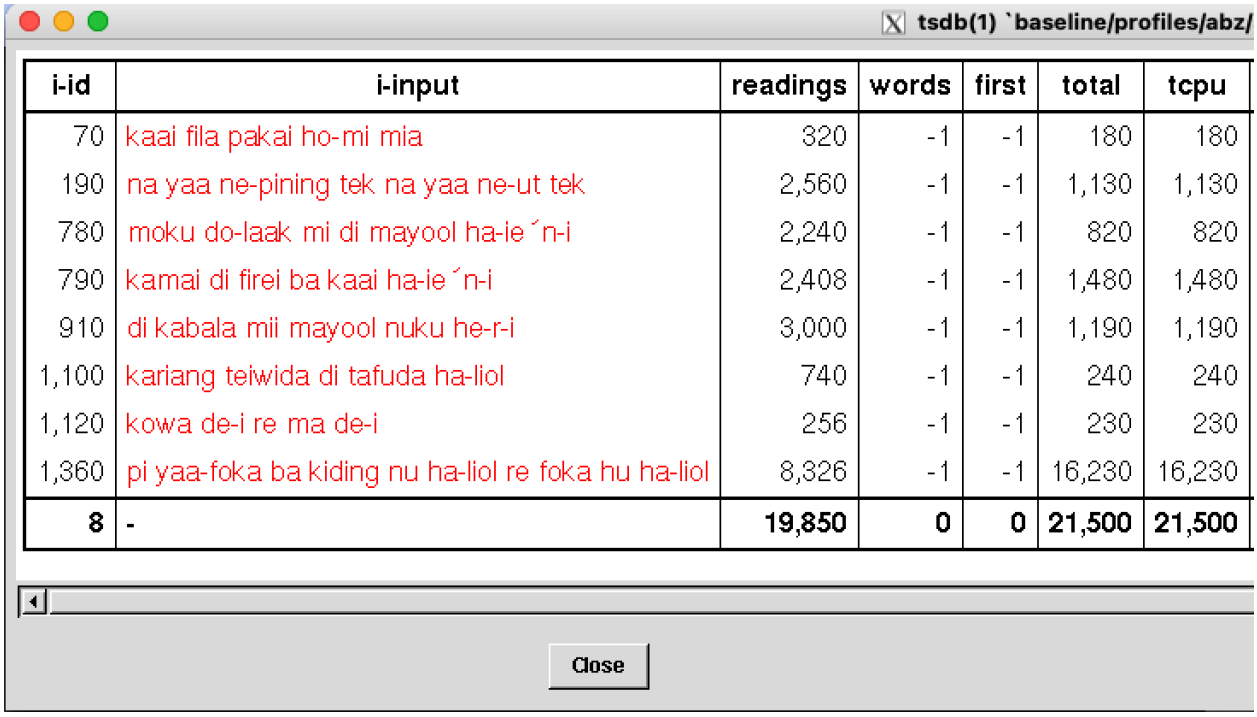
This chapter covered the four metrics that will be used to evaluate the results of the ambiguity reduction work. It also covered how the training and testing data is structured. The next chapter will explain the actual ambiguity reduction work that was implemented.

Chapter 4
IMPLEMENTATION

This chapter will go over the changes I made to attempt to reduce lexical and morphological ambiguity. It will also cover how I selected them as potential avenues for ambiguity reduction. In order to investigate potential sources of morphological ambiguity, three development languages were used: Abui [abz] (Kratochvíl, 2019), South Efate [erk] (Thieberger, 2006), and Hiaki [yaq] (Harley, 2019).

4.1 Finding Ambiguity Culprits

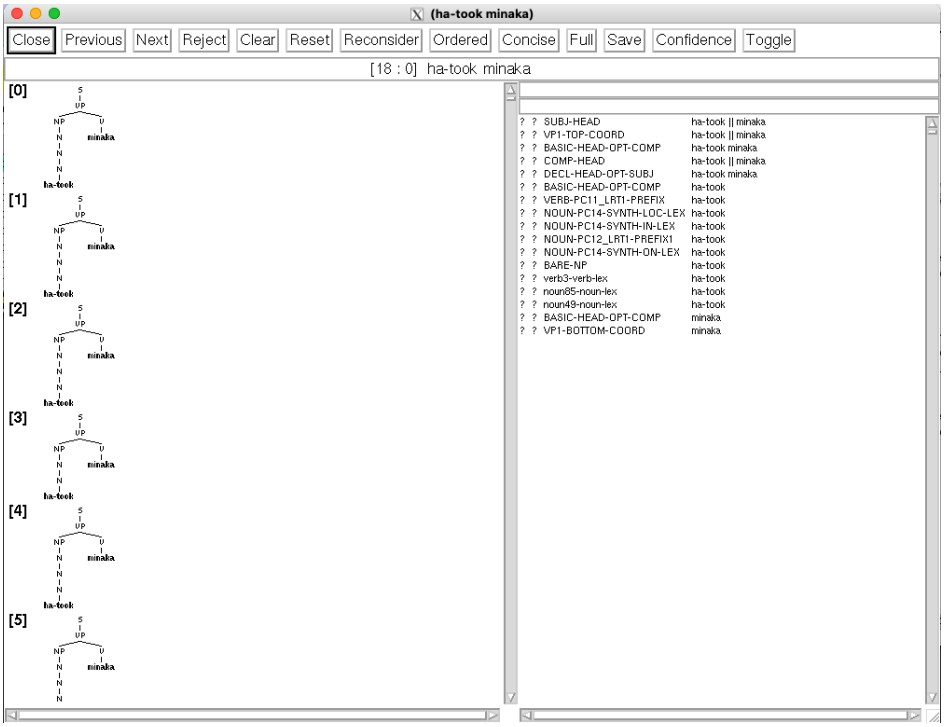
Once baseline runs of the system for each of the three development languages were completed, it was time to look for sources of lexical and morphological ambiguity. This was primarily done using the LKB (Copestake, 2002) and [incr tsdb()] (Oepen, 2001) in tandem. Figure 4.1 shows Abui sentences that had over 100 readings in the baseline. I looked at highly ambiguous sentences such as these in order to pinpoint and subsequently tackle sources of lexical and morphological ambiguity to improve the system. In the LKB, there is an option to compare different parse trees for a particular sentence. An example of this for an Abui sentence with 18 readings is shown in Figure 4.2. Figure 4.2a shows what the interface looks like initially. The left panel shows the parse trees that have been found for this sentence, and the right shows which rules are responsible for the ambiguity. Figure 4.2b shows an example of the result after rules have been selected in succession until only one tree remained (though the rules in this example were selected at random for illustrative purposes and are not necessarily part of a correct analysis of the sentence). Because the right panel of this interface shows specific rules that account for the ambiguity the grammar licenses for the sentence, it is an excellent tool for uncovering culprits that are causing lexical and morphological ambiguity.



i-id	i-input	readings	words	first	total	tcpu
70	kaai fila pakai ho-mi mia	320	-1	-1	180	180
190	na yaa ne-pining tek na yaa ne-ut tek	2,560	-1	-1	1,130	1,130
780	moku do-laak mi di mayool ha-ie´n-i	2,240	-1	-1	820	820
790	kamai di firei ba kaai ha-ie´n-i	2,408	-1	-1	1,480	1,480
910	di kabala mii mayool nuku he-r-i	3,000	-1	-1	1,190	1,190
1,100	kariang teiwida di tafuda ha-liol	740	-1	-1	240	240
1,120	kowa de-i re ma de-i	256	-1	-1	230	230
1,360	pi yaa-foka ba kiding nu ha-liol re foka hu ha-liol	8,326	-1	-1	16,230	16,230
8	**-**	**19,850**	**0**	**0**	**21,500**	**21,500**

Figure 4.1: Abui (abz) test profile screenshot showing sentences with over 100 readings in the baseline

Figure 4.3 shows a closer look at the rules in this sentence that account for this type of ambiguity for the token *ha-took*. One thing that was present, not only in this sentence but in many others in Abui as well, was what are from here on called "synthetic rules." That is, the rules that contain the string `-SYNTH-` in their name. The existence of these rules is the first ambiguity culprit to be tackled, which is covered in section 4.2. Another issue that can be seen in this example is that the token is being tagged as both a verb and a noun, which is shown by the three options for lexical entries: `verb3-verb-lex`, `noun85-noun-lex`, and `noun49-noun-lex`. This sort of part of speech tagging issue is the second ambiguity culprit tackled, covered in section 4.3.



(a) Parse compare before making selections

(b) Parse compare after narrowing down the options to one tree

Figure 4.2: Parse compare interface

```
? ? VERB-PC11_LRT1-PREFIX     ha-took
? ? NOUN-PC14-SYNTH-LOC-LEX ha-took
? ? NOUN-PC14-SYNTH-IN-LEX   ha-took
? ? NOUN-PC12_LRT1-PREFIX1   ha-took
? ? NOUN-PC14-SYNTH-ON-LEX  ha-took
? ? BARE-NP                  ha-took
? ? verb3-verb-lex           ha-took
? ? noun85-noun-lex          ha-took
? ? noun49-noun-lex          ha-took
```

Figure 4.3: Closer look at rules contributing to morphological and lexical ambiguity in an Abui (abz) sentence

4.2 Spurious Adpositional Case Rules

The synthetic rules causing ambiguity in this sentence and many others exist as a result of the implementation of the case library in Drellishak (2009). Per the implementation of the library, nouns in languages that mark case morphologically via an affix will have a mandatory lexical rule applied to them which marks them for the appropriate CASE value. The OUTPUT of these lexical rules is therefore a noun which is marked appropriately for case, and verbs may take NP arguments whose case unifies with the arguments that verb is expecting.

Alternatively, languages which mark case adpositionally achieve this by having case-marking adpositions as lexical entries. These adpositions take nominal complements and mark them for the appropriate case, and the resulting structure is an adpositional phrase that is also marked with the appropriate case. Verbs then take adpositional phrases as arguments, and this prevents noun phrases that are not marked for case from being arguments.

In addition to these two marking strategies, in some of the world's languages, case may be marked either morphologically or adpositionally. This is referred to as "mixed marking" in Drellishak 2009. Take for example a language with mixed marking that uses an affix to mark nominative and an adposition to mark nouns or noun phrases as accusative. For the nominative, a mandatory lexical rule must be applied to the noun, just as in a language

with pure morphological case marking. Then for the accusative, the lexical entry of the accusative case-marking adposition must take a non-case-marked NP as its complement, but because of the existence of the mandatory nominative lexical rule, no lexical entry that is a noun can serve as the head of a bare non-case-marked NP. Therefore, in the case library implementation, the solution is to "synthesize an additional, non-spelling changing lexical rule that applies to nouns and marks the appropriate value of case," (Drellishak, 2009). Hence, the existence of the synthetic rules. These rules sit in the same position class alongside the mandatory case-marking lexical rules and mark a lexical entry that is a noun as having the CASE value that a case-marking adposition would require. So in the nominative/accusative example, in addition to a lexical rule which makes a spelling change on a noun to mark the nominative morphologically, there would also be a non-inflecting rule that marks the noun as having CASE accusative, the result being an NP whose CASE value may unify with the accusative adposition, allowing it to be taken as the adposition's argument.

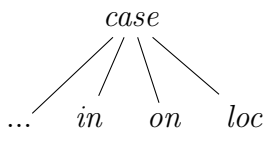
When it was implemented, there was one synthetic rule created for each adpositional case marker. This is shown in Figure 4.3, which shows that for `NOUN-PC14` there are three options for synthetic non-spelling changing lexical rules that may apply to this token: `NOUN-PC14-SYNTH-LOC-LEX`, `NOUN-PC14-SYNTH-IN-LEX`, and `NOUN-PC14-SYNTH-ON-LEX`. This signals that the system inferred three possible case-marking adpositions, and therefore it must be possible for a noun to be marked as having any of these three cases in the event that an adposition is to take that noun as an argument. However, this creates a large amount of ambiguity in such mixed marking languages, because any sentence that has a reading that makes use of one of those synthetic rules must also necessarily have readings that make use of the others as well.[1] For a mixed marking language, the number of readings of sentences

[1]Additionally, it appears there may be a bug that is causing these rules to be applied to tokens even when there is no case-marking adposition present. If this bug were fixed, presumably a sentence could not have analyses with the wrong synthetic rule, as only the one matching the present adposition could apply, but it is still worth handling this ambiguity because it reduces the number of edges in the parse chart since the system would not need to test whether each rule unifies.

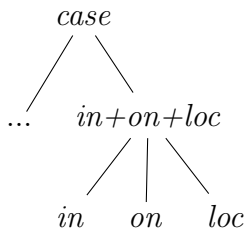
with analyses making use of these synthetic rules will be multiplied by a factor of the number of adpositional markers in the language.

In order to combat this, these three rules are replaced by one synthetic rule that does the same thing, except instead of marking a noun that has not been morpholigically marked for case with one of the adpositional case values, it marks such nouns as having a case value that is underspecified between all the adpositional case values. So in the above example, what was previously a choice between `NOUN-PC14-SYNTH-LOC-LEX`, `NOUN-PC14-SYNTH-IN-LEX`, and `NOUN-PC14-SYNTH-ON-LEX` now becomes one rule called `NOUN-PC14-SYNTH-ADP-LEX`. To make this work, the case hierarchy must be updated at the time of creating this rule. The diagram in (4) shows what the case hierarchy for this example would look like before I made this change; the adpositional cases and any morphologically marked cases all show up as direct subtypes of the type *case*. After the change, adpositionally marked cases are a subtype of a newly created supertype meant to represent any adpositionally marked case, shown in (5). The synthetic rule can now assign the CASE value as this new type, allowing for the same functionality as before with mixed marking languages, but with reduced ambiguity.

(4)



(5)



4.3 POS Mis-tagging

The next avenue for reducing ambiguity was improving the way MOM marked stems as being either nouns or verbs in the lexicon. As shown in Figure 4.2, the system saved verb and noun lexical entries for the token *ha-took*. After looking through a number of sentences in

the development languages in the LKB parse compare interface, it became evident that this problem was fairly pervasive. While it is possible for there to be words that are both verbs and nouns (as is ubiquitous in English), a closer look made it clear that this phenomenon was happening too frequently for words that were very unlikely to serve as both nouns and verbs. A cursory glance of some of the baseline lexicons produced by the system for Hiaki showed the following likely nouns as having both verb and noun entries: *chuu'u* ('dog'), *hoara* ('home'), *kovanao* ('governor'), *tami* ('tooth'), and others.

Example (6) shows a sentence from the Hiaki corpus that contributed to the mis-tagging problem when MOM was reading the data.

(6) nee aa= tu'ure hiakinooka-po nee-u nooka-wa-m-ta
1.SG.NOM 3.SG.ACC like speak.in.Hiaki-LOC 1.SG.ACC-DIR speak-PASS-PL-ACC
'I like speaking Hiaki' [yaq] (Harley, 2019)

The final token in the sentence, *nooka-wa-m-ta* contains the verbal root *nooka*, which means 'to speak,' however when encountering this sentence MOM classified this as a nominal root. Close inspection of the code showed that when looking for verbal or nominal roots in an IGT item, the first thing the system would do is search for the Xigt tier that encoded POS information. The POS tier for this sentence is shown in Listing 4.1.

```
<tier id="x" type="syntax" alignment="w">
    <item id="x1" alignment="w1">WkPro</item>
    <item id="x2" alignment="w2">Pro</item>
    <item id="x3" alignment="w3">V</item>
    <item id="x4" alignment="w4">PP</item>
    <item id="x5" alignment="w5">PP</item>
    <item id="x6" alignment="w6">N</item>
  </tier>
```

Listing 4.1: Xigt POS tier for the sentence in (6) (Harley, 2019)

Based on this POS tagging MOM will select four of the tokens to contribute to the lexical and morphological components of the output grammar: *nee* (POS tag WKPRO), *aa* (POS

tag PRO), *tu'ure* (POS tag V), and *nooka-wa-m-ta* (POS tag N). Because MOM is only interested in nouns and verbs, it will only analyze morphology on tokens whose POS tag belongs to the list of noun POS tags or the list of verb POS tags. So even though *nooka* is a verbal root, the Xigt corpus POS information for the whole token is N. Looking at other information in the Xigt item, it becomes clear that this is because the literal translation of the token is 'that.which.was.spoken (obj)' as stated in the tier that translates each token in full.

```
<tier id="wt" type="wordtranslations" alignment="w">
  <item id="wt1" alignment="w1">1.SG.NOM</item>
  <item id="wt2" alignment="w2">3.SG.ACC</item>
  <item id="wt3" alignment="w3">like</item>
  <item id="wt4" alignment="w4">speak.in.Hiaki</item>
  <item id="wt5" alignment="w5">to.1.SG.ACC</item>
  <item id="wt6" alignment="w6">that.which.was.spoken (obj)</item>
</tier>
```

Listing 4.2: Per Word translations tier for the sentence in (6) (Harley, 2019)

This shows that it is not necessarily that MOM is making a bad inference, but rather that a finer level of granularity is required to correctly classify the root. The token as a whole is a noun, but that is because of the combination of suffixes that follow the root allow it to function as a noun rather than a verb.

It is with this clue that the mis-tagging can be reduced. Before MOM creates a lexical entry for a candidate verb or noun, I inserted a mixed-morpheme check. This check looks at (1) the POS tag of the full token and (2) the glosses of each affix attached to the root and if there is a mismatch such that there appears to be both "nouny" and "verby" information, then MOM will not make a lexical entry based on this token. The motivation behind using such a check is that if there is a mixture of affixes largely associated with nouns and affixes largely associated with verbs, then there is less reason to be confident about what part of speech the root is, as is the case in the above Hiaki example, where a verbal root had its

part of speech changed in the sentence due to the attached affixes.

To implement this check, it was necessary to establish which glosses would likely be found exclusively on nominal roots and which glosses would likely be found exclusively on verbal roots. Luckily, throughout the development of the AGGREGATION project, a large inventory of glosses and their associated feature values has been collected, and they are marked with their general syntactic function.

The list of primarily noun-associated feature values included the various collected glosses for CASE. Listing 4.3 shows these collected glosses at the time of writing.

```
caseFeatures = {'nom':NOMINATIVE, 'noms':NOMINATIVE, 'acc':ACCUSATIVE ,'
    erg':ERGATIVE, 'abs':ABSOLUTIVE, 'loc':LOCATIVE, 'gen':GENITIVE, 'instr
    ':INSTRUMENTAL, 'inst':INSTRUMENTAL, 'ins':INSTRUMENTAL, 'dat':DATIVE,
    'obl':OBLIQUE, 'abl':ABLATIVE, 'com':COMITATIVE, 'voc':VOCATIVE, 'ben':
    BENE FACTIVE, 'benef':BENEFACTIVE, 'prp':PURPOSIVE_CASE, 'purp':
    PURPOSIVE_CASE, 'rel':RELATIVE_CASE, 'prt':PARTITIVE, 'compv':
    COMPARATIVE, 'eqtv':EQUATIVE, 'eqt':EQUATIVE, 'priv':PRIVATIVE, 'propr'
    :PROPRIETIVE, 'avr':AVERSIVE, 'frml':FORMAL, 'fml':FORMAL, 'trans':
    TRANSLATIVE, 'byway':ESSIVE, 'ess':ESSIVE, 'inter':INTERSETCTIVE, '
    among':INTERSETCTIVE, 'at':AT, 'post':POSTERIOR, 'behind':POSTERIOR, '
    in':IN, 'near':NEAR, 'circ':NEAR, 'infront':ANTERIOR, 'ante':ANTERIOR,
    'nextto':NEXTTO, 'apud':NEXTTO, 'on':ON, 'onhr':ONHONRIZONTAL, 'onvr':
    ONVERTICAL, 'sub':UNDER, 'under':UNDER, 'rem':DISTAL, 'dist':DISTAL, '
    prox':PROXIMATE, 'all':ALLATIVE, 'apprx':APPROXIMATIVE, 'approx':
    APPROXIMATIVE, 'term':TERMINATIVE, 'prol':PROLATIVE, 'vers':VERSATIVE}
```

Listing 4.3: Glosses for case

Regarding verbs, the list of primarily verb-associated feature values included glosses collected for TENSE, ASPECT, MOOD, VOICE, and FORM. Many of the other glosses could not be reliably assigned as primarily noun- or verb-associated, such as glosses for NUMBER, since this feature is often encoded via affixes of both nominal and verbal roots.

With this check in place, the size of the lexicon is reduced. Using *nooka* from (6) as an example, because the mixed-morpheme check will detect the presence of a "verby" gloss

(PASS) and a "nouny" gloss (ACC), this instance of the root *nooka* won't be entered into the lexicon as a noun as it would have before. Therefore, when sentences that contain *nooka* are parsed, there won't be readings making use of a lexical entry that is a noun alongside the more sensible readings that make use of verb lexical entries. Thus elimination of these entries reduces ambiguity.

4.4 Summary

This chapter covered the two improvements that were made as a part of this thesis to reduce lexical and morphological ambiguity. The first change, consolidation of synthetic rules for languages with both morphological and adpositional case marking, reduces morphological ambiguity by inserting fewer non-inflecting lexical rule instances in the output grammars. The second change, performing a check on the feature values that affixes of a particular word assign, reduces lexical ambiguity by imposing stricter requirements on when a lexical entry may be added to the lexicon. The next chapter will cover the results of the experiments on the development and test languages.

Chapter 5
RESULTS OF EXPERIMENTS

In this chapter, I present the results of my experiments on three development languages and then on three test languages. The three development languages, discussed in 5.1, are Abui [abz] (Kratochvíl, 2019), South Efate [erk] (Thieberger, 2006), and Hiaki [yaq] (Harley, 2019). The three test languages, discussed in 5.2, are Tsova-Tush [bbl] (Hauk, 2016–2019), Meitei [mni] (Chelliah, 2019), and Wakhi [wbl] (Kaufman et al., 2020). I then discuss error analysis in section 5.3 to highlight potential issues with the implemented changes. Note that in all results tables, boxes with purple text represent that the metric improved from the previous run. Boxes with orange text represent that the metric worsened from the previous run. Boxes with black text represent baseline or no change. All runs of the experiments were performed with 1 round of compression and a minimum overlap parameter of 0.2 for merging of word and position classes.[1]

5.1 Evaluation Results for Development Languages

5.1.1 Results for Abui (abz)

Table 5.1 shows the results for Abui for each metric for the baseline run, the run after the consolidation of synthetic rules, and the final run after the mixed-morpheme check was added.

Between the baseline run and the run following the consolidation of synthetic rules, the only metric that changed was average readings per sentence, falling from 919.07 to 344.67, reducing ambiguity by about a factor of 3 with no additional effects. This was just about

[1]Code to replicate the results of the final version of the system can be found at `https://git.ling.washington.edu/agg/repro/ecc-2021`

ABUI [abz]	**Baseline**	**Synth Consolidation**	**POS Tagging**
Average readings per sentence	919.07	344.67	141 .77
Coverage by count	657/1568	657/1568	529/1568
Coverage %	41.90%	41.90%	33.73%
TB sentences with valid analysis	12	12	10
TB sentences with only invalid analyses	52	52	41
'Good loss'	–	0	11

Table 5.1: Evaluation statistics for Abui (abz)

the expected result. Since the output grammars had three adpositional case markers, any sentence in Abui that had an analysis with a synthetic rule would necessarily have two companion analyses with the other two available synthetic rules in that same position class. It is also unsurprising that this reduction in ambiguity came with no reduction in coverage. This change served only to consolidate existing analyses, so if for example a sentence in the baseline had three analyses with synthetic rules (one for each adpositional case marker as originally implemented), the system after the consolidation would have an analogous analysis that still used a synthetic rule, but constrained to an underspecified case rather than three analyses each constraining to an individual case value. Because coverage was completely unaffected, and the new analyses would in theory have compatible semantics with the baseline analyses, it follows that the new system would preserve valid treebanked readings, and also not lose any of the sentences that were parsed with only invalid readings, as was confirmed in running the experiments.

Between the version of the system with synthetic consolidation and the version of the system with mixed-morpheme checking, the results are also positive but not as clearly so. There was an additional reduction in ambiguity, where the average number of readings per sentence fell from 344.67 to 141.77. However, coverage also fell from 41.90% to 33.73%.

Overall, though, this means that ambiguity fell 58.87% while coverage only fell 8.17%, making this a seemingly good trade-off. Beyond that, the treebanking metrics give some insight into what sort of coverage loss was experienced. While 2 valid analyses were sacrificed as a result of the coverage loss (a loss of 16.67%), there was also 'good loss' of 11 sentences that were previously being parsed with only spurious analyses (a loss of 21.15%). Most ambiguity reduction work will be paired with a loss in coverage, which is what happened here in implementing the mixed-morpheme check, however the ambiguity loss appears to outweigh the loss of coverage, partially because it is much greater numerically, but additionally because the metrics here show that some of the coverage loss was in fact beneficial.

5.1.2 Results for South Efate (erk)

Table 5.2 shows the results for South Efate for each metric across all runs of the experiment.

SOUTH EFATE [erk]	**Baseline**	**Synth Consolidation**	**POS Tagging**
Average readings per sentence	9284.82	9309.95	9085.89
Coverage by count	139/1875	139/1875	146/1875
Coverage %	7.41%	7.41%	7.78%
TB sentences with valid analysis	2	2	2
TB sentences with only invalid analyses	114	114	119
'Good loss'	–	0	4

Table 5.2: Evaluation statistics for South Efate (erk).

The results table for South Efate shows largely no change between any of the runs of the experiment. Though upon careful observation, it is clear that the average readings per sentence is not exactly the same in any column, and in fact is *higher* following the synthetic rule consolidation, these numbers are overall too similar to make any claims that ambiguity decreased or even increased between any version of the system for this language. Similar

differences appear in the coverage row, where the final column has a slightly higher coverage, and the rows dealing with illegitimately parsed sentences and 'good loss' show that the final version of the system did not keep all of the treebanked sentences that had no valid reading in the baseline. Note that this means that there are 9 sentences with only invalid analyses being parsed in the final version of the system that were not being parsed in the baseline: there was a 'good loss' of 4, which means the number of treebanked sentences with only invalid analyses would drop to 110, but because the value is 119 it means 9 that were not being parsed before are now being parsed.[2] There is some level of nondeterminism in the system, meaning that any two runs of the same version of the system for a given language may not always have exactly the same ambiguity and coverage results, which explains why the numbers do differ slightly, but not significantly.

When investigating some of the sentences that had analyses in South Efate, the synthetic rules that were present in Abui were also found here, suggesting the system determined that South Efate was also a mixed marking language with respect to case. However, the system only inferred that one possible value of case could be marked adpositionally, so the synthetic rule consolidation would not reduce the ambiguity in the same way it did for Abui. For the issue of making a noun lexical entry and verb lexical entry for the same root, this appeared to be less pervasive in South Efate, explaining why this version of the system did not reduce ambiguity either. It is possible that a more finely tuned version of this check could reduce ambiguity here to some degree, but that is left to future work.

5.1.3 Results for Hiaki (yaq)

Table 5.3 shows the results for Hiaki for each metric across all runs of the experiment. Between the baseline run of the system and the run of the system with synthetic rule consol-

[2]This is possible because between the version of the AGGREGATION pipeline in Howell 2020 and my baseline, changes were made to the Matrix which resulted in not all of the sentences that parsed in her version to be parsed in mine. So these 9 sentences were parsing in Howell 2020, but not my baseline, but upon making changes in the course of this thesis, the system picked some of them back up in some of the datasets.

HIAKI [yaq]	Baseline	Synth Consolidation	POS Tagging
Average readings per sentence	130.75	130.75	43.42
Coverage by count	231/2 235	231/2235	203/2235
Coverage %	10.34%	10.34%	9.08%
TB sentences with valid analysis	4	4	3
TB sentences with only invalid analyses	18	18	15
'Good loss'	–	0	3

Table 5.3: Evaluation statistics for Hiaki (yaq)

idation, there were no changes because Hiaki is not a language that the system inferred to have mixed marking on case. However, the version of the system with the mixed-morpheme check showed notable ambiguity reduction, with the average number of readings per sentence falling from 130.75 to 47.33, a reduction of 63.80%. Of course this came with a loss in coverage, falling from 10.34% coverage to 9.08%, but this is only a loss of 1.26%, which is far less drastic than the change in ambiguity. One valid treebanked analysis was lost, which is unfortunately a 25% loss as prior to this change there were only 4. However, this was still paired with a 'good loss' of 3 sentences, meaning 16.67% of sentences with only spurious parses were not being parsed anymore.

5.2 Evaluation Results for Test Languages

The next section covers the results for the test languages: Tsova-Tush (bbl) (Hauk, 2016–2019), Meitei (mni) (Chelliah, 2019), and Wakhi (wbl) (Kaufman et al., 2020).

5.2.1 Results for Tsova-Tush (bbl)

Table 5.4 shows the results for Tsova-Tush for each metric across all runs of the experiment. For Tsova-Tush, the synthetic rule consolidation did not make any notable difference in any

TSOVA-TUSH [bbl]	Baseline	Synth Consolidation	POS Tagging
Average readings per sentence	509.37	512.76	416.64
Coverage by count	355/1601	356/1568	255/1601
Coverage %	22.17%	22.24%	15.93%
TB sentences with valid analysis	5	5	3
TB sentences with only invalid analyses	38	38	28
'Good loss'	–	0	10

Table 5.4: Evaluation statistics for Tsova-Tush (bbl)

of the metrics, however the mixed-morpheme check did. The average number of readings per sentence fell from around 510 to 416.64, a decrease of about 18.31%. This came with an expected drop in coverage, but only of around 6.28%. Furthermore, 2 valid treebanked analyses were lost, but this was paired with a 'good loss' of 10 sentences that were parsing with only invalid analyses.

5.2.2 Results for Meitei (mni)

Table 5.5 shows the results for Meitei for each metric across all runs of the experiment.

The results for Meitei are similar to the results for Tsova-Tush in that the synthetic rule consolidation did not impact the Meitei results between runs but the mixed-morpheme check did. The number of readings per sentence fell from 1261.78 to 607.06, a decrease of 51.88%. Coverage fell from 5.24% to 3.45%, a decrease of 1.79%. Part of this coverage loss being 6 valid treebanked analyses, but additionally losing 18 sentences with only invalid parses.

5.2.3 Results for Wakhi (wbl)

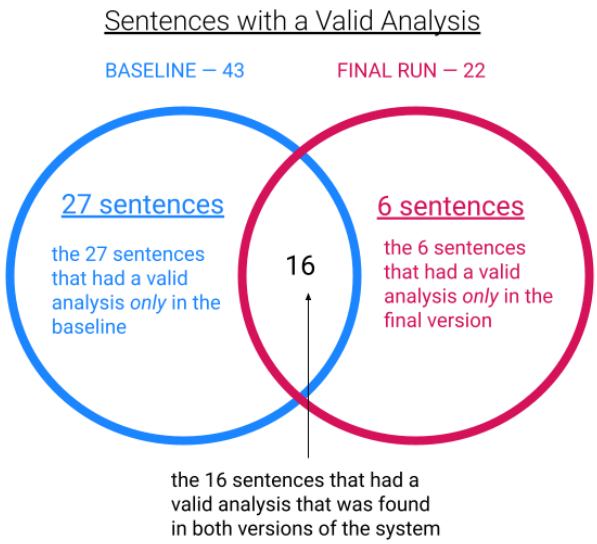
Table 5.6 shows the results for Wahki for each metric across all runs of the experiment. The results for Wakhi show the same pattern as the previous two test languages: no change

MEITEI [mni]	Baseline	Synth Consolidation	POS Tagging
Average readings per sentence	1261.78	1261.78	607.06
Coverage by count	50/955	50/955	33/955
Coverage %	5.24%	5.24%	3.45%
TB sentences with valid analysis	10	10	6
TB sentences with only invalid analyses	40	40	22
'Good loss'	–	0	18

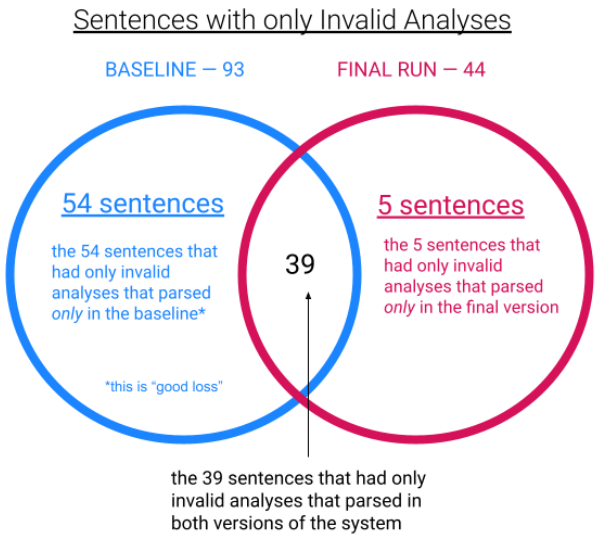
Table 5.5: Evaluation statistics for Meitei (mni)

for synthetic consolidation but a change upon the addition of the mixed-morpheme check. Because the average number of readings per sentence was already fairly low for this language, the reduction from 8.26 to 7.56 readings per sentence (a decrease of 8.84%) is fairly modest, especially when compared with how much coverage was lost. Coverage dropped from 24.16% to 13.03%, a drop of 11.13%. For this language, the details of just exactly what the change in coverage is representing is a lot more complicated than the other datasets. Figure 5.1 illustrates the specifics of how the treebanked sentences were being captured in Wakhi by the baseline system as compared to the final version of the system.

Figure 5.1a shows the coverage of treebanked sentences that had a valid analysis across the baseline and the final version of the system with the mixed-morpheme check. The baseline found a total of 43 of these valid analyses, whereas the final system only found 22. This would suggest that 21 valid analyses were lost in the new version of the system. However, after looking more carefully at which sentences with valid analyses the system was finding, it became evident that this was not the case. Due to non-determinism in the AGGREGATION pipeline, in addition to changes that occurred between the version of the system in Howell 2020 (where the treebanking was performed) and the version of the system that served as my baseline, not all of the valid treebanked analyses were being found in my baseline for



(a) Coverage of treebanked sentences that had a valid analysis



(b) Coverage of treebanked sentences that had only invalid analyses

Figure 5.1: Coverage of treebanked sentences in Wakhi between baseline and final run

WAKHI [wbl]	**Baseline**	**Synth Consolidation**	**POS Tagging**
Average readings per sentence	8.26	8.26	7.56
Coverage by count	165/683	165/683	89/683
Coverage %	24.16%	24.16%	13.03%
TB sentences with valid analysis	43	43	22
TB sentences with only invalid analyses	93	93	44
'Good loss'	–	0	54

Table 5.6: Evaluation statistics for Wakhi (wbl)

some datasets. Specifically for Wakhi, 270 sentences were treebanked; 82 of these had a valid analysis and 188 did not. In my basline, as is shown in Figure 5.1, 43/82 valid analyses were being found and 93/188 sentences with only invalid analyses were being parsed. Because of this, it leaves open the opportunity for changes in the system to "pick up" analyses (valid or invalid) that were missing in my baseline. For the other datasets (with the exception of a small case of this happening in South Efate), this turned out to not be an issue and sentences that were missing in the baseline stayed missing in the second and third versions, and any loss was new loss of sentences that were being parsed in the baseline. However, for Wakhi this was not the case. Between the baseline and the final version, 27 valid analyses were no longer being found, but 6 were "picked up." There were 16 sentences with valid analyses that both systems found.

For sentences that were found to have only invalid analyses during treebanking, the baseline parsed 93, and the final version parsed 44. Of the 93 that were parsed in the baseline, 54 were lost in the form of "good loss" in the final version, but 5 were "picked up," resulting in 44 of them being parsed in the final version, 39 of them being shared between versions.

5.3 Error Analysis

This section will investigate the treebanked sentences with valid analyses that were lost after changes were made to the system for Abui, Hiaki, Tsova-Tush, and Meitei. For South Efate, potential reasons for the excessive ambiguity will be explored including whether it is largely morphological or not. Lastly, for Wakhi, the valid analyses that were lost will be covered in addition to the treebanked sentences that were "picked up" in the final version.

5.3.1 Abui [abz]

Two valid treebanked analyses were lost after the mixed-morpheme check was implemented for Abui. The first Abui sentence that had a valid analysis in the baseline that was lost upon the addition of the mixed-morpheme check is shown in (7).

(7) isi mia
body be.in
'hefty strong' [abz] (Kratochvíl, 2019)

In this short sentence, the gloss for the second token is 'be.in'. In the current mixed-morpheme check, when checking for glosses that are primarily noun- or verb- associated, compound glosses that are delimited by a period are split up and their parts are analyzed. The reason for this is in some cases an affix might encode both tense and aspect, in the case of something such as PST.IMP. Because the collection of currently attested glosses only contains individual glosses and not potential combinations of those glosses, splitting these compound glosses enables the check to realize that the compound gloss PST.IMP is a "verby" gloss. However, in this case, the second part of the gloss, 'in,' gets incorrectly registered as a "nouny" gloss because IN can show up as a gloss for instrumental case. Therefore in this sentence, the splitting of compound glosses is a little too strict and prevents this word from being entered into the lexicon since the POS tag of the word *mia* is "verby" but the second half of the gloss 'be.in' gets mistakenly analyzed as "nouny."

The next Abui sentence with a valid analysis that was lost is shown in (8).

(8) soling iti
edge lie.on
'leaned against' [abz] (Kratochvíl, 2019)

This sentence exhibits the same problem as the one above, except in this case the element of the compound gloss that gets mistakenly registered as "nouny" is 'on' because this sometimes appears as a gloss for case as well, so *iti* does not get entered into the lexicon as the appropriate verb either.

A potential solution to this problem could be outlining which glosses may show up as members of a compound gloss and in what combinations. For example, in the case of a gloss encoding PST.IMP, this would be considered a standard compound because it is common for affixes to encode both TENSE and ASPECT at once. However, in the case of 'be.in', the system would reject the idea of a gloss for instrumental case appearing as a compound with an unidentified gloss (as 'be' is a translation and not a syntactic gloss). This would improve the accuracy of the check and reintroduce the licensing for these lost analyses back into the output grammar. Another solution could involve updating the system to take advantage of whether glosses or parts of glosses are in upper or lowercase, thus the glosses 'in' and 'on' here would not be registered as glosses of CASE in the way that IN or ON would be.

5.3.2 South Efate [erk]

In the case of South Efate, neither of the implemented changes had any notable effect on the metrics. This warrants taking a closer look at what is causing ambiguity in South Efate, since it is the most ambiguous on average of the six languages that were tested in this thesis. Figure 5.2 shows the rules contributing to lexical and morphological ambiguity in a sentence from South Efate. Firstly, one of the tokens, *nen* has a lexical entry as a noun and a determiner. Though it could be possible this string serves both of those purposes, this is a similar problem that the mixed-morpheme check targets, and similar restrictions may need to be implemented when it comes to distinguishing whether a word is a determiner or not.

At the bottom, there are synthetic rules, one for the token *wak* and one for the token

```
? ? det2-determiner-lex           nen
? ? noun180-noun-lex              nen
? ? VERB-PC15-NO-DROP-LEX         i1-pi
? ? VERB-PC36_LRT6-PREFIX         i1-pi
? ? VERB-PC3-NO-DROP-LEX          i1-pi
? ? VERB-PC36-NO-DROP-LEX         i1-pi
? ? VERB-PC15_LRT1-PREFIX1        i1-pi
? ? VERB-PC12-NO-DROP-LEX         i1-pi
? ? NOUN-PC1-SYNTH-ADP-LEX        wak
? ? NOUN-PC1-SYNTH-ADP-LEX        nmatu
```

Figure 5.2: Lexical and Morphological ambiguity for a sentence in South Efate (erk)

nmatu. Inference only found one case that may be marked adpositionally in South Efate, so the consolidation of synthetic rules did not reduce ambiguity as happened in Abui. However, more work could be done on this front as in the case of this grammar these rules are both optional. This is likely due to a potential bug where these rules appear on nouns even when the adpositional case marker that warrants their necessity is not in the sentence. If these rules were only used when the adpositional case marker was in the sentence, this would further reduce ambiguity for South Efate and all other languages with mixed case marking, such as Abui.

There are also a number of rules that are a result of the argument optionality library (Saleem, 2010). The purpose of these rules is to prevent verbs that do not have an affix that marks a particular argument of a verb from being the daughter of certain phrase structure rules. For example, if a language allows for object dropping but requires that verbs take an object marking affix when the object is dropped, verbs without this marker should not be permitted to be the daughter of a rule such as `head-opt-comp-phrase` (Saleem, 2010). Therefore, the `-NO-DROP-LEX` rules were added and they constrain the OPT value on the verb to be −, meaning verbs that go through one of these rules are not eligible to be the daughter of any of the phrase structure rules with dropped verb arguments, as the value for

OPT on these rules is +, creating a conflict. Like the synthetic rules, these rules are also optional and therefore contribute to morphological ambiguity. It may be possible to tweak when these rules are used, like with the synthetic rules. When these rules are present in a grammar, they should apply obligatorily when there is no argument marking affix present, reducing ambiguity as there will no longer be an analysis for the sentence without the rule in addition to an analysis with the rule. Furthermore, the fact that they are showing up in multiple position classes suggests that perhaps the position class merging operation is not consolidating the position classes enough, resulting in the overt markers being scattered across many position classes, therefore causing the strategy of pairing one `-NO-DROP-LEX` rule with every overt argument marking affix to not work well.

5.3.3 Hiaki [yaq]

One treebanked sentence with a valid reading in the baseline was lost in Hiaki after the mixed-morpheme check was inserted. This sentence is shown in (9)

(9) wiikit-m RED2-ne'e
bird-PL Progressive-fly
'Birds are flying.' [yaq] (Harley, 2019)

It is not immediately obvious how the mixed-morpheme check caused the valid analysis for this to be lost, but looking at the grammar specification provides some clues. Listing 5.1 shows the relevant snippets of the grammar specification in the baseline when the valid analysis was present.

```
verb20_stem170_orth=ne'e
verb20_stem170_pred=_fly_v_rel

verb-pc50_name=verb-pc50
verb-pc50_order=prefix
verb-pc50_inputs=verb20
  verb-pc50_lrt1_name=verb-pc50_lrt1
    verb-pc50_lrt1_lri1_inflecting=yes
```

```
9     verb-pc50_lrt1_lri1_orth=red2-
```

Listing 5.1: Baseline specifications snippets for Hiaki (yaq)

The lexical entry for the stem *ne'e* belongs to verb class `verb20`, and stems from this class are eligible to take affixes from position class `verb-pc50` where the prefix *RED2-* lives (inputs not relevant to the current sentence were removed from the snippet for clarity). Listing 5.2 shows the lexical entry for *ne'e* and the position class with *RED2-* in the grammar specification for the final version of the system.

```
1 verb2_stem14_orth=ne'e
2 verb2_stem14_pred=_fly_v_rel
3
4 verb-pc44_name=verb-pc44
5 verb-pc44_order=prefix
6 verb-pc44_inputs=verb40, verb147, verb149
7   verb-pc44_lrt1_name=verb-pc44_lrt1
8     verb-pc44_lrt1_lri1_inflecting=yes
9     verb-pc44_lrt1_lri1_orth=red2-
```

Listing 5.2: Specification snippets from the final version of the system for Hiaki (yaq)

Here, the class that *ne'e* belongs to (`verb2`) is not on the `inputs` list for the position class with the prefix *RED2-*. As a result, this sentence cannot be parsed in the final version of the system. The addition of the mixed morpheme check of course impacted which tokens were able to contribute to the morphology graph and how they were able to contribute. For example, if the prefix *RED2-* appeared with the stem *ne'e* along with other affixes that disqualified it from contributing lexical and morphological information due to the mixed-morpheme check, then the system may never learn this verb can appear with this affix. In a more complex case, it's possible that even in the baseline the training data never contained the exact combination of the prefix *RED2-* with *ne'e*, but that the merging operation allowed them to interact with one another, and the new version of the system started with a different initial graph whose merging operation did not yield this same result.

5.3.4 Tsova-Tush [bbl]

In Tsova-Tush, two valid treebanked analyses were lost after adding the mixed-morpheme check. These sentences are very similar and are shown below in (10) and (11).[34]

(10) meq tet'-o-s
bread cut-PRES-1SG.ERG
'I am cutting bread.' [bbl] (Hauk, 2016–2019)

(11) meq tet'-o-as
bread cut-1S.ERG
'I cut bread.' [bbl] (Hauk, 2016–2019)

The reason these analyses are lost after the mixed-morpheme check is inserted is similar to why the Abui sentences were lost. The glosses 1SG.ERG and 1S.ERG are split due to being compound glosses, and then the presence of ERG gets registered as a gloss for ergative case, causing a conflict between the POS tag for the whole word (which is VERB) and the "nouny" information that ERG is providing. This means that any word in the training data that has this combination of affixes will not have its root entered into the lexicon, and subsequently the system will never learn that the affixes *-o* and either *-s* or *-as* can appear together. This can be seen clearly when comparing the grammar specification of the baseline to that of the version of the system with the mixed-morpheme check. Listing 5.3 shows the relevant snippets of the grammar specification in the baseline.

```
verb166_stem86_orth=tet'
verb166_stem86_pred=_cut (impf)_v_rel

verb-pc2_name=verb-pc2
verb-pc2_order=suffix
```

[3]Originally, what is now glossed as 'cut' was glossed as 'cut (impf) present formant (+impf) cut' in both sentences, but it has been changed for readability of the IGT example.

[4]The glossing for these two sentences is a little unclear, as it appears that the only difference between them is the final suffix, but the glossing seems to indicate that the affix *-o* is what is making the first sentence progressive, despite the fact that this affix is also present in the second sentence.

```
verb-pc2_inputs=verb166
  verb-pc2_lrt1_name=verb-pc2_lrt1
    verb-pc2_lrt1_lri1_inflecting=yes
    verb-pc2_lrt1_lri1_orth=-o

verb-pc22_name=verb-pc22
verb-pc22_order=suffix
verb-pc22_inputs=verb-pc2
  verb-pc22_lrt1_name=verb-pc22_lrt1
    verb-pc22_lrt1_lri1_inflecting=yes
    verb-pc22_lrt1_lri1_orth=-s

verb-pc13_name=verb-pc13
verb-pc13_order=suffix
verb-pc13_inputs=verb-pc2
  verb-pc13_lrt3_name=verb-pc13_lrt3
    verb-pc13_lrt3_feat1_name=person
    verb-pc13_lrt3_feat1_value=1st
    verb-pc13_lrt3_feat1_head=subj
    verb-pc13_lrt3_lri1_inflecting=yes
    verb-pc13_lrt3_lri1_orth=-as
```

Listing 5.3: Baseline specifications snippets for Tsova-Tush (bbl)

For clarity, all members of the `inputs` lists and all lexical rule types and instances that are not relevant to the sentences of interest have been removed. As is shown, the lexical entry for the verb *tet'* is a member of verb class `verb166`, verbs in this class are elgible to have lexical rule instances from position class `verb-pc2` applied to them, and this is where the relevant affix *-o* lives. Then, lexical structures resulting from lexical rule instances in this position class may be input to position classes `verb-pc22` and `verb-pc13`, where the affixes *-s* and *-as* live, respectively. This shows the exact chain of lexical rules that permitted the treebanked analysis to show up in the baseline system. Listing 5.4 shows the relevant snippets in the grammar specification after the mixed-morpheme check was inserted and

illustrates why these sentences were lost in this version of the system.

```
verb178_stem72_orth=tet'
verb178_stem72_pred=_cut (impf)_v_rel

verb-pc2_inputs=verb178
  verb-pc2_lrt1_name=verb-pc2_lrt1
    verb-pc2_lrt1_lri1_inflecting=yes
    verb-pc2_lrt1_lri1_orth=-o
```

Listing 5.4: Specification snippets from the final version of the system for Tsova-Tush (bbl)

In this specification, the verb *tet'* is a member of verb class `verb178`, which may have lexical rules applied from position class `verb-pc2`, where the *-o* affix lives. However, there are no lexical rule instances in any verb position class for the affixes *-s* and *-as* in this case, preventing the sentences from parsing.

Linguists sometimes use case morphemes together with person, number, and gender (PNG) morphemes on verbs to show which argument is being agreed with, as is the case in these lost sentences in Tsova-Tush. The current state of the mixed-morpheme check is too strict for cases like this, so it could be worth doing a frequentistic study in a large corpus of IGT, such as ODIN (Lewis and Xia, 2010), to see how often permitting verb entries with an affix marking something like 1SG.ERG in the lexicon would actually cause problems. It's possible the mixed-morpheme check could treat case morphemes that show up alone differently than those found with PNG glosses, which may benefit languages with corpora like this one.

5.3.5 Meitei [mni]

Four sentences with a valid analysis that were found in the baseline were lost in the final version of the system for Meitei. One of these sentences is shown in (12).

(12) má mə́-yúm=tə hən-lək-e
he 3P-house==LOC return-DISTAL-ASRT

'After that he came home.' [mni] (Chelliah, 2019)

For this sentence, the predication value for the stem *hən*, which is translated here as 'return' is different in the baseline and the final version of the system. In the baseline, the PRED value is `_old-or-plant-or-return_v_rel`, but after the mixed-morpheme check is inserted it is `_plant-or-return_v_rel`. This discrepancy prevents readings of this sentence in the baseline from ever being isomorphic to those in the final version. When the lexicon is being built, MOM keeps track of the various ways a particular stem is glossed, and merges them before the final grammar specification is output. Since the new version of the system changes what data can contribute to the lexicon, it seems that the meaning of 'old' has been lost from this lexical entry and it did not contribute to the PRED value. This same discrepancy where the baseline PRED values were slightly different than the PRED values in the output grammars in the final version of the system also caused the remaining three valid treebanked analyses to be lost. However, even though these sentences did not register as having the same MRS as the original treebanked analyses, the actual analyses in the output grammars of the final version of the system are not any worse, it is just a matter of different PRED values preventing the check of whether the MRSes are isomorphic from passing.

5.3.6 Wakhi [wbl]

As discussed in 5.2.3, the loss of coverage for Wakhi was rather significant, and many more valid analyses were lost between runs than in any of the other datasets. There are three groups of sentences of interest for error analysis: sentences whose valid treebanked analysis was lost between runs, sentences whose valid analysis was "picked up" in the final version, and sentences with only invalid analyses that were "picked up" in the final version. Table 5.7 has one row for each reason that the status of a sentence changed between the baseline and the final run (e.g. in the case of sentences whose valid analysis was lost, a "status change" refers to this loss, but for sentences whose valid analysis was found, a "status change" refers to this finding). The columns represent the three groups of interest, and each cell indicates

how many sentences in that group had their status changed for that reason.

	Valid Lost	**Valid Gained**	**Invalid Gained**
CASE Conflict	3	6	5
ARGS Mismatch	6	0	0
Mixed-Morpheme	18	0	0
TOTAL	27	6	5

Table 5.7: Error analysis statistics for Wakhi (wbl). The count in each cell represents how many sentences in that group were impacted by the reason given in the row title.

As the table shows, one reason, CASE conflict, is the only one that contributes to any group besides the sentences that were lost, and is therefore the sole reason for any sentences being "picked up" between runs of the experiment. Each reason will be covered below.

CASE *Conflict*

The first reason for many sentences to change their status between runs of the experiment was due to CASE conflicts between verbs and nouns. This is the only reason that contributed to sentences being both lost and gained between versions. Example (13) shows one of the sentences that had a vaild analysis in the baseline, but said analysis was lost in the final run of the system.

(13) maz̧ tʂəz̧m-ək di-tu
1SG.OBL eye-DIMINUTIVE hit-PLPF
'I winked.' [wbl] (Kaufman et al., 2020)

In this sentence, the subject has a CASE value of OBL, and the verb *di* has a case frame that requires its subject to be of CASE OBL. However, in the final run of the system, the analysis is lost because the verb *di* now has a case frame that requires its subject to be CASE

NOM, causing a conflict such that the proper analysis is no longer licensed. However, this issue appears to be unrelated to changes I made to the system. Below in (14) is a sentence from another fold of the data that has the same problem between versions with the same verb, however in this case the valid analysis was not present in the baseline and was "picked up" in the new system.

(14) j-a-v j-a-w di-t
DEM-MED-OBL.PL DEM-MED-PRO hit-PST
'They just hit him.' [wbl] (Kaufman et al., 2020)

Here again, the subject of the sentence is of CASE OBL and the verb *di* has a case frame that requires its subject to be CASE OBL in the correct analysis. However, this analysis was *not* present in the baseline, because in the baseline run for this fold *di* had a case frame that wanted its subject to be CASE NOM. This means that in the baseline, the lexical entry for *di* had a case frame requiring an oblique subject in the output grammar from one fold of the data (shown in sentence (13)), and it had a case frame requiring a nominative subject in the output grammar from another fold of the data (shown in sentence (14)). This shows that even within the same version, the system is making different inferences about case frame, suggesting that it is not my change that is causing this discrepancy.

ARGS *Mismatch*

The next reason for valid analyses to be lost was that certain verbs in the lexicon had a different number of ARGS between versions, such that in one version of the system some verbs would be intransitive and then be transitive in another version. This issue affected 6 of the valid analyses. An example sentence where this occurred is shown below in (15). The MRS for the valid treebanked analysis is shown in Figure 5.3, where it is shown that the verb that means 'to hear' is transitive and the ARG2 of this verb is the news, which is sensible based on the translation.

(15) maz̨ j-a xobar kṣ̣ij-tk
1SG.OBL DEM-MED news hear-PRF

'I heard the news.' [wbl] (Kaufman et al., 2020)

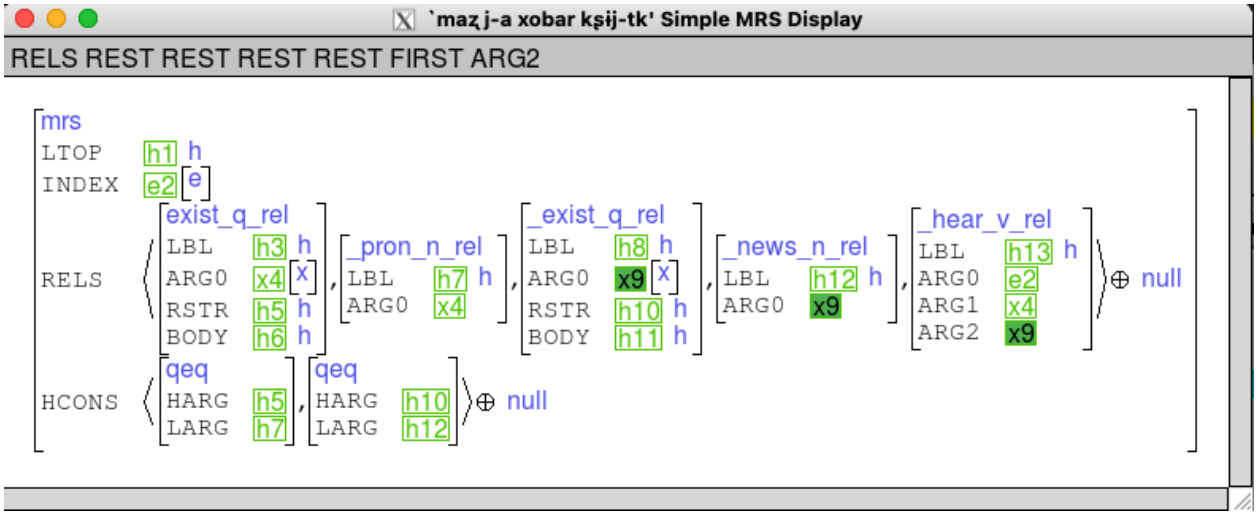


Figure 5.3: MRS for the sentence in (15) [wbl]

However, in the final version of the system, the lexical entry for *kṣɨj* states that it is an intransitive verb, lacking an ARG2 relation, as is shown in Figure 5.4, thus causing the valid analysis of the sentence to be lost.

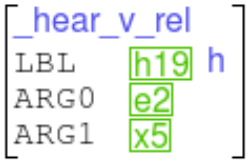


Figure 5.4: MRS for *kṣɨj* [wbl]

Because this only contributed to valid analyses being lost, this suggests something about the changes I made is causing this, but it is unclear how.[5] However, though out of the 6 sentences that were impacted by this, 4 of them were impacted such that the lexical entry's

[5]A tiny bug fix was made that changed the check for whether two verbs had the same case frame, so this could be contributing to this issue.

valence was changed to a less correct valence frame, there were two cases where the change seemed to be more sensible. One such example of this is shown in (16), the MRS of which in the baseline is shown in Figure 5.5 which shows that the ARG2 relation is not identified with anything, suggesting perhaps the intransitive version of the lexical entry in the new version is better.

(16) wuz=əm woq-tu
1SG.NOM==1SG vomit-PLPF
'I threw up.' [wbl] (Kaufman et al., 2020)

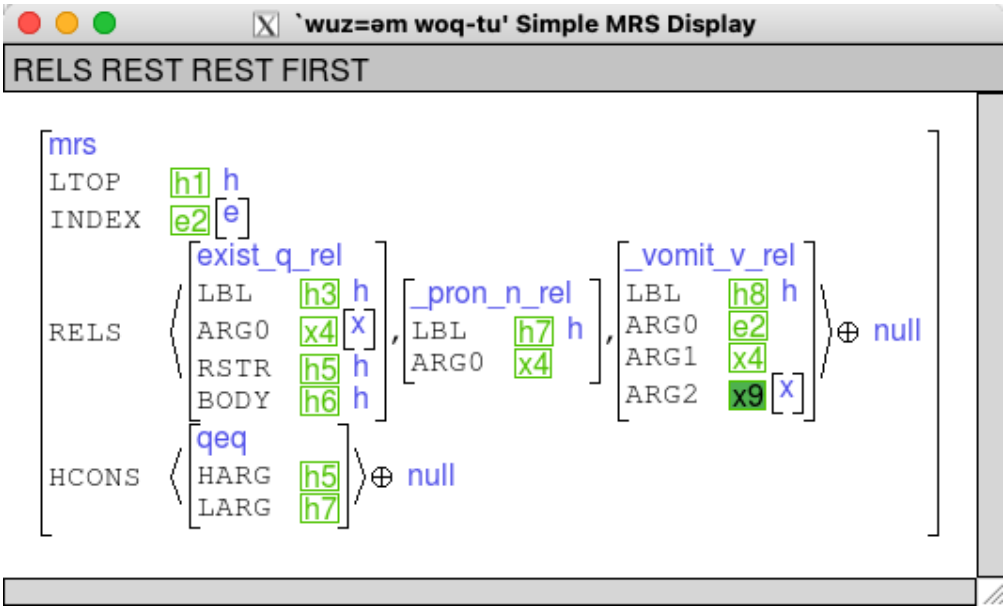


Figure 5.5: MRS for the sentence in (16) [wbl]

Mixed-Morpheme

Finally, the most significant reason for valid analyses lost is due to the addition of the mixed-morpheme check, accounting for 18 of the 27 lost valid analyses. An example of one such sentence is shown below in (17).

(17) sak=ṣ gəfs-ən
1PL.NOM=PROG run-1/3PL.NPST
'We are running.' [wbl] (Kaufman et al., 2020)

The first token in the sentence has a combination of "nouny" information (from the gloss NOM) and "verby" information (from the gloss PROG), which would prevent any word with such a combination of affixes contributing to the lexicon or morphological system in any way as a result of the mixed-morpheme check. Such words with both CASE and ASPECT information are extremely common throughout the Wakhi corpus. One reason this is so pervasive is because in Wakhi there is an aspectual enclitic particle, glossed as PROG in (17), that may be attached either to the verb in the sentence or to another constituent in the sentence that appears before the verb (Obrtelová, 2019). The ability of this enclitic to attach not only to the verb but other constituents explains why there are so many tokens with both "verby" and "nouny" information in this corpus and thus why the loss in coverage following the mixed-morpheme check was so significant.

5.4 Summary

This chapter covered the results of the experiments for both the development languages and the test languages in addition to investigating the valid treebanked analyses that were lost in the new system and pointing out why that loss occurred. Overall, the first change of synthetic rule consolidation only made an improvement to Abui as the problem this change targeted turned out to only be present in the Abui output grammars, but the reduction in ambiguity in the Abui grammars from this change was fairly large with no loss in coverage at all. In terms of the mixed-morpheme check, almost every language benefited from a reduction in ambiguity from this check with an expected loss in coverage, but these losses in coverage for the most part were outweighed by the benefit in the reduction of ambiguity except in the case of Wakhi. The final chapter presents some avenues for future work and closing thoughts.

Chapter 6

CONCLUSION

6.1 Future Work

Though the implemented changes did succeed in reducing ambiguity, there is room to fine-tune particularly the mixed-morpheme check described in section 4.3. One way to improve the check would be to sift through the various available corpora and collect unattested glosses belonging to affixes. Once collected, the sentences they appear in could be investigated and then the glosses could be assigned to the appropriate feature values. Doing this would allow for the discovery of more glosses that could improve the mixed-morpheme check by being classified as either primarily noun- or verb-associated. Beyond improving the mixed-morpheme check, it would expand the system's general knowledge of the ways particular feature-value pairs are glossed in linguist-provided IGT data.

Another way to improve the check would be to allow for the system to take in a file that contains a list of glosses that show up with nouns and a list of glosses that show up with verbs for the given language. Using these lists, greater accuracy of the check on a per-language basis could be ensured. For example, if a particular language only ever marks nouns for number, then affixes with number glosses could be added to the noun-associated gloss list for the language, whereas normally number is too ambiguous of a feature to assign one way or the other.

6.2 Conclusion

The methodology used to find potential lexical and morphological ambiguity culprits involved looking individually at highly ambiguous sentences and noting the kinds of issues that appeared repeatedly across these sentences. Over the course of this thesis, not all of the

spurious forms of lexical and morphological ambiguity were eliminated, so the opportunity exists to use this same method to find more ways to reduce ambiguity. Continuing to reduce ambiguity by this methodology will make the grammars output by the AGGREGATION project more useful, because they will be easier to work with. Though the grammars produced by the system in its current state have relatively low treebanked coverage, they are good starting points for further development of grammars with more coverage, and pursuing more ambiguity reduction work will make them even more useful. Fortunately, anybody who wishes to continue this work will have the ability to use the development and evaluation environment that was set up for this thesis, thus streamlining the process and reducing overhead.

This project has highlighted the value of investigating particular languages in the context of a cross-linguistic system. For example, the extra synthetic rule issue was discovered via investigating Abui, and though, out of the six languages evaluated for this thesis, Abui turned out to be the only one to reap any benefit from this change, that doesn't make it a less important improvement. It may not be as broad as the issue of POS mis-tagging, but it made drastic improvements to the ambiguity of the Abui output grammars. This alone presents a case that doing careful, sometimes narrow, investigation of the output of a complex system is worth the time. Without such investigation, it would be easy to overlook more rare problems such as this, and that would not be appropriately aligned with the spirit of developing a system intended to be a resource for a highly diverse range of languages and linguistic phenomena.

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. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan, 2002.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation*, 8(1):23–72. ISSN 1570-7075. URL `http://dx.doi.org/10.1007/s11168-010-9070-1`. 10.1007/s11168-010-9070-1.

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

Balthasar Bickel, Bernard Comrie, and Martin Haspelmath. The Leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. Max Planck Institute for Evolutionary Anthropology and Department of Linguistics, University of Leipzig, 2008. URL `http://www.eva.mpg.de/lingua/resources/glossing-rules.php`.

David Carter. 1997. The TreeBanker: A tool for supervised training of parsed corpora. In

Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering, pages 9–15, Madrid, Spain, 1997.

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

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. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 695–710.

Scott Drellishak. 2009. *Widespread but not universal: Improving the typological coverage of the Grammar Matrix*. PhD thesis, University of Washington.

Scott Drellishak and Emily M. Bender. 2005. A Coordination Module for a Crosslinguistic Grammar Resource. In Stefan Müller, editor, *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, pages 108–128, Stanford, 2005. CSLI Publications.

Antske Fokkens. 2010. Documentation for the Grammar Matrix word order library.

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. 2013. Generation of machine-readable morphological rules from humanreadable input. *Seattle: University of Washington Working Papers in Linguistics*, 30.

Michael Wayne Goodman. October 2019. A python library for deep linguistic resources. In *2019 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC)*, Singapore, October 2019.

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

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2020. *Glottolog 4.3*. Jena. doi: 10.5281/zenodo.4061162. URL `https://glottolog.org/Accessed2021-02-19`.

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. 2016–2019 (collection date).

Kristen Howell. 2020. *Inferring grammars from interlinear glossed text: extracting typological and lexical properties for the automatic generation of HPSG grammars*. PhD thesis, Seattle.

Fred Karlsson. 2017. *Finnish: A Comprehensive Grammar*. Taylor and Francis, Milton. ISBN 9781138821040.

Daniel Kaufman, Husniya Khujamyorova, and Ross Perlin. 2020. Wakhi texts. *Digital collection managed by KRATYLOS*. Uploaded from www.elalliance.org, Wakhi. In Finkel, R. and Kaufman, D., Kratylos: Unified Linguistic Corpora from Diverse Data Sources. Uploaded April 28, 2020 and retrieved from https://www.cs.uky.edu/ raphael/ela/ on May 20 2020.

František Kratochvíl. 2019. Abui Corpus. Electronic Database: Unpublished toolbox project (accessed March 2019). Nanyang Technological University, Singapore.

William D Lewis and Fei Xia. 2010. Developing odin: A multilingual repository of annotated language data for hundreds of the world's languages. *Literary and linguistic computing*, 25(3):303–319. ISSN 0268-1145.

Elizabeth Nielsen and Emily M. Bender. 2018. Modeling adnominal possession in multilingual grammar engineering. In Stefan Müller and Frank Richter, editors, *Proceedings of the 25th International Conference on Head-Driven Phrase Structure Grammar, University of Tokyo*, pages 140–153, Stanford, CA, 2018. CSLI Publications. URL `http://cslipublications.stanford.edu/HPSG/2018/hpsg2018-nielsen-bender.pdf`.

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

Stephan Oepen. [incr tsdb()] — Competence and performance laboratory User manual. Technical report, Saarbrücken, Germany, 2001.

Stephan Oepen, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Research on Language and Computation*, 2(4):575 – 596.

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

Laurie Poulson. 2011. Meta-modeling of tense and aspect in a cross-linguistic grammar engineering platform. *University of Washington Working Papers in Linguistics (UWWPL)*, 28.

Chris Rogers. 2010. Fieldworks language explorer (FLEx) 3.0. *Language Documentation & Conservation*, 4.

Safiyyah Saleem. 2010. Argument optionality: A new library for the Grammar Matrix customization system. Master's thesis, University of Washington.

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

Nick Thieberger. 2006. 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, Seattle.

Fei Xia, William D. Lewis, Michael W. Goodman, Glenn Slayden, Ryan Georgi, Joshua Crowgey, and Emily Bender. 2016. Enriching a Massively Multilingual Database of Interlinear Glossed Text. *Language Resources and Evaluation (LRE)*, 50(2):321–349.

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

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