Automatic morphosyntactic analysis of Light Warlpiri corpus data

Gina Welsh

Australian National University

October 2020

This thesis is submitted in partial fulfilment of the requirements for Bachelor of Arts with Honours in Language Studies at the College of Arts and Social Sciences.

I hereby declare that, except where it is otherwise acknowledged in the text, this thesis represents my own original work. All versions of the submitted thesis (regardless of submission type) are identical.

Gina Welsh

Table of Contents

Index of Figures

Index of Tables

Acknowledgements

There are several people that I'd like to thank for their assistance in the writing of this thesis.

To my main supervisor, Carmel O'Shannessy, for her excellent supervision over the course of my Honours program. Carmel has provided me with academic, emotional and professional support (in the form of employment) that has helped me enormously in a journey that has at times been very challenging. I have cherished your encouragement, your feedback, and your enthusiasm for my project. Thank you so much.

To my co-supervisor, Mark Ellison, for contributing valuable feedback for my thesis even though he was physically very far away from Canberra. His comments have been much appreciated.

To members of the Centre of Excellence for the Dynamics of Language mentoring team at Appen who helped me with the management of the corpus data in this thesis, as well as the development of the computer program. These individuals include Simond Hammond, Sasha Wilmoth and Jacqui Zwanenburg. I would also like to thank Felicity Meakins who provided me with information on her Gurindji Kriol corpus, a project related to this thesis.

I would also like to thank the Light Warlpiri speakers whose words comprised the data used in this thesis, for their generosity in sharing their language.

To the ANU School of Linguistics for providing a fun and supportive environment to study this incredible discipline in my last couple of years at university. Many of my favourite lecturers have been a part of this school, and I'm so glad to have had them (sometimes more than once!). I am also grateful to have formed friendships in this school, including my pals in last year's Honours cohort: Amelia, Caroline, Kelsey and Steph. I wish you all the best for your future!

To Mum and Dad, and my siblings Emma, Clare, Dan and Max. Thank you for your immense support over the years. You have shown endless faith in me, and I'm so lucky to have a family like you. The group chat messages are always a nice distraction.

Finally, to the weird and wonderful people in Canberra who I came to know and love over the years I've lived here. Thank you for all your well-wishes – it's been a good one.

Abstract

Morphosyntactic analysis aligns a morphosyntactic tag ('gloss') for each word in a given text. Manual morphosyntactic glossing requires significant time and effort to implement on larger scale, such as for a language corpus. Computational methods of automatic analysis can aid in automating this process. In this thesis, I applied a method of automatic morphosyntactic analysis to a set of Light Warlpiri corpus data (O'Shannessy, 2005). The method used the software tool Computerised Language Analysis (MacWhinney, 2000) to apply rules-based word analysis and syntactic disambiguation to the data. My thesis will describe how this method was adapted to the morphosyntactic properties of Light Warlpiri, as well as its performance on the corpus data. Overall, the method was successfully adapted to the Light Warlpiri data, with some recurring challenges noted. Finally, the thesis will discuss the variables within the workflow that affected the adaptation of the method, with emphasis on practical considerations.

Table 1. Abbreviations used in examples.

SG	singular
DL	dual
1	First person
2	Second person
3	Third person
EXCL	Exclusive
INCL	inclusive
ERG	ergative case
DAT	dative case
ABS	absolutive case
ALL	allative case
PERL	perlative case
ABL	ablative case
EVIT	evitative case
POSS	possessive case
LOC	locative case
COM	comitative case
IMPF	imperfective aspect
PERF	perfective aspect
TR	transitive marker
DIS	discourse marker
PST	past
NPST	nonpast
FUT	future
NFUT	nonfuture
REDUP	reduplication
REFL	reflexive
PROG	progressive
IMP	imperative mood
SUBSECT	subsection term
INTERR	interrogative
DIM	diminutive
ANAPH	anaphoric reference
DEM	demonstrative
EPEN	epenthesis
CONJ	conjunction
CAUSE	causative
TOP	topic marker
EMPH	emphatic
DESID	desiderative
REL	relative
INCHO	inchoative

1. Introduction

1.1. Thesis introduction

This thesis will describe and evaluate a method of automatic morphosyntactic analysis applied to corpus data in Light Warlpiri, a mixed language in Northern Australia. Morphosyntactic analysis, that is, the alignment of lexical and grammatical information to textual data, forms a key component of language documentation. The interlinear gloss that results from morphosyntactic analysis adds overt linguistic information to the language data. This information allows researchers to extract meaningful insights into corpora of different languages. However, the manual process of morphosyntactic analysis is time intensive. Therefore, a computational method that automates the process of morphosyntactic analysis is a favourable option in projects with large amounts of transcribed language data. This thesis details the application of automatic morphosyntactic analysis as a contribution to an existing corpus in a minority language project, a topic that has been under-researched in current literature.

1.2. Relevance of thesis within the broader discipline

In the language documentation pipeline, morphosyntactic analysis comes after the audio recording and transcription of language data. The output of automatic morphosyntactic analysis can be used to help integrate corpus data to cross-linguistic language repositories and to assist a research team to analyse their language corpus. Morphosyntactic glossing is also used as input for computational tasks performed further down the pipeline of language documentation, such as parsing (Marcus et al., 1993) or machine translation (Yorick, 2009).

The computational linguistics field has developed numerous methods for automatic morphosyntactic analysis. Many studies have explored the outcomes of these methods applied to languages of a wide range of linguistic typologies (Brill, 2002; Dandapat et al., 2007; Dandapat & Sarkar, 2006; Greene & Rubin, 1971; Leech et al., 1983). However, in many of these studies, there is an over-representation of high-resource languages, and less of minority languages (Moeller & Hulden, 2018). As a result, there is a need in the current literature to represent how these methods are applied from scratch to minority languages.

Automatic morphosyntactic analysis can be divided into three main methods: rule-based, statistical and hybrid. The rule-based method is the oldest type, involving the application of a set of pre-written linguistic rules applied to the textual data (Brill, 2002; Leech et al., 1983). Statistical methods of morphosyntactic analysis involve the creation of a statistical model on a set of training data, that is, corpus data with morphosyntactic annotations. This method uses large amounts of data to predict the morphosyntactic glosses of the language data, without requiring pre-existing linguistic knowledge of the language by the system. Examples of the statistical method include the Hidden Markov model (Rabiner 1989) or the Maximum Entropy Model (Ratnaparkhi, 1996). Hybrid methods exploit the strengths of both rule-based and statistical methods to analyse language in a way that makes use of pre-written linguistic rules as well as training data. There have been numerous studies implementing hybrid methods of morphosyntactic analysis of multiple languages (Dandapat & Sarkar, 2006; Singh et al., 2014). In terms of performance, the application of these methods has ranged in glossing accuracy in multiple studies, with the best taggers performing at 96-98% accuracy (Lv et al., 2016). Each of these methods has its advantages and disadvantages in terms of

effectiveness on the language data as well as time and effort constraints in their development.

The topic of automatic morphosyntactic analysis for Indigenous Australian language data has limited exploration in the current literature. A rule-based morphological analyser for the polysynthetic Australian language Murrinh-Patha was developed to process its complex verbs (Seiss, 2012). In this study, the rule-based method was chosen for the language, since the morphological complexity of Murrinh-Patha proved to be too challenging for statistical methods. There has also been previous work in how computational methods are applied to Australian language data, however, these studies have focused on a different computational problem such as syntactic parsing (Kashket, 1987) or statistical machine translation (Zwarts and Dras, 2007), not morphosyntactic analysis. In terms of specific tools used for language data management, there have been several projects involving Australian languages and the use of computational linguistic tools, such as ELAN (The Language Archive, 2020), FLEx (SIL International, 2020), and Toolbox (SIL International, 2020). For instance, Toolbox implements an automated interlinear glossing method that segments text lines into morphemes using a lexicon.

A related project to this thesis is the ongoing project for the mixed Pama-Nyungan Gurindji Kriol language which involves the use of Computerised Language ANalysis (Meakins, 2007; Meakins and Turpin, 2018). However, unlike the study of this thesis, this project does not make full use of CLAN's automatic glossing feature. Therefore, while there have been several projects involving the management of Australian language data, there has been limited contribution to the literature on automatic morphosyntactic analysis in these languages.

1.3. Thesis objectives

The method in this thesis applies the software Computerised Language ANalysis, or CLAN (MacWhinney, 2000) to corpus data in Light Warlpiri. CLAN features an analyser program (MOR) that applies a hybrid method of automatic morphosyntactic analysis, using both a rule-based and a statistical method to analyse corpus data within the specialised format of CLAN (CHAT). This method was chosen as the corpus data of this project was already transcribed using the CLAN software. However, the MOR program had not been applied to it in a way that analysed the morphosyntactic properties of Light Warlpiri using the full functionality of the CLAN software.

The research objectives of this thesis are:

1) to describe the adaptation of the MOR analyser to the Light Warlpiri corpus data,
2) to report the outcomes of this adaptation, and
3) to discuss the advantages of this adaptation, its limitations and its effectiveness on a practical level.

Each of these research objectives aims to contribute to the current literature with some insights into how automatic morphosyntactic analysis can be applied to Indigenous Australian language data.

1.4. Thesis structure

Chapter 1 has introduced the topic, motivation and research objectives of this thesis. Chapter 2 will describe the Light Warlpiri language, with an outline of the history of the language and a description of key aspects of its morphosyntax. Chapter 3 will outline Computerised Language ANalysis (CLAN) and its MOR program, and the computational workflow for the Light Warlpiri automatic morphosyntactic glossing. Chapter 4 will detail each step of the implementation of CLAN on the Light Warlpiri corpus data and list recurring issues that emerged from this process. Chapter 5

will evaluate the performance of the MOR program on the corpus data, with detail of its performance on several aspects of the data. Chapter 6 will discuss the outcomes of the MOR program application on the Light Warlpiri data, with emphasis placed on the practical elements of the computational workflow. Finally, Chapter 7 will summarise each chapter of the thesis and contextualise its contribution to the current literature.

2. Light Warlpiri language

This chapter aims to familiarise the reader with the Light Warlpiri language. I will first outline a brief history of the language and its typological description. Then, I will highlight patterns in the morphosyntactic structure that serve as key interest in the Light Warlpiri textual data for the automatic glossing workflow.

2.3. Sociolinguistic background

Light Warlpiri is a mixed language spoken by the younger community members of Lajamanu, Australia (O'Shannessy 2006, 2009, 2012, 2013). Mixed languages are the fusion of two or more source languages which are consistently identifiable in the resulting language structure. Light Warlpiri emerged in the 1980s among younger speakers of the community as a result of frequent adult code-switching between two language sources: on the one hand Warlpiri, and on the other, Aboriginal English (AE) and/or Kriol. The influence of this code-switching enabled an innovative mixed language system conventionalised by the subsequent generations of the community (O'Shannessy 2006, 2012, 2013).

Figure 1. Towns and communities in the area in which Light Warlpiri, Kriol and classic Warlpiri are spoken. (O'Shannessy, 2006)

At the time of writing, the oldest speakers of Light Warlpiri are approximately forty years old (O'Shannessy, 2020). As such, the Light Warlpiri language has now been expanded and entrenched in the language of the younger community (Meakins & O'Shannessy, 2010; O'Shannessy, 2013, 2016). Children and young adults speak Light Warlpiri most of the time, but also learn and speak Warlpiri and varieties of English, code-switching between them (O'Shannessy, 2006). So, the linguistic situation of the community can be summarised as highly multilingual. In the Light Warlpiri literature, the Aboriginal English and Kriol languages tend to be merged as the same source (AE/Kriol) because of the difficulties in

separating the sources in their highly multilingual language contact environment (O'Shannessy 2006, 2013).

A combination of both Warlpiri and AE/Kriol structures is found consistently in the mixed grammar of Light Warlpiri. Some properties of these language sources are detailed in the next section.

2.4. Language sources

Warlpiri is a Pama-Nyungan language spoken by approximately 4,000 people in remote communities of the Northern Territory, Australia. It has been studied by several scholars (Hale, 1983; Nash, 1986; Simpson, 1991, Laughren 1999) as a non-configurational language of the area.

(1)	*Wati*	*ka– Ø*	*ngurra–kurra*	*ya–ni*
	Man	IMPF–3SG	home–ALL	go–NPST

"The man is going home." (Laughren et al. 1996: 71)

(2)	*Karnta–ngku*	*ka–Ø–Ø*	*ngapa*	*nga–rni*
	Woman–ERG	IMPF–3SG–3SG	water	drink–NPST

"The woman is drinking water." (Laughren et al. 1996: 85)

The examples (1) and (2) above show how a minimal finite verbal clause in Warlpiri consists of an inflected verb, one or more arguments, and an auxiliary cluster (Hale 1982). The verbal complex forms in Warlpiri are inflected according to five classes (Nash, 1986; Simpson, 1991). This verbal inflection combines with elements of the auxiliary cluster (which typically occurs in the second position of the clause) to provide readings for temporal and modal elements (O'Shannessy, 2006; Laughren 2002). Another notable aspect of Warlpiri is its ergative-absolutive marking system applied to core nominal arguments, in addition to its peripheral case-marking, including those for the locative, allative, evitative and comitative case. Both the ergative-absolutive and peripheral case-marking are carried over to the Light Warlpiri language.

Kriol is an English-lexified creole spoken across northern Australia (Dickson, 2015). Aboriginal English (AE) refers to the varieties of English spoken by the Australian Aboriginal communities.

(3) Aboriginal English

That my brother house.

"That's my brother's house." (Butcher 2008: 632)

(4) Kriol

Yu baj–im–ap det grin wan gap la mi

2SG pass-TR-DIR this green one cup PREP 1SG

"You pass the green cup to me." (O'Shannessy, 2006:13)

(5) Kriol

Hei wot yu luk–ing–at–bat

DIS what 2SG look-PROG-DIR-PROG

"Hey, what are you looking at?" (O'Shannessy, 2006:14)

The Aboriginal English language example in (3) shows the omission of the copula 'be' after the demonstrative pronoun 'that'. Example 4shows the English-derived verb stem 'baj' (pass) combined with a transitive marker '-im' and directional suffix '-ap' (up). Example 5 is an instance of the English-derived verb stem 'luk' combining with the progressive marker '-ing', directional marker '-at', and progressive marker '-bat'. These three examples show phenomena occurring in the Kriol verb complex that do not occur in standard Australian English, which are carried over to the Light Warlpiri language.

2.5. Light Warlpiri morphosyntactic properties

The mixed grammar of Light Warlpiri draws its nominal structures from the Warlpiri source, its verbal structures from AE/Kriol (O'Shannessy, 2012), and its lexicon from both sources (O'Shannessy, 2005, 2013). In the examples below, the elements of Warlpiri are in italicised text, and the elements of AE/Kriol are in plain text.

(6) Warlpiri

nyuntu-lu-rlu	*mayi-npa*	*purra-ja*	*kuyu-ju*
2SG-EUPH-ERG	question-2SG.S	cook-PST	meat-TOP

'Did you cook meat [supper]?' (O'Shannessy, 2012: 4)

(7) AE/Kriol

Yu	bin	kuk-im sapa	indit
2SG	PST	cook-TR supper	tag

'Did you cook supper?' (O'Shannessy 2012: 6)

(8) Light Warlpiri

nyuntu-ng	*mayi*	yu=m	kuk-im	sapa-*ju*
2SG-ERG	QN	2SG.S=NFUT	cook-TR	supper-TOP

'Did you cook supper?' (O'Shannessy, 2012: 5)

The morphosyntactic properties of the main parts-of-speech (nominal, verbal and auxiliary) will be outlined in subsections 2.3.1-2.3.3 below.

2.5.1. Nominal morphology

Light Warlpiri nominals draw their lexical elements from both Warlpiri and AE/Kriol sources, and its grammatical elements from Warlpiri.

2.5.1.1. Core argument marking

On the word level in Light Warlpiri, object arguments of transitive verbs are marked using the ergative marker, as derived from the Warlpiri language source.

(9)

Jinta–kari–ng na i–m ged–im kanta.

One-other–ERG DIS 3SG.S–NFUT get–TR bush.coconut

'The other one is getting the bush coconut." (O'Shannessy, 2016a:16)

(10)

Fatha–wan–ing i–m kam–at–im

wiil–jangka

Father–one–ERG 3SG.S–NFUT come–out–TR

wheel–ABL

'The father got it out from the wheel." (O'Shannessy, 2016a:17)

The ergative marker in Light Warlpiri has forms that are the reductions of allomorphs that occur in classic Warlpiri, as seen in Table 2. In the set of these allomorphs, the *-ng* form is used significantly more often than the other forms (O'Shannessy, 2016b).

Final vowel of word stem	/a/, /u/	/i/
Classic Warlpiri		
Word stem 2 morae	*-ngku*	*-ngki*
Word stem 3+ morae	*-rlu*	*-rli*
Light Warlpiri		
Word stem (2 morae)	*-ngku/-ngu/-ng*	*-ngki/-ngi/-ng/-ing*
Word stem (3+ morae)	*-rlu/-ngku/-ngu/-ng*	*-rli/-ngki/-ngi/-ng*

Table 2. Light Warlpiri ergative marker allomorphy according to word stem and final vowel (O'Shannessy, 2006).

The dative marker *-ik* or *-k* (additional forms *-ki* and *-ku*) attaches to nominals in Light Warlpiri, as highlighted in bold in examples (11) and (12) below.

(11)

Nungarrayi i-m tok-ing Napangardi-k

Subsect 3SG-NFUT talk-PROG subsect-DAT

"Nungarrayi is talking to Napangardi." (O'Shannessy 2006: 68)

(12)

Pakarra-ng i-m luk-raun futbal-ik.

Name-ERG 3SGS look-around football-DAT

"Pakarra looked around for the football." (O'Shannessy, 2016a:232)

The AE/Kriol dative preposition *bo/fo* (O'Shannessy, 2016b) is also used, derived from the English word *for*, as seen in (13), where the pronominal suffix *-im* is attached.

(13)

Wat yu-m do-im bo-r-im?

What 2sg-NFUT do-TR for-EPEN-3SG

"What did you do to her?" (O'Shannessy 2006: 66)

2.5.1.2. Peripheral case-marking

Light Warlpiri has peripheral case-marking derived from the Warlpiri language source. These instances occur as a suffix attached to the stem of the nominal, as seen highlighted in bold in example (14), or as an additional suffix attached to another nominal suffix, as seen in bold in examples (15) and (16).

(14)

Get-im kap Nungarrayi-**kirlang** kurnta-**nga**!

Get-TR cup skin.name-**POSS** shame-**LOC**

"Get the cup, its Nungarrayi's, shame!"

(Meakins and O'Shannessy, 2005: 59)

(15)

I-m *pantirn*-im naif-*kurlu*-***ng***

3SG-NFUT pierce-TR knife-COM-**ERG**

"He pierced it with a knife." (O'Shannessy, forthcoming)

(16)

Jarntu-ng i-m jeis-im pujikat-*pawu wita-pawu* wiri-jarlu-**ng.**

Dog-ERG 3SG-NFUT chase-TR cat-DIM small-DIM big-INTENS-**ERG**

"A big dog chased a small cat." (O'Shannessy, forthcoming)

In the case of an ergative marker occurring with an additional case-marker, the ergative marker attaches as the rightmost suffix, as seen in (15) and (16).

2.3.1.3. Nominal morphology template

Light Warlpiri nominal suffixes	Surface forms
Ablative	-warnu, -janga, -jangka
Allative	-kirra, -kurra
Comitative	-kirli, -kurlu, -kirl, -kurl
Dative	-k, -ki, -ku, -ik, -ing
Diminutive	-pawu, -pardu
Emphasis	-waja, -jala
Ergative	-ng, -ngi, -ngu, -ngki, -ngku, -rli, -rlu
Evitative	-kijaku, -kujaku, -kijak, -kujak
Locative	-ng, -nga, -ngka,-rla
Particle	-piya
Perlative	-wana
Possessive	-kirlangu, -kurlangu, -kangu, -kang
Suffix	-kari
Topical	-ji, -ju, -j

Table 3. Nominal suffixes and their allomorphs in Light Warlpiri.

The full set of nominal suffixes in Light Warlpiri are displayed in Table 3. Their surfaces forms have allomorphic variation, some of which are conditioned on vowel harmony (e.g. *-kirra* and *-kurra*) and others the result of phonological reduction (e.g. *-ng*).

Word	Slot 1 (stem)	Slot 2 (suffix)	Slot 3 (suffix)
1	karnta ('woman')		
2	watiya ('tree')	-nga (LOC)	
3	jarntu ('dog')	-pawu (DIM)	-ng (ERG)

Table 4. A morphological template of nominals in Light Warlpiri, with three examples.

The morphological template for Light Warlpiri nominals is in Table 4. This template shows the available 'slots' for the nominals regarding its suffixing patterns. In this table, word 1 shows an instance of a whole word with no suffixes (*karnta*, 'woman'). Word 2 shows an instance of a stem with one suffix (*watiya-nga*, 'tree-LOC). Word 3 shows an instance of a stem with two suffixes (*jarntu-pawu-ng*, 'dog-DIM-ERG'). In instances where an ergative marker occurs with a peripheral case marker, the ergative marker is always the terminal suffix (O'Shannessy, 2006).

2.5.2. Verbal morphology

Light Warlpiri verbal morphology is derived from the AE/Kriol language source, with lexical items drawing also from the Warlpiri source.

Multiple types of suffixing occur with the verb complex in Light Warlpiri. Transitive verbs are typically marked with the suffix *-im* on the verb stem, as highlighted in bold in examples 17 and 18 below. Example 19 has an intransitive verb "go", and therefore does not contain the *-im* suffix.

(17)

Jarntu-ng i-m jeis-im pujikat-pawu wita-pawu wiri-jarlu-ng.

Dog-ERG 3SG-NFUT chase-TR cat-DIM small-DIM big-INTENS-ERG

"A big dog chased a small cat." (O'Shannessy, forthcoming)

(18)

Botul-ing i-m *panturn*-um taya

Bottle-ERG 3SG-NFUT pierce-TR tyre

“A bottle pierced the tyre.” (O’Shannessy, forthcoming)

(19)

Jakarra yu garra go junga a kan rid

subsect you FUT go true 1SG can:neg read

“Jakarra you have to go really, I can’t read. “

(O’Shannessy, 2006:37)

The iterative marker *-bat* can attach after the *-im* suffix on a verb stem, regardless of the source language of the stem, as seen in (20).

(20)

De-m fix-im-bat kurupa-kurlu-ng

3PL-NFUT fix-TR-ITER crowbar-COM-ERG

“They fixed it using a crowbar.” (O’Shannessy, forthcoming)

The Light Warlpiri verb can also take the progressive *-ing* suffix on both transitive and intransitive verbs, as seen in (21) and (22). In (21), the suffix *-it* occurs in the progressive verb complex (*heb-ing*, ‘having’) functioning as the transitive marker.

(21)

Yakarra	i-m	heb-ing -it	loli
DIS	3SG-NFUT	have-PROG -TR	lolly

"Gosh, she's having the lolly." (O'Shannessy, 2006: 47)

(22)

A-m	weit-ing	tarnnga-juk
1SG-NFUT	wait-PROG	long-time-still.

"I've been waiting for a long time." (O'Shannessy, 2006: 48)

Overall, verbs in Light Warlpiri consist of a stem derived from either Warlpiri and AE/Kriol language sources, and up to three suffixes indicating tense, aspect or valency, as indicated in Table 5.

Light Warlpiri verbal suffixes	**Surface forms**
Iterative	*-bat*
Progressive	*-ing, -in, -*
Transitive	*-im, -um, -am, -it*
Directional	*-dan, -ap*

Table 5. Verbal suffixes in Light Warlpiri.

Table 5 shows a morphological template of Light Warlpiri verb complexes. The verb template has four 'slots', meaning that up to four morphemes can be found in the verb complex. The word "hit-im-bat-im" is an example of a word which takes the maximal number of morphemes, as shown in Example 4 of Table 6.

Example	Slot 1 (stem)	Slot 2 (suffix)	Slot 3 (suffix)	Slot 4 (suffix)
1	go			
2	hit	-im (TR)		
3	hit	-im (TR)	-bat (ITER)	
4	hit	-im (TR)	-bat (ITER)	-im (TR)

Table 6. Morphological template of verb complexes in Light Warlpiri, with four examples.

2.3.3. Auxiliary paradigm

Light Warlpiri has an auxiliary system that has been innovated by its young native speakers, in that it occurs in neither the Warlpiri nor AE/Kriol sources.

Row number	Tense/aspect	1SG	1PL	2SG	3SG	3DU/PL	2DUAL
1	future	a-l	wi-l	yu-l	i-l	de-l	yutu/yumob garra
2	future/should	a-rra	wi-rra	yu-rra	i-rra	de-rra	yutu/yumob garra
3	might/should	a-rra	wi-rra	yu-rra yu mada	i-rra	de-rra	yutu/yumob beta / mada
4 (rare)	might/should	a gada	wi gada	yu gada	i gada	dei gada	yutu/yumob gada
5	want to	a-na	wi-na	yu-na	i-na	de-na	yutu/yumob wana
6	non-future (i.e., present or past)	a-m	wi-m	yu-m	i-m	de-m	yutu/yumob / yutu/yumob bin
7 (rare)	past completed	a bin	wi bin	yu bin	i bin	dei bin	yutu/yumob bin
8	past progressive	a was	wi was	yu was	i was	dei was	N/A
9	might	a mait	wi mait	yu mait	i mait	dei mait	yutu/yumob mait
10	past negative	a neban	wi neban	yu neban	i neban	dei neban	N/A
11	present negative	a neba	wi neba	yu neba	i neba	dei neba	yutu/yumob neba
12	future negative	a won	wi won	yu won	i won	dei won	yutu/yumob won

Table 7. Detail of the Light Warlpiri verbal auxiliary system (O'Shannessy, forthcoming).

The forms of Table 7 show the pronominal inflections of the Light Warlpiri auxiliary paradigm, according to their tense or aspect. It derives its surface forms from its AE/Kriol source, but functions semantically and syntactically like the Warlpiri auxiliary, working in tandem with the verb complex to mark tense, mood and aspect. The free pronouns and their variations of Light Warlpiri are found in Table 8.

	First person	Second person	Third person
Singular	ngaju	yuntu	nyanungu
	a (sub)	yu (sub, obj)	i (sub)
	mi (obj)		im (obj)
Dual	ngajarra	nyuntu-jarra	nyanungu-jarra
	wi (sub)	yutu/yurru (sub,	de (sub)
	us (obj)	obj)	dem (obj)
Plural	ngalipa (incl)	nyurrla	de (sub)
	nganimpa(excl)	yumob (sub,	dem (obj)
		obj)	

Table 8. Forms of free pronouns in Light Warlpiri (O'Shannessy, 2006).

The auxiliary paradigm in Light Warlpiri comes with its system inflecting for tense, aspect and mood. Table 9 shows the word template for auxiliary words *a* and *arra* in Light Warlpiri, with at most two slots accounting for the bound pronoun and the tense or aspect morpheme.

Example	Slot 1 (stem)	Slot 2 (suffix)
1	a (1SG)	
2	a (1SG)	-rra (FUT)

Table 9. The morphological template for Light Warlpiri auxiliary words, with two examples.

2.5.3. Syntactic elements and word order

Light Warlpiri has variable word order. Grammatical relations are marked using both word order and case-marking, as influenced both language sources (O'Shannessy, 2006). However, the most common word order is SVO, as is the same in the AE/Kriol source. The ergative case marker is a key variable on the syntactic level. In a sample of earlier data (O'Shannessy, 2005), ergative case marking occurred on approximately 59% of agent arguments. However, in later data it occurred more often, in approximately 85% of agent arguments (O'Shannessy, 2016b). This indicates that ergative marking in Light Warlpiri is frequent, but not obligatory, as can be seen in the contrasting sentences of (23) and (24). The presence of ergative markers also appears to depend on the word order of the sentence: when the sentence word order is not SVO, ergative marking occurs on 95% of A arguments (O'Shannessy, 2016b). This occurrence can be seen in (25).

(23)

Watiya-ng	i-m	*katirn*-im
Tree-ERG	3SG-NFUT	squash-TR

"The tree squashed it." (O'Shannessy, forthcoming)

(24)

Nyampu-ju laitning na i-m straik-im dat lil boi

DET-TOP lightning FOC 3SG-NFUT strike-TR that little boy

"Here lightning struck the little boy." (O'Shannessy, forthcoming)

(25)

An jinta-kari *wirliya-nga* i-m puk-um jikarla-ng

CONJ one-other leg-LOC 3SG-NFUT poke-TR thorn-ERG

"And the thorn pierced the other one on the leg."

(O'Shannessy, forthcoming)

In addition, ergative marking occurs on instrumental nominals (as seen in 26 and 27) as well as adverbials.

(26)

I-m *pantirn*-im *naif-kurlu-ng*

3SG-NFUT pierce-TR knife-COM-ERG

"He pierced it with a knife." (O'Shannessy, forthcoming)

(27)

De-m fix-im-bat *kurupa-kurlu-ng*

3PL-NFUT fix-TR-ITER crow.bar-COM-ERG

“They fixed it using a crow-bar.” (ERGstoryLC09_2015)

Lastly, the pronominal auxiliary word and the verb complex must occur contiguously in a Light Warlpiri sentence, regardless of the word order variation (O’Shannessy, 2010). An example of this co-occurrence can be found in example (28).

(28)

de-m jeis-ing-it *kuuku det tu karnta-jarra-ng*

3PL-NFUT chase-PROG-TR monster DET two girl-DUAL-ERG

'Those two girls are chasing the monster.' (O’Shannessy, 2010:7)

Category	Part-of-speech	Abbreviation
1	Adverbial	ADV
2	Anaphora	ANAPH
3	Article	ART
4	Case	CASE
5	Conjunction	CONJ
6	Determiner	DET
7	Directional word	DIR

8	Directional suffix	SUF:DIR
9	Disjunction	DISJ
10	Topic marker	TOP
11	Interrogative	INTERR
12	Kinship term	N:KIN
13	Negation	NEG
14	Noun	N
15	Numeral	NUM
16	Preposition	PREP
17	Pronoun	PRO
18	Proper noun	N:PROP
19	Quantifier	QAN
20	Suffix	SUF
21	Verb inflection	V:INFL
22	Verb (intransitive)	V:INTRAN
23	Verb (modal)	V:MOD
24	Verb (transitive)	V:TRAN
25	Verbal auxiliary	V:AUX
26	Wh-word	PRO:QN

Table 10. The part-of-speech category set for the Light Warlpiri language.

The full set of parts-of-speech for Light Warlpiri is shown in Table 10. This part-of-speech schema was taken from the part-of-speech set designated for the Light Warlpiri corpus (O'Shannessy, 2005, 2010).

In summary, Light Warlpiri syntax uses a combination of case-marking and word order to determine grammatical relations. There is some variation in word order, but some elements must co-occur to be grammatical.

2.6. Chapter summary

This chapter introduced the Light Warlpiri language and its language sources, Warlpiri and Aboriginal English/Kriol. It showed various elements of its grammatical and lexical features, including its nominal, verbal morphology and its auxiliary paradigm. Some parts of speech (such as the disjunctions, adverbials and determiners) were not discussed explicitly in this chapter since they do not have morphosyntactic marking elements that are of focus for the automated workflow. However, these parts-of-speech are included in the results of the automated analysis in Chapter 5.

3. Computational workflow

This chapter aims to familiarise the reader with the computational workflow for the Light Warlpiri data analysis. The first section outlines the TalkBank and CHILDES projects, the collaborative projects relevant to the Light Warlpiri data. The second section outlines Computerised Language Analysis (CLAN), the software tool used to implement the automatic workflow. The third section details the morphosyntactic analyser (MOR grammar) embedded in the CLAN tool. The last section summarises the MOR grammar workflow.

3.3. The TalkBank and CHILDES projects

The Light Warlpiri data (O'Shannessy, 2005, 2010) have been transcribed according to the data format (CHAT) of the TalkBank repositories. The TalkBank repositories comprise a collaborative project that supports the open exchange of linguistic corpus data between language researchers (MacWhinney, 2000). This project has been contributed to by an international community of hundreds of researchers, with 34 languages represented in the project. TalkBank provides data sources relevant to multiple language-related research areas, including conversational analysis (CABank, ClassBank).

Speech-language pathology (AphasiaBank, FluencyBank) and child language acquisition (CHILDES).

Since the TalkBank project operates as an open-source collaboration network, any language researcher can commit their field data to the repositories with approval of the project administrators. Several researchers have contributed child language data to the CHILDES project, including French, Japanese, Indonesian, Tamil and Icelandic, enabling a means of cross-linguistic comparison of child speech patterns.

3.4. Computerised Language Analysis

The Light Warlpiri corpus data used in this thesis has been transcribed for the CHILDES project. So, it can be processed using the project's default tool for transcription and analysis, Computerised Language ANalysis (or CLAN). CLAN is a free, open-source software program that is downloadable from the TalkBank website. The files processed by CLAN are in the specialised CHAT data format, allowing audio and video files to be linked to the textual data. The CLAN software is regularly maintained, with updated versions having been released every few months for several years. Its users are also supported by a Google Group (`chibolts`),

providing a means for users of the CLAN program to receive technical help from its founder and maintainers.

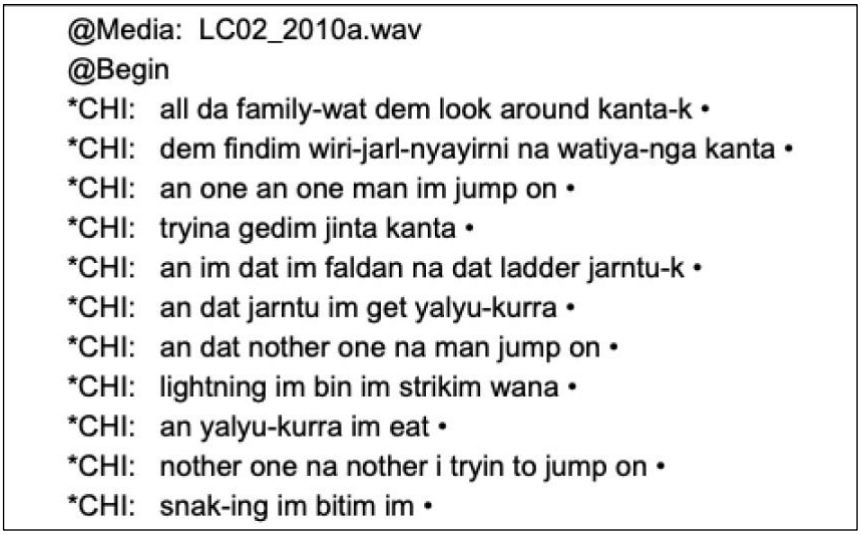

Figure 2. A CHAT transcript sample with Light Warlpiri textual data.

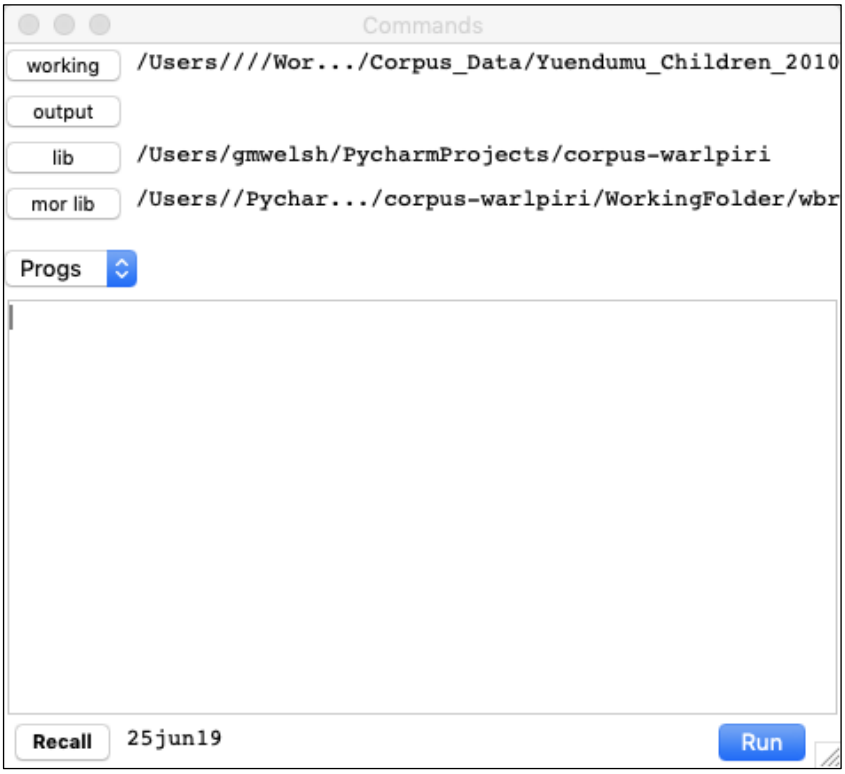

Figure 3. The CLAN user command prompt.

CLAN is run from a user command prompt (see Figure 3), where a set of transcribed CHAT transcripts (in the 'working directory') are used as input. The user prompt can use a variety of commands to analyse the language data, such as word frequency analysis (`FREQ` command) or searching for the context of a specific word in several files (`KWAL` command). The results of the analysis are shown in an output window (see Figure 4).

```
> kwal +t%mor +s*erg* -w1 *.mor.cex
kwal +t%mor +s*erg* -w1 *.mor.cex
Wed Jun 22 15:25:33 2005
kwal (06-Mar-2003) is conducting analyses on:
ALL speaker tiers
          and those speakers' ONLY dependent
tiers matching: %MOR;

From file <ERGstoryLA21_1.mor.cex>

*** File "ERGstoryLA21_1.mor.cex": line 44.

Keywords: cas|ng:erg@1, cas|ng:erg@1

*A21: kuuku-ng i-m tak-im jarntu .
%mor: N|monster@1-ng:erg@1 PRO|i:3sg-NONFUT
Vtr|take-im:trans N|dog@1 .
*A21: karnta-pawu-ng an baby-pawu-ng de-m
chas-im .
%mor: N|woman@1-dim@1-ng:erg@1 CONJ|an N|baby-
dim@1-ng:erg@1 pro|de-m:NONFUT Vtr|chas-
im:trans .

From file <ERGstoryLA21_2.mor.cex>
```

Figure 4. Example output of a CLAN command (KWAL) on Light Warlpiri data.

A significant number of CLAN commands are built to assist language researchers in analysis the morphosyntactic properties of their corpus data. One of the commands is the `mor` command, which enables morphosyntactic analysis of the textual data in the input transcripts. The analysis is performed by the MOR grammar, which is stored as a set of files in the program. The MOR grammar outputs the same transcripts with an added morphological tier (`%mor` tier)

containing the results of the MOR grammar's analysis (see Figure 5).

```
@Begin
@Participants:
@ID:   A21 ERGstoryLA21-1
*A21:   karnta-pawu i-m sit-ing jarntu-kurl swing-wana .
%mor:   n|woman@1-dim@1 pro|i:3sg-m:NONFUT v:intran|sit-ing:prog
      n|dog@1-com@1 n|swing-perl@1 .
*A21:   karnta-pawu i-m get-ap, get-im jarntu kuuku-kujaku .
%mor:   n|woman@1-dim@1 pro|i:3sg-m:NONFUT v:tran|get-up
      v:tran|get-im:tran n|dog@1 n|monster@1-evi@1 .
*A21:   karnta-pawu i-m get-im kuuku na kurdu, jarntu kuuku-kujaku .
%mor:   n|woman@1-dim@1 pro|i:3sg-m:NONFUT v:tran|get-im:tran n|monster@1
      dis|foc:now n|child@1 n|dog@1 n|monster@1-evi@1 .
```

Figure 5. A Light Warlpiri CHAT transcript with an added morphosyntactic (%mor) tier.

In the CHILDES repository, several working MOR grammars have been contributed publicly, including English, German, Chinese, Hebrew, Japanese and Italian. These MOR grammars can be accessed using an inbuilt command in the CLAN interface. However, if there is no working MOR grammar available for a specific language, it has to be built by a minimal set of files in the CLAN program. Since there is no contributed MOR grammar for Light Warlpiri data, building a MOR grammar is necessary to enable full use of the `mor` command for this data. As a result, this thesis describes the process of adapting a MOR grammar to the Light Warlpiri data so that it can be contributed to the set of MOR grammars.

3.5. The MOR grammar components

The rules-based component of the MOR grammar combines a two-level morphology model (Koskenniemi, 1983) with left-associative grammar analysis (Hausser 1986, 1999). Both these mechanisms are language independent, that is, able to be applied to any human language given that they are adapted to the structures of a language.

The two-level morphology model divides textual data into the *surface* and *lexical* levels. The surface level is the actual realisation of the words as they appear in textual form. The lexical level refers to the combination of stems and affixes that are divided by morphological boundaries.

```
Surface level: 'swimming'

Lexical level: 'swim' + 'ing'
```

Figure 6. A two-level representation of the English word 'swimming'.

The two-level morphology system breaks down a word's surface into standard forms of its component morphemes. Then, these morphemes are mapped onto morphosyntactic information in a lexicon. The left-associative component of the MOR grammar refers

to the direction in which the grammar analyses the word surfaces and their linguistic combinations. In the left-associative direction, the surface forms and their subsequent combinations are analysed from left-to-right, as opposed alternative directions, such as top-down word analysis. A formal description of left-associative grammar (LA-grammar), the underlying mechanism of this direction, is as defined as a 6-tuple <W, C, LX, CO, RP, rps> (Hausser, 1986), where:

4. W is a finite set of morpheme surfaces
5. C is a finite set of category segments
6. LX Ì (W x C+) is a finite set comprising the lexicon
7. CO is a finite sequence of total recursive functions called categorial operations
8. RP is a set of rule packages
9. rps is a set of start rule packages.

In the formal definitions above, there is a two-level distinction between string surfaces (W) and morphosyntactic categories I. The lexicon is the set of pairs of string surfaces and morphosyntactic categories (LX). The categorial operation sequence (CO) enables morphosyntactic categories of morphemes to co-occur where valid (the 'rules'). The rule packages (RP) enable certain rules to be fired

after the application of other rules, and the set of start rule packages (rps) contain the initial enabling of the rule chains. These variables are relevant to the implementation of the concatenation rules (C-RULES), explained further in section 3.3.3.

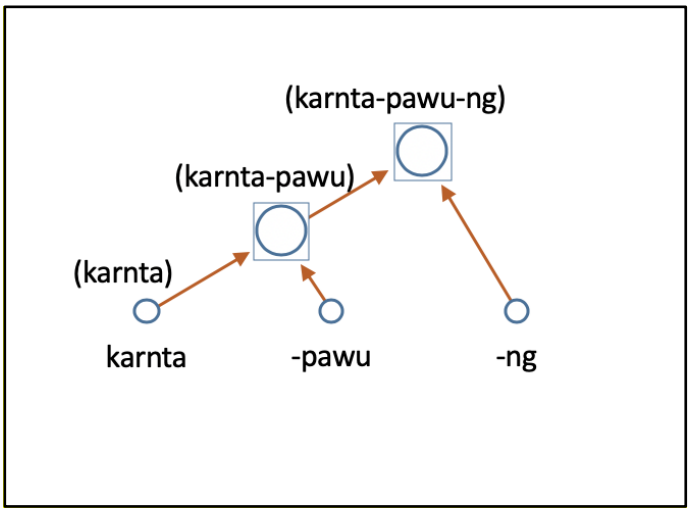

Figure 7. Left-associative direction of combining suffixes in 'karnta-pawu-ng' (woman-DIMINUTIVE-ERGATIVE).

Figure 7 shows a two-level representation of the LA-grammar applied on the word level to the word *karnta-pawu-ng* ('woman-DIM-ERG'). Here, the category information of the word stem *karnta* 'woman' (noun) must be enabled to be combined with the surface form '-pawu' (nominal suffix), the form which must be enabled to be combined with the subsequent form '-ng' (case suffix). Therefore, for the word *karnta-pawu-ng* to be analysed correctly by the rules-based component of the MOR grammar, there must be the categorial rule

{NOUN + NOMINAL:SUFFIX + CASE:SUFFIX} present in the word formation rules.

In the CLAN software program, the MOR grammar is run using several components that contribute different processes to the analyser. These files are:

1) the lexicon (corresponding to the LX set in the LA-grammar definition)
2) the allomorphic rules (A-RULES)
3) the concatenative rules (C-RULES) (corresponding to the CO, RP and rps sets in the LA-grammar definition)
4) the part-of-speech tagging (POST) program.

Each process builds on another sequentially: the contribution of the lexicon is built upon by the A-RULES, the output of which is built upon by the C-RULES file, the output of which is processed by the POST program (see Figure 8).

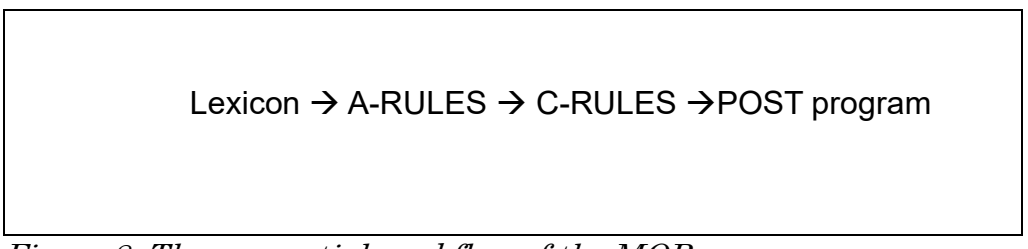

Figure 8. The sequential workflow of the MOR program.

The rule-based component of the MOR program is found in the lexicon, A-RULES, C-RULES and POST processes. The POST program uses both a rule-based and probabilistic method. Each part of the MOR program will be described in the subsections below.

3.3.1. The lexicon

The MOR grammar lexicon is a finite storage of words, stems and affixes of the input language. It contains a set of lexical entries, each entry containing three fields from left to right: a surface string field, a category field ([scat]) and a semantic gloss field (see Figure 9).

```
4056  wardingki      {[scat suf]}    "belonging@1"
4057  wardinyi       {[scat suf]}    "happy@1"
4058  wardiylka      {[scat n]}      "bushturkey@1"
4059  warla  {[scat n]}
4060  warlalja       {[scat n:kin]}  "family@1"
4061  warlapaju      {[scat v:intran]}       "stop@1"
4062  warlku  {[scat n]}     "axe@1"
4063  warlkurru      {[scat n]}      "axe@1"
4064  warlpaju       {[scat v:intran]}       "stop@1"
```

Figure 9. A section of the Light Warlpiri lexicon file. In each entry from left to right: the surface form, the morphosyntactic category ([scat]), and the corresponding gloss. The "@1" symbol indicates a word that is sourced from the Warlpiri language source.

The lexicon is processed first by the MOR grammar before the other components, since it contains the surface and category information of the textual data. If there is no entry for an item in the lexicon file, then the item will not be recognized by the MOR grammar. In addition, if a morpheme forms part of a word in the language data (such as the diminutive case suffix *-pawu* in the Light Warlpiri word *karnta-pawu* ('woman-DIMINUTIVE') and the morpheme is not present in the lexicon file, then the word cannot be recognized by the MOR grammar. Therefore, the lexicon file is the most fundamental component of the MOR grammar.

3.3.2. A-RULES

The allomorphic rules (A-RULES) component enables allomorphic or orthographic variation to occur for an entry in the lexicon. For instance, in the English word "ponies", the terminal symbol of "pony" ('y') is transformed to 'ie' before the plural suffix '-s' is added to the word ("ponies"). The A-RULES file allows this orthographic variation to occur on the surface form, with the rule relevant to "ponies" seen in Figure 10.

```
LEX-ENTRY:

LEXSURF = $Yy

LEXCAT = [scat n]

ALLO:

ALLOSURF = $Yie

ALLOCAT = LEXCAT, ADD [allo nYb]

ALLO:

ALLOSURF = LEXSURF

ALLOCAT = LEXCAT, ADD [allo nYa]
```

Figure 10. An A-RULES statement allowing the terminal symbol 'y' to be transformed to 'ie' in an English noun.

In the A-RULES statements of Figure 10, the surface form of a noun ending in "y" is designated in the LEXSURF statement (LEXSURF = $Yy, where $Y indicates a set of any character combinations). The lexicon item containing this surface must be paired with the noun category (as indicated by the LEXCAT = [scat n] statement). Then, two possible allomorphs of this lexical item are designated by the ALLO conditions. The first ALLO condition designates a possible

surface form $Yie, for the lexical item, meaning any character string ending in -ie (e.g. ponie for the noun stem in “ponies”). This allomorph is assigned the additional category tag [allo nYb]. The second ALLO condition enables the default surface ($Yy) to be assigned the category tag [allo nYa], to apply to all instances of stems ending in -y. In this kind of rule, a lexical item like “pony” can extended to the orthographic variant “ponie”. This allows word stems to change their form according to surface rules of the language. To summarise, the A-RULES component adds possible allomorphs or spelling variants to the lexicon by analysing patterns of surface strings. Light Warlpiri contains allomorphic variation in its suffixes, an example referenced in the ergative marking allomorph description of the previous chapter (2.3.1.1).

The output of the A-RULES program is input for the C-RULES program, which is described in the next section.

3.3.3. C-RULES

The concatenation rules (C-RULES) component enables the surface forms of morphemes to combine according to the word formation rules of the working language (legal word analyses). This component implements the two-level morphology and left-associative grammar

paradigms outlined in 3.3 to determine the parts-of-speech of each morpheme in the input word.

There are several types of rules in the C-RULES file. The main types are the START rules and the suffixing rules. The START rules enable the part-of-speech categories of word stems to be analysed by the MOR grammar. For instance, the rule 'v-start' (shown in Figure 11) uses a NEXTCAT statement to enable the 'verb' part-of-speech to be applied to surface stems in the language data. That is, the rule allows the look-up of all entries in the lexicon with [scat v], [scat v:tran] and [scat v:intran] in their syntactic category fields to be applied as a possible starting node in the analysis of a word.

```
RULENAME: v-start
CTYPE: START
if
NEXTCAT = [scat OR v v:tran v:intran]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {trans-suff, case-foc-v, v-infl, v-loc, suf-gen, bat, v-prep}
```

Figure 11. A START rule in the concatenation rules (C-RULES) file, enabling verb stems to be analysed by the MOR grammar.

The v-start rule of Figure 11 enables subsequent rules to be applied in the suffixes after a verb stem. One of these rules, trans-suff, can be seen in Figure 12.

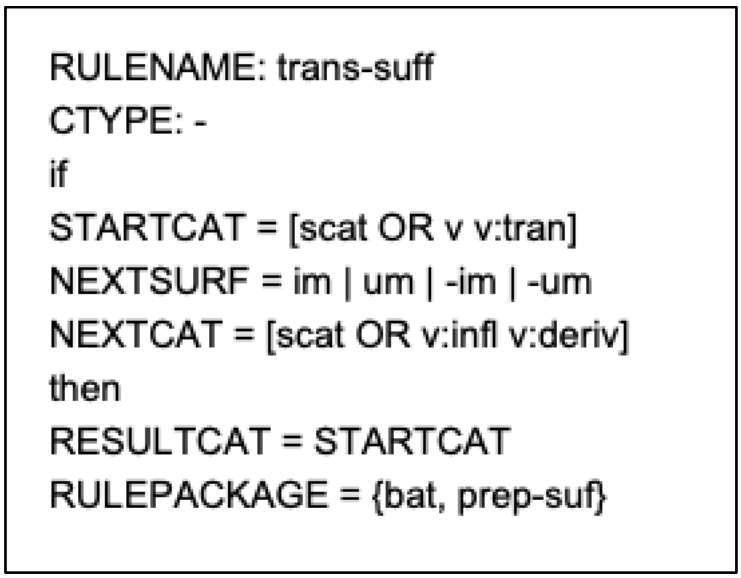

Figure 12. A suffixing rule 'trans-suff' that enables the transitive marker '-im' to be attached to a verb stem.

The trans-suff rule is called as a suffixing rule ("CTYPE: -"), meaning that it can only be called after a starting stem has been analysed in the word form. A conditional statement (STARTCAT) calls for the starting stem to be a verb stem, which rules out all other surface forms that do not have [scat v] or [scat v:tran] in their lexical entries. The rule then calls for the suffix that comes after the verb stem to have the surface form "-im" or "-um" (the NEXTSURF statement) and to have the syntactic category as a verb inflection ([scat v:deriv]). This means that a word analysed under this trans-suff rule must be of a word formation pattern such as ([scat v:tran] ++ (surface form "-im" and [scat v:deriv]). A word that would satisfy

this rule combination in Light Warlpiri would be 'gettim' ("hit" [scat v:tran] + "-im" [scat v:deriv]). Therefore, the rule of Figure 11 and the rule of Figure 12 would work sequentially to output the analysis of this word.

In the C-RULES file, the rules are called according to the categorial information that is applied by the lexicon file to the surface forms. The syntactic context is not taken into account at this point of the analysis. Therefore, the process can over-generate the analysis of an input word since the C-RULES only takes into account the word formation rules of each surface form and not the syntactic context of their occurrences. An example of this can be seen in Figure 13 where the word "im" is interpreted by the analyser as both the auxiliary 3SG-NONFUT (i-m) and the 3rd singular object pronoun (im). This results in the output "pro|3S.Obj^pro|i:3sg-m:NONFUT", where the caret symbol signifies an ambiguity.

```
        mor (:h help)(INPUT)> im
        *** File "/Users/gmwelsh/Downloads/MOR-LW-final-
test/debug.cdc"
        parse 1:
                lex info: {[scat pro]}
                morphemes (surface/stem): 3S.Obj
                compound:
                translation:
        parse 2:
                lex info: {[scat pro]}
                morphemes (surface/stem): i:3sg-m:NONFUT
                compound:
                translation:

        Result: pro|3S.Obj^pro|i:3sg-m:NONFUT
```

Figure 13. An example of an ambiguous output of the MOR grammar.

To summarise, the C-RULES component of the MOR grammar uses the LA-Grammar mechanism to enable morphemes to attach. At times, this results in over-generation, which the POST program must rectify.

3.3.4. POST rules

The POST component changes the output of the C-RULES program according to sentence-level context. It has two 'filters': the PREPOST and POSTMORTEM processes.The PREPOST process rules out ambiguous analyses formed by the MOR program (such as interpreting the word "flying" as both a progressive verb and a noun

with an ergative suffix, i.e. n|flying (fly-ERG) ^v|flying (fly-PROG) by designating hard rules that change the %mor tier when these instances occur to the correct analysis.

```
# two daughters of her own
det:poss|*^* adj|own^v|own => det:poss|* adj|own

# this beautiful dress
det:dem|*^* adj|* => det:dem|* adj|*
```

Figure 14. PREPOST rules in the English MOR grammar.

In Figure 14, the first PREPOST re-write rule narrows the possibility of the word "own" being interpreted as both an adjective or verb (adj|own^v|own) to just being interpreted as an adjective (adj|own) when it occurs after a possessive determiner (det:poss|*^*). The second PREPOST rule changes the MOR analysis of the English phrase 'this beautiful dress' such that the word 'this' cannot be interpreted as any part-of-speech other than demonstrative determiner (such as an object pronoun) when an adjective immediately follows the word. Therefore, it rules out multiple parses for certain words when they occur with other parts-of-speech.

```
# rules for "to"
v*|* prep|to v*|* => v*|* inf|to v*|*
inf|to n|* => prep|to n|*
inf|to det:*|* => prep|to det:*|*
```

Figure 15. The POSTMORTEM rules to interpret the word 'to' in the English MOR grammar.

While the PREPOST component reduces multiple parses for certain words to just one interpretation, the POSTMORTEM component changes the part-of-speech for a word according to sentence context. In Figure 15, the first POSTMORTEM rule changes the interpretation of the word 'to' from the preposition part-of-speech (prep|to) to the infinitive part-of-speech (inf|to) when it occurs between two verbs (v*|*…v*|*). This allows the word to be analysed correctly within specific contexts.

In sum, the POST rules can change the analysis of the MOR grammar where there are specific contexts for word-sense disambiguation.

3.3.5. POSTTRAIN implementation

The POSTTRAIN process is the probabilistic component of the POST program. It implements a Markov model of binary rules (Parisse and LeNormand, 2000) that is created using training data from the corpus. This is used as a complement to the POST rules for word-sense disambiguation on the corpus data that has not been processed by explicit rules. For the POSTTRAIN component to be built up, there must be a designated set of training files that have been analysed by the previous MOR components. These training files contain an extra tier (%trn) where the user indicates the correct reading of the %mor tier (see Figure 16). The POSTTRAIN program compares the differences between the %trn and %mor tiers and records these differences in a probabilistic model (post.db).

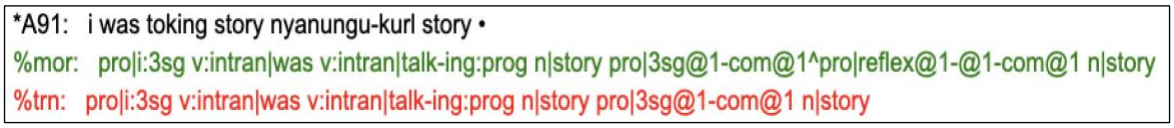

Figure 16. An excerpt of a POSTTRAIN training file. The extra tier '%trn' contains the disambiguated version of the %mor tier.

In Figure 16, the %trn tier has been disambiguated manually to create a discrepancy between the spurious parses of the %mor tier (in green) with its disambiguated counterpart %trn tier (in red). The

discrepancies between the two tiers are recorded in the POST training model, to be applied to later transcript sets.

3.3.6. Building and testing the MOR grammar

To build the MOR grammar for the Light Warlpiri language data, rules had to be devised according to the morphosyntactic structures found in the data. These rules will be further outlined in Chapter 4. The output of the MOR grammar can be tested by running the command prompt in interactive mode (see Figure 17), where specific words can be entered to test the application of the rules.

```
From pipe input

commands:
      word - analyze this word
      :q  quit- exit program
      :c  print out current set of c-rules
      :d  display application of a rules.
      :l  re-load rules and lexicon files
      :h  help - print this message

mor (:h help)> karnta-pawu
*** File "/Users/gmwelsh/Downloads/MOR-LW-final-test/debug.cdc"
parse 1:
      lex info: {[scat n]}
      morphemes (surface/stem): woman@1-dim@1
      compound:
      translation:

Result: n|woman@1-dim@1

mor (:h help)>
```

Figure 17. The word "karnta-pawu" ('woman-DIM') analysed using the interactive mode of the MOR program.

On a larger scale, a run of the MOR grammar can be applied to hundreds of thousands of words in a corpus. The results of the MOR grammar applied to a series of transcripts are described in Chapter 5.

3.4. The MOR grammar components

This chapter outlined the CLAN tool, a software program used to implement the morphosyntactic analyser for the Light Warlpiri data. It described the underlying mechanisms of the MOR program, including an explanation of the key word analysis components (ARULES and CRULES) as well as the word disambiguation component (the POST program). The application of these MOR grammar components to the Light Warlpiri data are described in the next chapter.

4. Implementation

This chapter describes the adaptation of the morphosyntactic analyser (MOR grammar) to the structures of the Light Warlpiri corpus data. The first section will describe the pre-existing set-up of the Light Warlpiri corpus. The second section will outline the implementation of the MOR grammar's word analysis component. The third section will detail the morphosyntactic disambiguation performed on the output of the morphological analysis method, which improves the precision of the analyser. The fourth section will list some issues that emerged from the implementation process. The final section will summarise the implementation of the workflow.

4.1. Implementation set-up

4.1.1. Pre-existing components

Before the MOR grammar development project, there existed a version of the CLAN software for the Light Warlpiri corpus, set up by Romauld Skiba (Max Planck Institute for Psycholinguistics) in 2004. This version contained a minimal set of MOR grammar files, including a lexicon file written by Carmel O'Shannessy which contained 6,846 word or morpheme entries (see Figure 18). The

number of transcription files in the Light Warlpiri corpus has increased since this lexicon file was created, meaning that extra vocabulary items were required to be added in the lexicon. The total number of items at the time of writing is 7,569.

```
_er    {[scat INF]}   "er"
_eye   {[scat N]}
_ga    {[scat TYP]}
_gka   {[scat CAS]}   "ngka:LOC:@1"
_goat  {[scat N]}
_good  {[scat ADV]}
_he    {[scat PRO]}   "2sgSub"
_i     {[scat EUP]}   "@1"
_iji   {[scat FOC]}   "@1"
_ik    {[scat CAS]}   "dat@1"
_iki   {[scat CAS]}   "dat@1"
_il    {[scat SUF]}   "then@1"
_ilki  {[scat SUF]}   "then@1"
_im    {[scat INF]}   "im:trans"
_ima   {[scat TYP]}
_imat  {[scat TYP]}
_imbat {[scat TYP]}
_imi   {[scat INF]}   "@1"
_in    {[scat INF]}
_ing   {[scat INF]}   "ing:prog"
_inga  {[scat CAS]}   "nga:LOC@1"
_ingi  {[scat CAS]}   "ngi:ERG@1"
```

Figure 18. The storage of single morphemes (with underscores) in the original lexicon file.

In the 2004 version of the Light Warlpiri corpus, there was a minimal version of the MOR grammar, including the A-RULES and C-RULES files. The A-RULES were set to a default setting where its processes were not enabled. The C-RULES file was edited by Skiba with a set of START rules to enable each part-of-speech of the Light Warlpiri lexicon file to be recognised as whole words (see Figure 19).

```
RULENAME: 1-start
CTYPE: START
 if
 NEXTCAT = [scat ADJ]
 then
 RESULTCAT = NEXTCAT
 RULEPACKAGE = {}

RULENAME: 2-start
CTYPE: START
 if
 NEXTCAT = [scat ADV]
 then
 RESULTCAT = NEXTCAT
 RULEPACKAGE = {}

RULENAME: 3-start
CTYPE: START
 if
 NEXTCAT = [scat ANA]
 then
 RESULTCAT = NEXTCAT
 RULEPACKAGE = {}
```

Figure 19. Original concatenation rules (C-RULES) file for the Light Warlpiri corpus.

However, as described in the previous chapter, the START rules do not analyse morphemes within transcript words. As a result, the lexicon file had a one-to-one correspondence between the entries and the transcript words. In the case of multi-morphemic words, the suffixes were required to be separated by a space and underscore to have their part-of-speech categories analysed by the START rules. This version of the MOR grammar could not analyse words without prior work to separate each stem and suffix in the corpus

transcripts. The original workflow also did not contain part-of-speech disambiguation (the POST program), meaning that there were recurring words in the corpus with an ambiguous gloss interpretation. The lack of morphosyntactic analysis also enabled incorrect glossing of morphemes (since they were not analysed in the context of their word stems) as well as ambiguous glosses for corpus words.

The sections below describe how this starting set of files was expanded upon to contain a workflow that analyses the morphosyntax of the Light Warlpiri corpus data.

4.1.2. Light Warlpiri corpus data

The Light Warlpiri corpus data were a set of files transcribed by O'Shannessy (2005, 2010, 2015), the data elicited from 2004 onwards. The data were divided into two sets: 1) the data elicited from storytelling (narrative set), and 2) the data elicited from spontaneous conversations (spontaneous set). In terms of transcription quality, the orthography in the SPON set was less consistent than in the narrative set. In terms of sample size, the spontaneous corpus data set amounted to 77,872 words, and the

narrative set amounted to 10,638 words, with a total of 88,510 words that were used to develop and apply the MOR grammar.

The design and implementation of the MOR grammar were based on both corpus data sets, to allow coverage of a broad range of Light Warlpiri vocabulary and word formation rules. There were some transcription errors and some variation present in both data sets, however, some common transcription errors were accounted for in the lexicon file in the form of alternate surface forms for a given word. For instance, many suffixes were marked with a hyphen attached to the preceding stem or suffix (e.g. karnta-pawu-ng), however, there were still a significant number of instances where this did not occur (e.g. “karntapawung” or “karnta-pawung”). As a result, the MOR grammar had to be adapted to inconsistencies in the corpus data, as well as the general vocabulary and morphosyntactic patterns.

4.2. Word analysis

As described in Chapter 3, the word analysis component of the MOR grammar (MOR analyser) processes the word formation patterns of the corpus data to determine the correct parts-of-speech of a word complex. The input of the MOR analyser is a given surface form of a

transcript word, and the output is a part-of-speech (POS) tag on the %mor tier with an accompanying English gloss. Adapting the morphological analysis component to the Light Warlpiri data contained three main steps: 1) identifying the types of words that were not recognised by the MOR analyser, 2) classifying unrecognised words by error type, and 3) addressing this issue through either editing the lexicon or by writing additional rules in the A-RULES or C-RULES files. Each step is discussed in the following sections.

4.2.1. Identifying unrecognised words

The first step was to identify the coverage of the MOR grammar on the corpus data sets, that is, the proportion of words that the MOR analyser could recognise in the transcripts. After each run of the mor command, the CLAN program listed the number of words that were not recognized by the MOR analyser for the input transcripts (see Figure 20).

```
Your transcript(s) have 10638 word(s) with 525 word(s) that MOR does not recognize.
These words are listed in a file labeled with the name of your input file that
ends in ".ulx.cex";
you will find this file in the same directory containing the input file.
Open the ".ulx.cex" file and triple-click to go to the place of each error.
After fixing the errors, please run MOR again.
If you choose to work with incomplete data, you can skip all these steps.
```

Figure 20. Output result of a MOR command for the narrative file set, listing the number of words that the MOR analyser had not recognised.

The words that were recognized in a set of files were contained in a separate file (.ulx.cex) output by the MOR program. This file contained all unrecognized transcript words with their file locations.

Figure 21. An unrecognised word file for the output of a MOR command run on the narrative file set.

If an unrecognised word occurred multiple times, it was repeated through the text document as many times as it appeared in the corpus data (see Figure 21). As there were often hundreds of unrecognised word instances appearing in this format, a Python script created by the author was run on the text document to list the frequencies of unrecognised words in the input data set (see Figure 22). The output of this script was used to gauge words in the corpus that required analysis by the MOR grammar, with the most

frequently unrecognised words prioritised over infrequently occurring words when integrating them into the lexicon or rule files.

```
1   nyan-nyang : 4
2   raiful-kurl : 4
3   wan-im : 4
4   aid : 3
5   karlanjirri : 3
6   karnta-karnta : 3
7   ngulaj : 3
8   oba : 3
9   olot : 3
10  pas : 3
11  raiful : 3
12  rarralykajirla : 3
13  tai-im-ap-im : 3
14  waif-ik : 3
```

Figure 22. Unrecognised words and their frequency in the narrative file set (output of unrecognized.py script)

There were two main reasons why a word was not recognised by the MOR grammar. The first reason was that the word, stem or affix was not in the lexicon file, meaning that an entry needed to be added to it. The second was that the word's morphological structure was not accounted for by the C-RULES file. If a word, word stem or affix was not contained in original lexicon, then it was added as an item with its morphosyntactic category and gloss. The second reason indicated that extra development was required in the analyser component of the MOR program. The source of each error had to be corrected to increase the overall coverage of the MOR grammar.

4.2.2. Enabling parts-of-speech analysis

For the MOR grammar to analyse words and stems, a START rule was written for each part-of-speech. Eleven START rules were written to accommodate for the Light Warlpiri part-of-speech categories. These rules did not cover bound morphology, such as case markers or pronominal auxiliary inflections. The START rules could cover whole words without suffixes (e.g. “ngapa”, noun for water) or the stem of a word with suffixes (e.g. the stem “karnta” in “karnta-pawu-ng”). Each character of the input string is processed from left-to-right, with each part of the string scanned in the lexicon file as a potential item. If a string or part thereof matches an item in the lexicon file, then its associated syntactic category ([scat]) is matched to all START rules in the C-RULES file that operate on the syntactic category.

Lexical surface: ngapa

Syntactic category: {[scat n]} (noun)

Gloss: “water@1”

RULENAME: n-start

CTYPE: START

if

NEXTCAT = [scat n]

then

RESULTCAT = NEXTCAT

RULEPACKAGE = {case, lk, suf-gen, focus, n-suf, p-encl, bound-pro, n-jarl, n-trans}

Figure 23. The noun “ngapa”, and the relevant n-start rule for nouns.

Step	String Position	Input String	Input string: match in lexicon?	Surface matches: associated categories	Applied C-RULES	C-RULES match?	Overall match outcome	Decision
0	|ngapa	‘ ‘	No	N/A	N/A	N/A	FAIL	NEXT; initiate step (1)
1	**n**|gapa	‘n’	No	N/A	N/A	N/A	FAIL	NEXT; initiate step (2)
2	**ng**|apa	‘ng’	Yes	[scat case]	None, no START rule	No	FAIL	NEXT; initiate step (3)
3	**nga**|pa	‘nga’	Yes	[scat case]	None; no START rule	No	FAIL	NEXT; initiate step (4)
4	**ngap**|a	‘ngap’	No	N/A	N/A	N/A	FAIL	NEXT; initiate step (5)
5	**ngapa**|	‘ngapa’	Yes	[scat n]	n-start	Yes	SUCCESS	**STOP; align part-of-speech tag (n) and gloss (‘water’) for entry *ngapa* with surface string ‘ngapa’**

Table 11. The string “ngapa” processed left-associatively by the MOR analyser.

Table 11 shows how the whole Light Warlpiri noun “ngapa” (‘water’) is processed by the MOR analyser. The surface string “ngapa” is matched to the lexicon file as a potential entry, but only the strings in steps 2, 3, and 5 have an associated lexicon item. Only the string in step 5 matches a START rule in the C-RULES file, so it is designated as the successful parse. While the strings ‘ng’ and ‘nga’ in steps 2 and 3 do have items in the lexicon file (the ergative and locative case markers, respectively), they do not have an associated

START rule, because they are suffixes. START rules operate on initial strings, that is, strings that occur at the start of a word. The 'ng' and 'nga' suffixes occur as bound morphemes to a stem, and therefore cannot be processed as initial strings by a START rule. As such, they do not have START rules, and the 'ng' and 'nga' strings are therefore skipped as potential parses in this position. The appropriate analysis is not applied to the string 'ngapa' until it is identified and aligned with the n-start rule. The operation statement RESULTCAT = NEXTCAT in the START rule, as seen in n-start (Figure 24) enables the specified grammatical category (in Figure 24, the noun category). In this example, the category corresponds to the grammatical category in the n-start rule, and therefore the string is recognised by the MOR analyser. The n-start rule also enables several rules to be activated after the noun stem, including the rule for case-marking (case) and the rule for nominal suffixes (n-suf). However, since the string "ngapa" in this instance does not contain any suffixes, it is output as a whole word with the category noun.

C-RULE name	Part-of-speech	Rule-package
n-start	Noun	nonfut, case, phon, lk, suf-gen, suf-foc-n, focus, n-suf, p-encl, n-jarl, noun-num, n-trans
pro-start	Pronoun	case, nonfut, namu, suf-gen, focus, n-suf, p-encl
qn-start	Question marker	qn-case
v-start	Verb	trans-suff, case-foc-v, v-tense, suf-gen, bat, v-prep
prev-start	Preverb	p-encl-2
det-start	Determiner	det-pl, det-case
dis-start	Discourse marker	case, gen-suf
anaph-start	Anaphora	suf-gen, case
num-start	Number	num-suf, num-case, num-dual
neg-start	Negation	n-suf

Table 12. All START rules contained in the Light Warlpiri MOR grammar.

Eleven START rules were devised for the Light Warlpiri parts-of-speech (see Table 12). Each start rule contained RULEPACKAGE statements that nested potential categories that could occur after the string enabled by the START rule. The rules contained in these RULEPACKAGE statements will be discussed in the next two sections.

4.2.3. Enabling suffixing

The MOR analyser C-RULES were designed to accommodate for the varying number of suffixes that can attach to a Light Warlpiri stem. Since the MOR analyser processes strings by identifying whole morphemes and their syntactic category, as well as their word positions and combinations with other morphemes, the C-RULES

had to accommodate for each possible suffixing combination in the Light Warlpiri data. In the Light Warlpiri corpus data, the maximum number of suffixes that occurred on a word was three, meaning that the C-RULES had to accommodate for up to four morphemes in a transcribed word. The other issue was to accommodate for the different parts-of-speech (e.g. nominal, verbal, auxiliary) and their possible stem-suffix category combinations. So, a set of rules were devised for the main parts of speech in Light Warlpiri (as further detailed in section 4.2.5).

To enable a suffix to occur immediately after a word stem, a set of NEXT rules had to be written for the applicable parts-of-speech. The NEXT rules were nested in the RULEPACKAGE statements of the START rules to enable the category of the word stem to combine with the category of the suffix, allowing the analyser to concatenate the two strings.

```
Trying rule case ...
word: yalyu–kurra
rest: –kurra
start: yalyu
start cat: {[scat n]}
current parse: blood@1
next: –kurra
```

```
        next cat: {[scat case]}
        next stem: allative@1

        trying clause/ if-then 1
        condition = CHECK STARTCAT {[scat OR n n:prop pro pro:qn dis pro:free
anaph num disj v:tense adv n free:pro]}
        condition is met
        condition = CHECK NEXTCAT {[scat case]}
        condition is met
        operation = COPY STARTCAT
        current result cat =
        case succeeded!

        Result cat: {[scat n]}
        current parse: blood@1-allative@1
```

Figure 24. The internal MOR analysis for "yalyu-kurra" (blood-allative), taken from the debugging CLAN file.

In Figure 24, the word "yalyu-kurra" has had its word stem "yalyu" identified as n|blood@1 by the preceding START rule, with the latter surface form "-kurra" needing to be identified. The c-rule [case] has matched "-kurra" with a surface form in the lexicon, where the syntactic category is [case] and its accompanying gloss "allative@1" (Warlpiri-derived allative marker). This rule then tags the whole word "yalyu-kurra" under the noun syntactic category ({[scat n]}) using the COPY STARTCAT operation. The resulting

parse “blood@1-allative@1” is the output of [n-start] and [case] combining their conditions in a left-associative process.

Input string: “ngapa-kurra-lk” (‘water-allative-then’)

(1) n-start: recognises “ngapa” lexical item and processes string “ngapa”; RULEPACKAGE statement includes ‘case’ rule

(2) case: recognises “-kurra” lexical item and processes string “ngapa-kurra” string, RULEPACKAGE statement includes ‘lk’ rule

(3) lk: recognises “-lk” lexical item and processes string “ngapa-kurra-lk”, no RULEPACKAGE statement

Output parse: “hit-im-bat-im” → v:tran|hit-im:tran-iter-im:tran

Figure 25. The nesting of three C-RULES to process the three-morpheme Light Warlpiri word "ngapa-kurra-lk".

For words with more than one suffix the rules to process each suffix were nested in one another using RULEPACKAGE statements (see the examples in Figure 25). To develop these sets of nested rules, the corpus data were examined to determine which morpheme combinations appeared in the corpus that were required to be enabled by the C-RULES. Compared to one and two-morpheme words, there were relatively fewer three- and four-morpheme words that appeared in the Light Warlpiri data. However, there was still a significant recurring presence of key words in this category, such as

nominals with a suffix and a case marker (e.g. *karnta-pawu-ng*) and verb complexes with transitive and iterative markers (e.g. *hit-im-bat-im*). As a result, these rules were essential to process this type of word. In the C-RULES file, 23 rules were written to enable morphemes after a word stem, and 29 rules were written as 'nested' RULEPACKAGE statement rules (to enable further suffixing).

4.2.4. Allomorphic rules

The A-RULES file allows the option to extend the lexicon to include variations of a stem or suffix. Some rules were written to account for variations in spelling for lexical items. For instance, the ARULE in Figure 26 accounts for the variation in spelling for nominal stems ending in "-er" or "-a" (e.g. "motha" for the item "mother").

```
RULENAME: a-ending
LEX-ENTRY:
LEXSURF = $Xer
LEXCAT = [scat n]
ALLO:
ALLOSURF = $Xa
ALLOCAT = LEXCAT, ADD [allo nErB]
ALLO:
ALLOSURF = LEXSURF
ALLOCAT = LEXCAT, ADD [allo nErA]
```

Figure 26. An ARULE for stems ending in "-er" or "-a".

The A-RULES use abstract surface rules to increase the storage of the lexicon. For the Light Warlpiri data, the A-RULES was used primarily for case-marker allomorphs or spelling variations. Compared to other languages that may have heavy conditions on allomorphic variation (such as German), the Light Warlpiri corpus did not require much work on this level of language.

```
RULENAME: ergative
LEX-ENTRY:
LEXSURF: -ngku
LEXCAT = [scat case]
ALLO:
ALLOSURF = -ng
ALLOCAT = LEXCAT, ADD [allo 1ngku]
ALLO:
ALLOSURF = -ngu
ALLOCAT = LEXCAT, ADD [allo 2ngku]
ALLO:
ALLOSURF: -ing
ALLOCAT = LEXCAT, ADD [allo 3ngku]
```

Figure 27. An ARULE for allomorphic variation in the Light Warlpiri ergative marker.

Figure 27 shows an example of an ARULE applied for the Light Warlpiri ergative marker *-ngku* and its very common allomorphic reductions *-ng*, *-ngu* and *-ing*. The LEXSURF and LEXCAT statements anchor the rule to the entry with the surface form "-ngku" and morphosyntactic category [scat case] within the lexicon.

The successive ALLOSURF and ALLOCAT statements each expand the lexical entry to include the surface strings "-ng", "-ngu" and "-ing", assigning system tags (e.g. [`allo 3ngku`]) for each one. As a result, the lexical entry *-ngku* also includes these expanded string forms, which are each aligned with the morphosyntactic category [`scat case`].

4.2.5. Example rule applications

MOR grammatical category	Light Warlpiri part-of-speech
[scat n]	Noun
[scat case]	ablative, allative, comitative, dative, ergative, evitative, locative, perlative, possessive
[scat suf]	diminutive, emphasis, particle, perlative, nominal suffix, topic

Table 13. System grammatical categories for the Light Warlpiri nominal part-of-speech.

A set of rules was devised for suffixes following the Light Warlpiri noun part-of-speech. These rules included the enabling of noun stems ([scat n]) along with case and suffix markers (including some spelling variations in the A-RULES file). The order in which certain Light Warlpiri suffixes occur was one factor to take into account

when writing the nominal-suffixing rules. For instance, the ergative case marker in Light Warlpiri occurs as the terminal suffix in a given word, regardless of the number of suffixes on the word stem (see examples 1-3 below). As a result, it cannot have another suffix bound to it.

(1) karnta-**ng**

woman-ERG

(2) karnta-pawu-**ng**

woman-DIM-ERG

(3) motorbike-kurlu-**ng**

motorbike-COM-ERG

For this phenomenon, the Light Warlpiri ergative marker lexical entry had an additional category tag [case erg] in its syntactic category field. Then, a C-RULE was written to enable the ergative marker to attach to a stem or nominal suffix, but without RULEPACKAGE statements. As a result, the analyser blocked the possibility of a suffix attaching to an ergative marker. This 'blocking' type rule ruled out certain instances of category over-generation by the analyser.

MOR grammatical category	Light Warlpiri part-of-speech
[scat v], [scat v:tran], [scat v:intran]	Verb
[scat v:deriv]	Transitive marker (‘-im’, ‘-um’)
[scat v:infl]	Progressive (e.g. ‘-ing’), iterative (‘-bat’)
[scat v:tense]	Non-past, past, imperative
[scat suf:dir]	Directional (e.g. ‘-dan’, ‘-ap’)

Table 14. Verb complex grammatical categories and their parts-of-speech in Light Warlpiri.

A set of rules was devised for the Light Warlpiri verb part-of-speech. The starting stem for this part-of-speech was categorised as [scat v], with the derivational, inflectional and tense suffixes enabled as possible bound suffixes to this stem. The prepositional and iterative markers were enabled to occur on the stem or after the transitive marker [scat v:deriv].

Lexicon category	Light Warlpiri part-of-speech
[scat pro]	Pronominal (‘a’, ‘wi’, ‘yu’ ‘i’ ‘de’)
[scat v:aux]	Non-past (‘-m’), future marker (‘-rra’), desiderative marker (‘-na’)

Table 15. The pronominal grammatical categories and their parts-of-speech in Light Warlpiri.

The C-RULES were also applied to the Light Warlpiri auxiliary paradigm. A START rule enabled the pronoun part-of-speech to attach to a verbal auxiliary NEXT rule, allowing the tense, aspect and mood inflections to be recognised in the data. In summary, the MOR analyser was adapted to the morphological structures of the Light Warlpiri language. The analyser was first expanded to cover whole parts-of-speech, then word suffixing patterns, including specific suffixing combinations.

4.3. POST program disambiguation

The POST program disambiguates the output of the MOR analyser where a corpus word is analysed as more than one gloss, as explained in the previous chapter. For each instance of an ambiguous analysis, each gloss is attached to another on the %mor tier with a caret (^) symbol (see examples in Figure 28).

1. Input string: “lap-kurra”

   n|lap-allative@1^n|lap-dat@1-PL

   Correct gloss: n|lap-allative@1

2. Input string: “i-m fly walya-wana”

pro|i:3sg-m:NONFUT **v:intran|fly^n|fly** n|ground@1-perl@1

Correct gloss: v:intran|fly

3. Input string: “kuja na”

   aux|relative^dis|thus@1 dis|foc:now .

   Correct gloss: dis|thus@1 dis|foc:now

Figure 28. Examples of ambiguous glossing by the MOR program.

Since ambiguous analyses affect the accuracy and readability of the MOR analyser output. The POST program was developed to reduce the number of glosses to one gloss per word.

4.3.1. POST rule applications

The rule-based component of the POST program applies definitive part-of-speech disambiguation to selected words in the corpus, based on a given syntactic context.

```
prep|for pro|3S.O^pro|i:3sg-m:NONFUT => prep|for pro|3S.O
```

Figure 29. A PREPOST rule in the Light Warlpiri POST program.

Figure 29 shows a POST rule application to the Light Warlpiri data. It refers to the word-sense ambiguity between "im" as a third person singular subject pronoun with non-future marker (i-m) and "im" as a third person singular object pronoun. As these two words have the same surface string and part-of-speech category (pronoun), the MOR analyser always output the two glosses together with a caret (`prep|for pro|3S.O^pro|i:3sg-m:NONFUT`). For this, a PREPOST rule was devised such that if "im" appeared after a preposition, then it would be interpreted as the third person singular object pronoun (pro|3S.O). This allowed the analyser to successfully rule out the inflected subject pronoun 'i-m' to be interpreted as such by the MOR program when it occurred after a preposition. In the Light Warlpiri MOR grammar, three POST rules were implemented on frequently occurring word-sense ambiguities in the corpus data.

4.3.2. POSTTRAIN

Four transcript files from the spontaneous and narrative sets (two files from each set), a total number of 4,243 words, were used as training data for the probabilistic model of the POSTTRAIN program. These files were glossed using the mor command before they were disambiguated by hand to produce a second comparison tier (%trn) (see Figure 30).

```
*A91:   hitim is waif •
%mor:   v:tran|hit-im:tran^v:tran|hit-3S.Obj pro|3sg:poss n|wife
%trn:   v:tran|hit-im:tran pro|3sg:poss n|wife
%eng:   he hit his wife
*A91:   kurdu-kurl wen im oldim •
%mor:   n|child@1-com@1 qn|when pro|3S.Obj^pro|i:3sg-m:NONFUT
      v:tran|hold-im:tran^v:tran|hold-3S.Obj^v:tran|old-im:tran^n|old-3S.Obj
%trn:   n|child@1-com@1 qn|when pro|i:3sg-m:NONFUT v:tran|hold-im:tran
```

Figure 30. A POSTTRAIN training file with the %mor and %trn tiers. The POSTTRAIN database compares the two tiers, %mor and %trn to generate probabilistic rules for disambiguation.

The accuracy of the POSTTRAIN database was significantly reliant on high word coverage by the MOR program. The starting MOR coverage on the POSTTRAIN file sets was 96.5%, and as a result, the MOR grammar lexicon had to be updated with new vocabulary to increase the reliability of the POSTTRAIN program. After three cycles of training the POSTTRAIN database, the model was substantial enough to provide predicted glosses for some unrecognised words in the corpus. This model was integrated by CLAN into the post command, performed in each glossing cycle after the mor command.

```
qn * pro ==>  1 / [nth:~, nth_cnt:1] [1st:16-qn, 2nd:28-pro] /
neg * prep ==>  1 / [nth:~, nth_cnt:1] [1st:36-neg, 2nd:37-prep] /
v:intran * n ==>  1 / [nth:~, nth_cnt:5] [1st:18-v:intran, 2nd:8-n] /
v:intran * adv ==>  1 / [nth:~, nth_cnt:1] [1st:18-v:intran, 2nd:4-adv] /
prep * pct ==>  1 / [nth:~, nth_cnt:2] [1st:37-prep, 2nd:1-pct] /
det * v:tran ==>  1 / [nth:~, nth_cnt:1] [1st:13-det, 2nd:21-v:tran] /
v:intran/v:intran—on * pct ==>  1 / [nth:~, nth_cnt:2] [1st:19-v:intran—on, 2nd:1-pct]
/
```

Figure 31. An excerpt from the list of rules of the POSTTRAIN model for the Light Warlpiri corpus data.

Figure 31 shows a subset of rules in the POSTTRAIN model for the Light Warlpiri corpus data (retrieved using the CLAN `postlist +r` command) after running the model on some training data. The tuple on the left-hand side of the rule (e.g. `qn * pro`) refers to the instance where one part-of-speech comes immediately after another in the corpus data (for this instance, where a pronoun comes immediately after an interrogative). The first item in the right-hand side of the rule `([nth:~, nth-cnt:1])` refers to the number of times this instance occurs in the training data (which is 1 occurrence for the `qn*pro` tuple) and the second item (`1st:16-qn, 2nd:28-pro]`) is the number of times the morphosyntactic category occurs overall in in the training data (16 times for the interrogative part-of-speech, 28 times for the pronoun part-of-speech). The information shown in the POSTTRAIN rules set show how the different parts of the training data contribute to the binary rules Markov model.

4.4. Issues

In the original lexicon file, some lexical entries contained surface forms with more than one morpheme.

(3) ngularra	{[scat anaph]}	“who:which-PL
(4) ngula	{[scat anaph]}	“who:which”
(5) rra	{[scat suf]}	“pl”

For instance, examples 3, 4 and 5 above show how the word form *ngularra* ‘anaphora-PL’ was contained in the same lexicon file as the separate morpheme entries “ngula” (anaphora) and “rra” (plural marker). Since the “ngularra” entry overlaps with the combination of “ngula” and “rra”, the output of a “ngularra” input string would produce two parses, producing an ambiguous result (see Figure 32 for an example of this output).

```
anaph|whowhich@1-pl@1^anaph|whowhich@1-PL
```

Figure 32. An ambiguous gloss as a result of overlapping lexical entries.

The presence of overlapping entries was the result of having an older version of the Light Warlpiri lexicon that conflicted with newer versions. This meant that the lexicon file either had to be edited to only contain whole words, or word stems and suffixes, or the

resulting parses had to be disambiguated by the POST program to enable one correct parse. This issue produced some ongoing work in editing the lexicon file and improving the POST program to resolve these conflicts.

Another issue was that there were instances of suffixes occurring throughout the corpus data that were separated from their stems. These occurrences were the result of human error while transcribing the data. The separation of the suffix from the stem with a space meant that the C-RULES could not recognise the suffix, as it was not enabled to occur as a standalone word. This issue could be resolved in two ways: by 1) implementing a separate script on the corpus transcripts that replaced the spaces between suffixes and their stems with a hyphen, or 2) to enable suffixes to occur as standalone words in the corpus data. The second strategy was tested on the corpus data by writing a START rule to enable suffixes as a 'stem' category. However, this strategy can lead the POSTTRAIN component to weigh suffixes too heavily in its model. As a result, the model categorised suffixes in the data that were incorrect classifications, which is further explained in Chapter 5. The first strategy (pre-processing with a script separate to the MOR program) would not affect the POSTTRAIN model and therefore might be more beneficial for the workflow.

4.5. Summary

This chapter outlined the implementation of the morphological analyser (MOR program) and part-of-speech disambiguation tool (POST program) on the Light Warlpiri corpus data. The MOR program was applied to the morphological structures of the data, before the rules-based and statistical components of the POST program were adapted to it. Some issues in implementation relating to the lexicon and the corpus data transcription were noted. In the next chapter, I will describe and discuss the results generated from this implementation.

5. Evaluation

This chapter evaluates the output of the MOR program on the Light Warlpiri corpus data. The first section outlines the grammar's performance on several aspects of the data. The second section describes the error analysis undertaken on the output. The third section summarises the MOR grammar output, with significance placed on key results.

5.1. Performance

This section outlines how the MOR program performed on the Light Warlpiri corpus data. The selected performance measurements for this chapter are coverage and accuracy, since these variables have a direct impact on the effectiveness of the program.

5.1.1. Coverage

The output of the MOR grammar was measured for coverage of the corpus data, that is, the proportion of accurate word recognition by the analyser. The coverage was measured repeatedly during the development of the MOR grammar, with each run of the mor command outputting this variable. The coverage was 95.7% for the spontaneous transcript words (n = 80 914), and 94.05% for the story

transcript words (n=10 538). A large proportion of the effort towards improving coverage of the corpus data was directed at highly frequent errors appearing in multiple transcripts. If such words were unrecognised, the lexicon was updated to accommodate them.

5.1.2. Accuracy

The output of the MOR grammar was also measured for accuracy, that is, the rate of correctly analysed words out of the total number of analysed words in the corpus. For this measurement, the MOR output of a sample of corpus words (n=1 571) from both the spontaneous and story data was evaluated and manually corrected by Carmel O'Shannessy, a primary researcher of Light Warlpiri. The resulting corrected sample was used as a 'gold-standard' data set. In this data set, the coverage rate of the MOR grammar was 98.44% and the error rate was 7.89% (124 errors out of 1 571 words). The comparison of the MOR output with the gold-standard glosses was used to provide insight into recurring errors that the MOR program produced. These errors are further explained in section 5.2.

5.2. Error analysis

This section will outline the common types of errors that occurred in the output of the MOR program. The most common errors were 1) non-recognition, 2) overapplication of the C-RULES, and 3) incorrect part of speech classification, with each assigned a sub-section below.

5.2.1. Non-recognition

In the accuracy measurement data sample, failure of the MOR grammar to recognise a word accounted for 34 of the 124 errors (27.4%). Many of the unrecognised word forms were spelling variants of words that did exist in the lexicon file (e.g. "ed-aik" for the lexical item "headache", "jeinj" for the lexical item "change", "-pura" for the lexical item *-purda* '-want'). These alternate forms were added to the lexicon to account for slight variations in transcription orthography in the corpus data, to adhere to Kriol orthography (e.g. "sliip" instead of the standard English spelling "sleep"), for instance. If some word form variations had a regular pattern, such as words ending in "-er" spelt with "-a", the A-RULES file was updated to add this variation to the lexical item's surface.

5.2.2. Overapplication of C-RULES

In some cases, the C-RULES correctly classified the overall part-of-speech of a word but generated an overly complex analysis for its suffixing pattern. For instance, the word *wiri-jarlu* 'big-very' was classified correctly as a nominal but generated four morphemes in its interpretation instead of two ("n|big-FOC-1SG-ERG") where the correct gloss is "n|big-very". In the overall corpus, this type of error was found in 11 of the 124 errors (8.87%).

While building the C-RULES, it was sometimes difficult to predict whether certain rules would increase coverage by enabling more complex morpheme combinations to occur (therefore recognising words with three or more morphemes), or decrease the precision of the analyser by over-applying these rules to simpler words, and subsequently generating incorrect glosses. One solution was to enable more complex rule applications to open-class lexical words, that is, verb complexes and nouns, and to reduce the applications for closed class parts of speech (pronominals, conjunctions, disjunctions, discourse markers). As a result, there would be no over-application of rules for words that seldom contained more than two morphemes.

The C-RULES file was modified to block certain morpheme combinations from occurring in the Light Warlpiri data. For example, a rule that allowed pronouns to occur on words was removed from the Light Warlpiri grammar to prevent analysis of bound pronouns on Light Warlpiri words (e.g. the "i" ending of "mayi" being interpreted as "1sg").

5.2.3. Incorrect part-of-speech classification

A significant number of accuracy errors were due to incorrect part-of-speech classification or gloss, with this kind of error accounting for 79 of the 124 errors (63.71%) in the corpus sample. Some errors were due to transcription inconsistencies, such as missing hyphens.

For example, there were instances of the form “watiya wana” classified as “n|tree v|want:to” when the correct reading was “n|tree-PERLATIVE”. In this example, the “-wana” suffix was not attached to the nominal stem “watiya” in the surface form of the word, so the “wana” form was interpreted as the desiderative verb.

Part-of-speech	Number of tagged words in accuracy sample	Number of incorrect classifications	Error rate (%, incorrect classification)
Adverbial	10	0	0.00
Anaphora	5	0	0.00
Article	2	2	100
Conjunction	72	0	0.00
Determiner	79	5	6.3
Directional suffix	1	0	0.00
Disjunction	0	0	0.00
Kinship noun	3	0	0.00
Negation	11	0	0.00
Noun	382	7	1.83
Number	54	1	1.85
Preposition	20	1	5.00
Pronoun	264	7	2.65
Proper noun	23	0	0.00
Suffix	20	5	25.00
Intransitive verb	149	19	12.75
Modal verb	5	0	0.00
Transitive verb	188	36	19.14
Auxiliary verb	2	0	0.00
Interrogative	45	1	2.22
Discourse marker	185	26	14.05
Not recognised	51	N/A	N/A

Table 16. Error rate of MOR program on Light Warlpiri data by part-of-speech.

Table 16 shows the accuracy rate by part-of-speech in the sample data. This table sections the data by their classified part-of-speech

category, along with their number of incorrect classifications (for instance, a word is tagged as a noun when it should have been tagged as a verb). Nine morphosyntactic categories had no error instances in their accuracy measurement. What these categories had in common was their low morphological complexity, many of their instances being closed-class words with one morpheme. The categories with the highest error percentages were in the suffix, transitive verb and discourse word categories. All suffix category classification errors were due to the incorrect analysis of the word *wati* ('man') as a suffix (*-wati*, 'PL') by the POSTTRAIN program. This error instance shows how the POSTTRAIN model may have weighed instances of potential suffix classifications over potential noun classifications in its model.

Some incorrect classifications in the transitive verb category were due to errors in the lexicon (e.g. *kip* "keep" classified as a transitive verb instead of an intransitive verb in the lexicon) while other incorrect classifications were due to incorrect disambiguation by the POSTTRAIN model. One example was the word "it" being classified as an alternative spelling for the transitive verb "hit" instead of a third person object pronoun. The POSTTRAIN model appeared to weigh transitive verbs heavily in its disambiguation, with a significant number of incorrect classifications being tagged as the transitive verb category instead of its correct category. A similar

phenomenon occurred in the discourse category classifications, with an over-representation of words being incorrectly tagged as discourse markers by the POSTTRAIN model. These results show that the quality of the training data in the POSTTRAIN program was important for the MOR program's accuracy on the Light Warlpiri data.

In some instances, certain words that could be interpreted as more than one morphosyntactic category were reduced to just one item in the lexicon file if they could be repeatedly classified in one category, and rarely another. For example, the word "bin" was listed as both a noun and a verbal auxiliary in the lexicon, but the former definition was omitted from the lexicon since the latter item was used significantly more frequently in the corpus data.

5.3. Results summary

This chapter outlined the resulting output of the MOR grammar when its performance was evaluated. The output was evaluated for coverage and accuracy, with insights from these measurements informing the improvement of the grammar. The most common cause of lowered accuracy was incorrect glossing classification by either the MOR or POST program, with non-recognition of words and over-application of concatenation rules also being contributing causes. To solve these errors, the lexicon file, concatenation rules

and the POST program were updated to account for variations in the spelling of corpus words, to reduce the number of rules applied to certain word types, or to improve the probabilistic model for part-of-speech disambiguation. The next chapter will provide a further discussion of these findings.

6. Discussion

This chapter will discuss the findings of the Light Warlpiri MOR grammar project. The first section will discuss the findings of the morphological analysis component of the MOR program. The second section will discuss the morphosyntactic analysis program as a workflow. The third section will compare the workflow to other methods of morphosyntactic analysis. The fourth section will describe the limitations of the study. The final section will summarise the discussion chapter, with emphasis placed on significant findings.

6.1. MOR program word analysis and disambiguation

There were some differences in the morphosyntactic analysis performance according to the data type in the corpus. The coverage on the spontaneous subset measured at 95.7%, with the coverage on the story-telling subset measured at 94.05%. These coverage rates are close to one another, but the number of words in the spontaneous subset was significantly higher (n=80 914) than the story-telling subset (n=10 538). In terms of unique words to the overall number of words, the story-telling dataset had a higher unique word to token ratio (0.186) than the spontaneous dataset (0.067). This means that

there was significantly more word repetition in the spontaneous dataset and more novel words in the narrative dataset. The presence of more repetition in the spontaneous dataset meant that there were fewer words to add to the lexicon for these transcripts, whereas there were more words to process in the narrative dataset. The narrative dataset included many more utterances that used argument and peripheral case-marking than in the spontaneous set. This discrepancy could be explained by the different contexts in which the participants spoke. For example, the participants in the narrative set had an incentive to use case-marking when talking about subjects in the storybook pictures, as their descriptions were referring to people and objects in the third person. The differences in unique word to token ratio and morphological complexity between the narrative and spontaneous data affected the MOR program's coverage. Finally, while the vast majority of corpus data were in Light Warlpiri, some of the coverage rate was affected by the presence of code-switching into classic Warlpiri. These words could not be enabled by the rules of the Light Warlpiri MOR program, since their morphological template is different from that of Light Warlpiri.

On the level of rules-based morphological processing, the MOR program could be enabled on a wide variety of Light Warlpiri corpus words, as seen in Chapter 5. Therefore, the rules-based enabling of

morphological compounding was successful for multiple aspects of the Light Warlpiri corpus data. I would cite two main reasons for this straightforwardness. The first is that the project started with a comprehensive lexicon contributed by other researchers (O'Shannessy and Skiba, 2004) that covered a significant number of whole words in the corpus. However, this reason does not account for the high coverage of multi-morphemic words in the corpus. For this, I would cite the ability for the corpus words to be processed with a relatively small set of morphological categories as a contributing reason for the straightforwardness of the C-RULES application. In languages with increased morphological complexity, there requires a large set of morphosyntactic categories (including categories within morphemes) to process multiple elements in a morpheme. However, in Light Warlpiri, the regularity and low level of complexity in morphological structures meant that the grammatical categories in the lexicon did not have to be altered significantly to result in correct output for a large number of Light Warlpiri words. The underlying mechanisms of the MOR program (two-level morphological analysis, left-associative direction) were relatively straightforward to implement on the morphosyntactic structures of the Light Warlpiri data. Light Warlpiri's vocabulary is derived from two different language sources (Warlpiri and AE/Kriol); however, all vocabulary items are stored in the same manner in the lexicon: as a

word string, with a part-of-speech category and gloss. This means that the analyser can handle a mixed language lexicon without difficulty. In addition, the Light Warlpiri analyser uses a relatively straightforward annotation schema that does not require extensive modification to be processed by the MOR program.

The POST disambiguation component improved the accuracy and readability of the morphologically analysed Light Warlpiri data, however, there were still improvements to be made to its probabilistic component. The rules-based disambiguation (POST rules) was valuable for highly frequent ambiguous tokens, such as "i-m" and "im". Integrating these rules into the POST program was therefore a reliable way of implementing disambiguation for tokens that were marked as likely to show up as ambiguous in the data from the outset. The intended value of the probabilistic component (POSTTRAIN) was to disambiguate the ambiguous tokens where their syntactic contexts would have been hard to describe by hand. This component varied in its accuracy. While there were instances where the POSTTRAIN proved useful (see example 29), there were other instances where it incorrectly classified the output (see example 30).

29) Correct disambiguation:

Utterance:

*A21: an im sliip dat wati murru-murru

Ambiguous:

%mor: conj|an pro|3SG.O^pro|i:3sg-m:NONFUT

v:intran|sleep^n|sleep det|that n|man n|sick@1 .

Disambiguated:

%mor: conj|an pro|i:3sg-m:NONFUT **v:intran|sleep** det|that suf|pl@1

n|sick@1 .

30) Incorrect disambiguation:

Utterance:

*AXY: yuwayi bat Lajamanu stail Lajamanu stail

Ambiguous:

%mor: dis|yes@1 **disj|but^n|bat n:prop|Lajamanu** n|style

n:prop|Lajamanu n|style

Disambiguated:

%mor: dis|yes@1 **n|bat (correct: disj|but)** n:prop|Lajamanu n|style

n:prop|Lajamanu n|style

An additional difficulty with these incorrect instances of disambiguation is that the correct ambiguous option is deleted from the %mor tier, and the mor command (run without the POST component) would have to be run on the transcript again to retrieve the alternative glosses. On a practical level, this is tedious, and it calls into question the value of a POSTTRAIN analyser that has some unreliable output in the MOR program workflow, as opposed to disambiguating the MOR word analysis output by hand. However, with some additional training data added (with increased coverage on this data), the POSTTRAIN component would improve its accuracy. It also remains an essential part of the workflow as it improves the quality of the MOR program output in instances that are hard to account for with the rule-based method.

6.2. Evaluation of workflow: practical elements

The results of this thesis show the outcome of the MOR program implementation for the Light Warlpiri language. The factors that made this implementation more straightforward were:

a. the existence of a pre-existing lexicon for the language project,

b. using a frequency-based strategy when building the coverage for the corpus data, by prioritising recognition for the most frequent words or structures over rarer words or structures, and

c. adapting the program to a language that was not overly morphologically complex and had a left-associative suffixing pattern.

The pre-existing Light Warlpiri lexicon in the corpus project saved a significant amount of time in the development of the MOR analyser. Although the lexicon required 723 additional entries to cater for the large number of unanalysed corpus transcripts added to the data sample, there were a large number of lexical items already in the MOR program that were considered highly frequent Light Warlpiri vocabulary. However, there were difficulties attached to this lexicon file. As stated in the fourth chapter, the lexicon contained conflicting entries that reduced the precision of the analyser, such as a surface item in its entire inflected form being stored in the same file as the inflected form's word parts. These conflicting entries were due to the older lexicon relying on a one-to-one correspondence between corpus words and their morphosyntactic analysis, instead of a morpheme-by-morpheme analysis. Therefore, while the pre-existing lexicon helped reduce time to implement the analyser, there was

nonetheless time dedicated to modifying the lexicon to suit the mechanisms of the analyser. Overall, a recommendation for a project involving a rules-based morpheme concatenation program would be to ensure that the lexicon stored at most one entry for each word or word part to allow the program to concatenate the morphemes neatly.

One significant practical disadvantage of the MOR program is that it cannot perform partial word analysis. For instance, if a word had a stem that was contained in the lexicon, but a suffix that was not contained in it, then the analyser tagged the whole word as unrecognised, despite the word stem being recognised. For instance, there were words in the corpus data that were unrecognised by the MOR program since they had unknown suffixes attached, but its stem (such as a verb or nominal stem) would have been categorisable by part-of-speech. On one hand, there is an advantage to this system limitation, since it could prevent incorrect analyses based on partial information of the word. For instance, the beginning of a string may be incorrectly recognised as a noun by an analyser that allows partial analysis, but the unknown suffix in the rest of the word could be a derivational morpheme that changes the whole word part-of-speech. If the program still glossed the whole word as a noun (despite the unknown suffix changing it to a verb), this would enable an incorrect analysis, therefore lowering the

accuracy of the program. On the other hand, there is a closed set of nominal and verbal suffixes in Light Warlpiri which could function as 'flags' for certain parts-of-speech. Changing the MOR program to facilitate partial analysis based on suffixes would mean that the program itself would be modified using an external tool. The issue here is that the left-associative grammar requires all morphemes of a word to be identified to process the word sequentially. This is an inflexibility built into the CLAN software and cannot be resolved by modifying the workflow using this system – needs external modification or even a different underlying mechanism to the program.

Another limitation of the MOR program is in its inflexibility with transcription errors. While some common transcription error surface forms were added to the lexicon (e.g. 'fihgt' for the verb stem 'fight'), it is impossible to employ this strategy for all possible transcription errors. An additional script that checked for spelling transcription errors (and changed them accordingly) in the workflow would be desirable to improve the performance of the MOR analyser. In general, the high accuracy of the MOR analyser on a wide range of the corpus data indicates that it is implementable on a large-scale. In the CHILDES network, the most highly developed MOR grammar currently is the one for English, achieving 99.18% accuracy in tagging for adult native speaker language. This version of the MOR

grammar has been developed over years (since 2000), with suggestions for improvement by users informing the grammar's improvement with each new version. This accuracy indicates that the MOR analyser can be a highly reliable system when developed fully. The challenges in adapting the analyser to future corpus data would include the requirement to keep adding vocabulary to the lexicon (for unknown words) and managing recurring transcription errors.

There are numerous tools employed by language projects morphosyntactic analysis, including minority language projects with sparse amounts of transcribed data. The introductory chapter mentions several other tools available for transcribing and processing corpus data, such as ELAN, FLEx, and Toolbox (or Shoebox). Like CLAN, these tools host their own advantages and disadvantages on the practical front. Toolbox, for instance, implements an automated interlinear glossing method that segments text lines into morphemes using a lexicon. This method processes words from the outside in (prefixes and suffixes before roots) and prioritises large substrings to short substrings (e.g. "throughout" is found over "though + out"). Where there is more than one parse possible, the user is prompted with an interface dialogue to resolve the ambiguity. This method relies on more interaction on the user interface than the MOR program and

therefore may be more intuitive to implement for a linguistic worker. However, there is still a large amount of manual work to be done in organizing the lexicon files and databases, and there is no 'training' component for the automatic segmentation program. Nevertheless, the ease of the Toolbox program for the user to navigate the glossing program (as opposed to dealing with technical files) may be a beneficial aspect to this workflow that does not exist for the CLAN program.

6.3. Overall findings

In general, the MOR program was shown to be implementable for a significant amount of the Light Warlpiri data. The results of this study showed how the MOR program applied to a sample of Light Warlpiri data (n = 88 506 words). The accuracy measurement was relatively high for the Light Warlpiri MOR analyser (93.12%). This measurement included data from both narrative and spontaneous data sets. The relevant files for the MOR analyser are discoverable on a public Github repository[1] used for version control of the project.

The sample size of the accuracy measurement was limited (n=1 803), given that manual verification of glosses takes time on the part of

[1] https://github.com/ginawelsh/mor-program-lw

the evaluator. One of the limitations in the accuracy measurement is that the corpus data included a small amount of code-switching language in the use of classic Warlpiri verbs. This reduces the overall coverage of the results. It was difficult to ascertain how much of the data were classic Warlpiri verbs. One solution to this would be to integrate the morphosyntactic rules of the classic Warlpiri data into the MOR grammar. However, this would involve ensuring that the Light Warlpiri word analysis and disambiguation do not overlap with those of classic Warlpiri grammar, a task which was not explored in the topic of this thesis.

6.4. Summary

This chapter discussed the findings of this thesis. The word analysis and disambiguation components of the MOR program were discussed, including how the data type affected the results and how the POSTTRAIN component performed on the data. The practical aspects of the MOR program were discussed, including some of its limitations and a comparison to a similar corpus tool. Some of the limitations of the study included the small sample size of accuracy data as well as the presence of classic Warlpiri code-switching data. However, the thesis findings provided some insight into the MOR program implementation on the Light Warlpiri corpus data.

7. Conclusion

This thesis described and evaluated the implementation of a method of automatic morphosyntactic analysis for Light Warlpiri corpus data. The method involved a rule-based method of word analysis and a hybrid method of word disambiguation.

Chapter 1 introduced the topic of automatic morphosyntactic analysis. It included a brief literature review of methods, tools and related projects within this discipline that involve the analysis of language data, before stating the research objectives of this thesis. Chapter 2 described the Light Warlpiri language, including sociolinguistic background, its language sources Warlpiri, English and Kriol, and key properties of its morphosyntax. It also provided formal templates for suffixing in the Light Warlpiri nominal, verbal and auxiliary paradigms, as well as the part-of-speech category set for the Light Warlpiri language data. Chapter 3 outlined the computational workflow of the MOR program, illustrating how the underlying mechanisms of the analyser contributed to the workflow's word recognition and disambiguation. These mechanisms included two-level morphology, left-associative grammar, and a binary rules Markov model of disambiguation. Chapter 4 showed how these underlying mechanisms were applied to the Light

Warlpiri corpus data. The application of word-level analysis was applied to the Light Warlpiri nominal, verb and auxiliary paradigms. Rule-based disambiguation was applied to frequently occurring ambiguous items in the data, as well as probabilistic disambiguation using a model created by the POSTTTRAIN component of the MOR program. Chapter 5 showed some performance measurements of the MOR program applied to the Light Warlpiri data. The MOR output was measured for coverage and accuracy and error analysis was applied for the corpus data. The most common errors in the resulting output were lack of word recognition, over-application of the concatenation rules, and incorrect classification by the disambiguation component. Chapter 6 discussed the overall findings of the thesis. This discussion included how the type of corpus data affected the performance of the MOR analysis, what the practical limitations of the MOR program were, and what aspects of the implementation made the development process more straightforward.

Overall, the method was implementable on a significant proportion of Light Warlpiri data, with the word analysis enabled on different morphosyntactic categories, and the disambiguation applied on frequently occurring ambiguous items using contextual rules and training data. This type of automatic morphosyntactic analysis

extended the corpus data to include morphosyntactic annotation. This type of information enriches the corpus data for the researcher and provides grounds for further computational work on the corpus. In the larger context, the thesis contributes insight into how automatic morphosyntactic glossing can be applied in a minority language using a software tool that has been applied to numerous majority languages worldwide.

Bibliography

Brill, E. (2002). *A Simple Rule-based Part of Speech Tagger.* Proceedings of the Third Conference on Applied Natural Language Processing, Trento, Italy.

Butcher, A. (2008). *Linguistic aspects of Australian Aboriginal English.* Clinical Linguistics & Phonetics, 22(8), 625-642.

Dandapat, S., & Sarkar, S. (2006). *Part of Speech Tagging for Bengali with Hidden Markov Model.* NLPAI ML workshop on part of-speech tagging and chunking for Indian languages.

Dandapat, S., Sarkar, S., & Anupam, B. (2007). *Automatic Part-of Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario.* ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions.

Dickson, G. F. (2015). *Marra and Kriol: the loss and maintenance of knowledge across a language shift boundary (Doctoral dissertation).* Canberra: Australian National University.

ELAN (Version 5.9) [Computer software]. (2020). Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive.

FieldWorks Language Explorer (Version 9.0.12) [Computer software]. (2020) SIL International.

Greene, B., & Rubin, G. (1971). *Automated Grammatical Tagging of English.* Technical Report, Department of Linguistics, Brown University, RI.

Hale, K. (1983). Warlpiri and the Grammar of Non-Configurational Languages. *Natural Language and Linguistic Theory*, *1*, 5–47.

Hausser, R. (1986). NEWCAT: Parsing Natural Language Using Left-Associative Grammar. Springer-Verlag, Berlin. 10.1007/3-540-16781-1_2.

Hausser, R. (1999). *Foundations of computational linguistics: Man-machine communication in natural language*. Springer.

Kashket, Michael Brian. (1987). *A Government-Binding based parser for Warlpiri, a free-word order language.* M.S. thesis, Department of Electrical Engineering and Computer Science, M.I.T. 165pp.

Koskenniemi, Kimmo. (1983). *Two-level morphology: A general computational model for word-form recognition and production*. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Laughren, M. (1999). *Constraints on the Pre-auxiliary Position in Warlpiri and the Nature of the Auxiliary*. Proceedings of the 1999 conference of the Australian Linguistic Society.

Laughren, M., Hoogenraad, R., Hale, K. L., Granites, R. J., & Institute for Aboriginal Development (Alice Springs, N.T.). (1996). *A learner's guide to warlpiri: Tape course for beginners: Wangkamirlipa warlpiriki*. Alice Springs, N.T: IAD Press.

Laughren, M. (2002). *Syntactic constraints in a 'free word order' language*. In: Mengistu Am- Berber and Peter Collins (eds.) Language Universals and Variation. Westport CT: Praeger Publishers.

Leech, G., Garside, R., & Atwell, E. (1983). *The Automatic Grammatical Tagging of the LOB Corpus.* ICAME Journal: International Computer Archive of Modern and Medieval English Journal, 7, 13–33.

Lv, C., Liu, H., Dong, Y., & Chen, Y. (2016). *Corpus based part-of speech tagging*. International Journal of Speech Technology, 19(3), 647–654.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Child Language Teaching and Therapy, 8.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). *Building a large annotated corpus of English: The Penn treebank.* Computational Linguistics - Association for Computational Linguistics, 19(2), 313-330.

Meakins, F. (2007). *Review of Computerized Language Analysis (CLAN). Journal of Language Documentation and Conservation, 1, 107–112.*

Meakins, F., & O'Shannessy, C. (2010). *Ordering arguments about: Word order and discourse motivations in the development and use of the ergative marker in two Australian mixed languages*. Lingua, 120(7), 1693–1713.

Meakins, F., Green, J., & Turpin, M. (2018). *Understanding linguistic fieldwork.* London: Routledge.

Moeller, S., & Hulden, M. (2018). *Automatic Glossing in a Low Resource Setting for Language Documentation*.

Nash, D. (1986). *Topics in Warlpiri grammar*. Garland Pub.

O'Shannessy, C. (2005). *Light Warlpiri: A New Language*.* Australian Journal of Linguistics, 25.

O'Shannessy, C. (2006). *Language contact and children's bilingual acquisition: learning a mixed language and Warlpiri in northern Australia.*

O'Shannessy, C. (2009). *Language variation and change in a North Australian indigenous community*. Variation in Indigenous Minority Languages. 10.1075/impact.25.21os.

O'Shannessy, C. (2010). *Competition between word order and case-marking in interpreting grammatical relations: a case study in multilingual acquisition.* Journal of Child Language. 38. 763-92.

O'Shannessy, C. (2012). *The role of codeswitched input to children in the origin of a new mixed language.* Linguistics. 50.

O'Shannessy, C. (2013). *The role of multiple sources in the formation of an innovative auxiliary category in Light Warlpiri, a new Australian mixed language.* Language, 89, 328–353.

O'Shannessy, C. (2016a). *Entrenchment of Light Warlpiri morphology.* Loss and renewal: Australian languages since colonisation, 217–252. De Gruyter Mouton.

O'Shannessy, C. (2016b). *Distributions of case allomorphy by multilingual children speaking Warlpiri and Light Warlpiri.* Linguistic Variation, 16(1), 68-102. https://doi.org/10.1075/lv.16.1.04osh

O'Shannessy, C. (2020). *How ordinary child language acquisition processes can lead to the unusual outcome of a mixed language.* International Journal of Bilingualism.

O'Shannessy, forthcoming. *Conventionalised creativitiy in the emergence of a mixed language - a case study of Light Warlpiri.*
Simpson, J. H. (1991). *Warlpiri morpho-syntax: A lexicalist approach.* Vol. 23. Kluwer Academic.

Parisse, Christophe & Normand, Marie-Thérèse. (2000). *Automatic disambiguation of morphosyntax in spoken language corpora.* Behavior research methods, instruments, & computers: a journal of the Psychonomic Society, Inc. 32. 468-81.

Rabiner, L. (1989). *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2):257–286.

Ratnaparkhi, Adwait. (2002). *A Maximum Entropy Model for Part-Of-Speech Tagging*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Seiss, Melanie. (2013). *Murrinh-Patha Complex Verbs : Syntactic Theory and Computational Implementation*. Doctoral Thesis. University of Konstanz.

Singh, S., Mohnot, K., Bansal, N., & Kumar, A. (2014). *Hybrid*

approach for Part of Speech Tagger for Hindi language. International Journal of Computer Technology and Electronics Engineering.

Wilks, Yorick. (2009). Machine translation: Its scope and limits. 10.1007/978-0-387-72774-5.

Zwarts, Simon & Dras, Mark. (2009). *Statistical Machine Translation of Australian Aboriginal Languages: Morphological Analysis with Languages of Differing Morphological Richness.* Proceedings of the Australasian Language Technology Workshop 2007, pages 134-142.

Appendix A: MOR program samples of output

A1) From spontaneous dataset:

*A22: Marda you sid-an iya eh .

%mor: n:prop|Marda pro|2sg v:intran|sit-down:loc det|here dis|eh .

*A22: you sid-an iya i-m warungka-warungka dat lyili .

%mor: pro|2sg v:intran|sit-down:loc det|here pro|i:3sg-m:NONFUT

%mor: unk|warungka–warungka det|that n:prop|@1 .

*A62: baby-pawu my baby-pawu my baby-pawu my baby-pawu my baby-pawu .

%mor: n|baby-dim@1 pro|1sg:poss n|baby-dim@1 pro|1sg:poss n|baby-dim@1 pro|1sg:poss n|baby-dim@1 pro|1sg:poss n|baby .

*A22: you www talk na nyampu-rra-ju .

%mor: pro|2sg v:intran|talk dis|foc:now ?|nyampu–rra–ju .

*A62: yaka nyiya nampu ?

%mor: dis|yaka qn|what@1 det|thishere@1 ?

*A62: nyiya nyampu ?

%mor: qn|what@1 det|thishere@1 ?

*A62: nyiya nyampu ?

%mor: qn|what@1 det|thishere@1 ?

*A22: yakarra nganta wirnkirrpa nyuntu ah don like wirnkirrpa-rla-ju .

%mor: dis|yakarra dis|@1-ta n|naughty@1 pro|2sg@1 dis|ah v:tran|dont:neg v:tran|like n|naughty@1-loc@1-top@1 .

*A62: sleep baby-pawu monkey-pawu .

%mor: v:intran|sleep n|baby-dim@1 n|monkey-dim@1 .

*C01: www .

%exp: can’t hear

*A62: monkey-pawu an orju .

%mor: n|monkey-dim@1 conj|an n|horse .

*C05: &dadadu .

*A62: orju-pawu nyampu .

%mor: n|horse-dim@1 det|thishere@1 .

*C01: baby-pawu nampu .

%mor: n|baby-dim@1 det|thishere@1 .

*A62: an nampu www nampu www .

%mor: conj|an det|thishere@1 det|thishere@1 .

*A62: papap-pawu .

%mor: n|pup-dim@1 .

*C01: nampu papap-pawu .

%mor: det|thishere@1 n|pup-dim@1 .

*C08: an nampu teddy bear-pawu .

%mor: conj|an det|thishere@1 n|teddy n|bear-dim@1 .

*A62: an nampu a-m look nampu Kordi house-rla .

%mor: conj|an det|thishere@1 pro|a:1sg-m:NONFUT v:tran|look
det|thishere@1 n:prop|Kordi n|house-loc@1 .

*A62: blah blah blah .

%mor: dis|blah dis|blah dis|blah .

A2) From narrative dataset:

*A57: dis three little boys dem go wirlinyji-kurra

%mor: det|this num|CARD n|little n|boys pro|3PL.S-m:NONFUT v:intran|go

n|hunting@1-allative@1

*A57: an dem findim jurlpu

%mor: conj|an pro|3PL.O v:tran|find-im:TR n|bird@1

*A57: an shanghai

%mor: conj|an n|slingshot

*A57: yeh dey bin chas-im

%mor: dis|yeh:aff pro|3pl:S n|bin v:tran|chas-TR

*A57: an dem findim nes-rla na dat jurlpu

%mor: conj|an pro|3PL.O v:tran|find-im:TR n|nest-loc@1 dis|foc:now
det|that n|bird@1

*A57: an dat jurlpu im jump at nes-janga

%mor: conj|an det|that n|bird@1 pro|i:3sg-m:NONFUT v:intran|jump prep|at
n|nest-abl@1

*A57: an dem chas-im na dat jurlpu im run kilji

%mor: conj|an pro|3PL.O v:tran|chas-TR dis|foc:now det|that n|bird@1
pro|i:3sg-m:NONFUT v:intran|run adv|hard@1^adv|hard@1-top@1

*A57: jinta kari shangai im fall down kanunju

%mor: num|one:CARD@1 suf|other@1 n|slingshot pro|i:3sg-
m:NONFUT v:intran|fall prep|down n|below@1

*A57: dis jinta-kari little boy

%mor: det|this num|one:other n|little n|boy

*A57: watiya-ng im trip-im down

%mor: n|tree@1-erg@1 pro|i:3sg-m:NONFUT v:tran|trip-TR prep|down

*A57: another two was running

%mor: n|another num|CARD^num|two v:intran|was v:intran|runn-ing:prog

*A57: still chasing dat jurlpu

%mor: adv|still v:tran|chas-ing:prog det|that n|bird@1

*A57: an jinta-kari wirliya-nga im pukum jilkarlan-ng

%mor: conj|an num|one:other n|foot@1-loc@1 pro|i:3sg-m:NONFUT
v:tran|poke-im:tran n|jilkarlan–ng

*A57: dis nother boy

%mor: det|this n|other n|boy

*A57: jinta kari im still ru:n chasim jurlpu

%mor: num|one:CARD@1 suf|other@1 pro|i:3sg-m:NONFUT adv|still n|run
v:tran|chas-im:TR n|bird@1

Appendix B: Concatenation rules (C-RULES) file

```
% ******************************************************
% GENERAL STARTS
% ******************************************************

% This rule starts all words that have full form listings

RULENAME: misc-start
CTYPE: START
if
NEXTCAT = [scat OR adv anaph art aux case conj det intj
suf:dir dis disj v n neg num prep pro pro:free v:intran
v:mod v:tran pro:qn n:prop typ n:kin com qan v:prev qn www]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {wat, bound-pro, dis-suf}

% ************************
% NOUN STARTS
% ************************

RULENAME: n-start
CTYPE: START
if
NEXTCAT = [scat OR n n:prop adv]
then
RESULTCAT = NEXTCAT
```

```
RULEPACKAGE = {nonfut, case, phon, lk, suf-gen, suf-foc-n,
focus, n-suf, p-encl, bound-pro, aux-root2, n-jarl, noun-
num, n-trans}

RULENAME: pro-start
CTYPE: START
if
NEXTCAT = [scat OR pro pro:free pro:lw pro:qn]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {case, nonfut, namu, suf-gen, block-erg-dat,
focus, n-suf, p-encl}



RULENAME: qn-start
CTYPE: START
if
NEXTCAT = [scat qn]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {qn-case}

% ************************
% VERB STARTS
% ************************

% start all verbs
RULENAME: v-start
```

```
CTYPE: START
if
NEXTCAT = [scat OR v v:tran v:intran]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {trans-suff, case-foc-v, v-tense, suf-gen,
bat, v-prep}

% start all preverbs
RULENAME: prev-start
CTYPE: START
if
NEXTCAT = [scat v:prev]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {p-encl-2}



RULENAME: det-start
CTYPE: START
if
NEXTCAT = [scat det]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {det-pl, det-case}

RULENAME: dis-start
CTYPE: START
```

```
if
NEXTCAT = [scat dis]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {case, gen-suf}

RULENAME: suf-category-start
CTYPE: START
if
NEXTCAT = [scat OR suf case]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {lk, lk2}



RULENAME: anaph-start
CTYPE: START
if
NEXTCAT = [scat anaph]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {suf-gen, case}



RULENAME: num-start
CTYPE: START
if
NEXTCAT = [scat num]
```

```
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {num-suf, num-case, num-dual}

RULENAME: num-case
CTYPE: -
if
STARTCAT  = [scat num]
NEXTCAT = [scat case]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

RULENAME: num-dual
CTYPE: -
if
STARTCAT  = [scat num]
NEXTSURF = pala | -pala
NEXTCAT = [scat aux]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

RULENAME: neg-start
CTYPE: START
if
NEXTCAT = [scat neg]
then
```

```
RESULTCAT = NEXTCAT
RULEPACKAGE = {n-suf}

% ************************
% VERB SUFFIXING
% ************************

% NEXTCAT [v:infl], DEL [allo]

RULENAME: trans-suff
CTYPE: -
if
STARTCAT = [scat OR v v:tran]
NEXTSURF = im | um | -im | -um
NEXTCAT = [scat v:deriv]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {bat, prep-suf}

RULENAME: n-trans
CTYPE: -
if
STARTCAT = [scat n]
NEXTSURF = im | -im
NEXTCAT = [scat v:deriv]
then
RESULTCAT = ADDCAT [scat v:tran]
RULEPACKAGE = {bat}
```

```
RULENAME: case-foc-v

CTYPE: -

if

STARTCAT = [scat v]

NEXTCAT = [scat suf:foc]

then

RESULTCAT = STARTCAT

RULEPACKAGE = {}


RULENAME: v-aux

CTYPE: -

if

STARTCAT = [scat OR v v:tran v:intran]

NEXTCAT = [scat v:aux]

then

RESULTCAT = STARTCAT

RULEPACKAGE = {}


RULENAME: v-tense

CTYPE: -

if

STARTCAT = [scat OR v v:tran v:intran]

NEXTCAT = [scat OR v:infl v:tense]

then

RESULTCAT = STARTCAT

RULEPACKAGE = {}
```

```
RULENAME: v-prep
CTYPE: -
if
STARTCAT = [scat OR v v:tran v:intran]
NEXTCAT = [scat OR prep suf:tel]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

RULENAME: prep-suf
CTYPE: -
if
NEXTCAT = [scat prep]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% future suffix {-rra}
% a-rra, yu-rra, i-rra wi-rra, de-rra
% nonfuture suffix {-m}
% a-m, yu-m, i-m, wi-m, de-m

RULENAME: nonfut
CTYPE: -
if
STARTCAT = [scat OR pro pro:lw]
NEXTCAT = [scat OR v:aux aux:lw]
then
```

```
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% ************************
% BOUND AND FREE PRONOUNS (WARLPIRI)
% ************************

% perhaps consider making a 'pro:bound' scat.
RULENAME: bound-pro
CTYPE: -
if
NEXTCAT = [scat pro] % change this if changing bound pro
scat
then
RESULTCAT = STARTCAT
RULEPACKAGE = {p-encl, case}

RULENAME: free-pro
CTYPE: -
if
NEXTCAT = [scat free:pro pro:free]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {case, suf-gen, p-encl}

RULENAME: pro-aux
CTYPE: -
if
```

```
STARTCAT = [scat pro]
NEXTCAT = [scat aux]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}


% ************************
% NOMINAL CASE MARKINGS
% ************************

RULENAME: case
CTYPE: -
if
STARTCAT = [scat OR n n:prop pro pro:qn dis pro:free \
anaph num disj v:tense adv n free:pro]
NEXTCAT = [scat case]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {lk2, case2, p-encl, suf-gen, focus, bound-
pro, dis-suf}


RULENAME: case2
CTYPE: -
if
NEXTCAT = [scat case]
then
```

```
RESULTCAT = STARTCAT
RULEPACKAGE = {case3}

RULENAME: case3
CTYPE: -
if
NEXTCAT = [scat case]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

RULENAME: ergative-case
CTYPE: -
if
NEXTCAT = [case erg]
then
RESULTCAT = STARTCAT, ADD [erg end]
RULEPACKAGE = {}

% FOCus

RULENAME: suf-foc-n
CTYPE: -
if
STARTCAT = [scat n]
NEXTCAT = [scat suf:foc]
then
RESULTCAT = STARTCAT
```

```
RULEPACKAGE = {}


% ************************
% OTHER KINDS OF SUFFIXING
% ************************

% general suffixing

RULENAME: suf-gen
CTYPE: -
if
NEXTCAT = [scat suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {case}


% demonstrative plural {-rra}

RULENAME: det-pl
CTYPE: -
if
STARTCAT = [scat det]
NEXTCAT = [scat OR case suf suf:foc]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}
```

```
% 2dual suffix, occurs on "yu" or anaph

RULENAME: 2dual
CTYPE: -
if
STARTCAT = [scat pro]
NEXTCAT = [scat suf]
then
RESULTCAT = ADD [scat pro]
RULEPACKAGE = {extra-suffix-ng}

% lk

RULENAME: lk
CTYPE: -
if
NEXTCAT = [scat suf]
NEXTSURF = lk | -lk
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% lk occurring after another suffix


RULENAME: lk2
CTYPE: -
```

```
if
NEXTCAT = [scat case]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {extra-suffix-ng}

% e.g. karnta-pawu-ng
RULENAME: extra-suffix-ng
CTYPE: -
if
NEXTCAT = [scat OR euph suf case:erg]
then
RESULTCAT = ADD [scat n] % added because ergative suffixes
occur with nouns
RULEPACKAGE = {}

% FOCus second suffix

RULENAME: extra-foc
CTYPE: -
if
NEXTCAT = [scat suf:foc]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {}

% -wat suffix (plural)
```

```
RULENAME: wat
CTYPE: -
if
STARTCAT = [scat OR n suf]
NEXTSURF = wat | -wat
NEXTCAT = [scat suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% phonotactics *infix*

RULENAME: phon
CTYPE: -
if
NEXTCAT = [scat euph]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% -bat suffix e.g. lickimbat

RULENAME: bat
CTYPE: -
if
NEXTCAT = [scat suf:iter]
then
RESULTCAT = STARTCAT
```

```
RULEPACKAGE = {bat-im}

% hit-im-bat-im

RULENAME: bat-im
CTYPE: -
if
NEXTSURF = im | um | -im | -um
NEXTCAT = [scat v:deriv]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% -namu reflexive

RULENAME: namu
CTYPE: -
if
STARTCAT = [scat pro]
NEXTSURF = namu | -namu
NEXTCAT = [scat suf]
then
RESULTCAT = ADD [scat refl]
RULEPACKAGE = {}

RULENAME: juk
CTYPE: -
if
```

```
STARTCAT = [scat det]
NEXTSURF = -juk
NEXTCAT = [scat suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% when "-juk" appears as a second suffix

RULENAME: juk-2
CTYPE: -
if
NEXTSURF = -juk
NEXTCAT = [scat suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% num+suf

RULENAME: num-suf
CTYPE: -
if
STARTCAT = [scat num]
NEXTCAT = [scat OR suf suf:foc]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {gen-suf}
```

```
% num+suf+suf

RULENAME: gen-suf
CTYPE: -
if
NEXTCAT = [scat suf]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {suf-case}


RULENAME: suf-case
CTYPE: -
if
NEXTCAT = [scat OR case suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% block ergative + dative combination

RULENAME: block-erg-dat
CTYPE: -
if
STARTCAT = [scat case]
NEXTCAT = [scat case], ![block ku]
```

```
then
RESULTCAT = STARTCAT
RULEPACKAGE = {focus, bound-pro, dis-suf, case-2, p-encl}

RULENAME: case-2
CTYPE: -
if
NEXTCAT = [scat OR case n], ![block OR dat-rla rli rlu]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {focus, bound-pro, dis-suf, p-encl}

RULENAME: det-case
CTYPE: -
if
STARTCAT = [scat det]
NEXTCAT = [scat case]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {det-case-2, focus, p-encl}

RULENAME: det-case-2
CTYPE: -
if
NEXTCAT = [scat OR case suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}
```

```
RULENAME: n-suf

CTYPE: -

if

NEXTCAT = [scat suf:n]

then

RESULTCAT = STARTCAT

RULEPACKAGE = {focus, case, bound-pro, p-encl}


RULENAME: n-jarl

CTYPE: -

if

NEXTCAT = [scat n]

NEXTSURF = -jarl

then

RESULTCAT = STARTCAT

RULEPACKAGE = {}


% derivational enclitics - do not follow a p-encl; ju is not

follow by other encl(except aux)


RULENAME: no-encl

CTYPE: -

if

NEXTCAT = [scat OR p-encl suf:dir], ![scat d-encl], ![block

OR p-enclju pencl-ju]

then

RESULTCAT = STARTCAT
```

```
RULEPACKAGE = {p-encl}



% ************************
% EXTRA SUFFIXING
% ************************

RULENAME: qn-case
CTYPE: -
if
STARTCAT = [scat qn]
NEXTCAT = [scat OR case suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

RULENAME: bin-vaux
CTYPE: START
if
NEXTSURF = bin
NEXTCAT = [scat v:aux]
then
RESULTCAT = NEXTCAT
RULEPACKAGE = {}

% FOCus first suffix
RULENAME: focus
CTYPE: -
```

```
if
NEXTCAT = [scat suf:foc]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {bound-pro}

% "fatha-wan"
RULENAME: noun-num
CTYPE: -
if
STARTCAT = [scat n]
NEXTCAT = [scat num]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {noun-num-suf}

% "fatha-wan-ing" / "fat-wun-pawu"

RULENAME: noun-num-suf
CTYPE: -
if
NEXTCAT = [scat OR case suf]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% ************************
% END RULE
```

```
% ************************

% ergative case END rule (since ergative markers always
occur last)
RULENAME: erg-end
CTYPE: END
if
NEXTCAT = [erg end]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}

% default END rule

RULENAME: all-end
CTYPE: END
if
STARTCAT = ![scat x]
then
RESULTCAT = STARTCAT
RULEPACKAGE = {}
```

Appendix C: Allomorphic rules (A-RULES) file

```
X = .*     % anything


% for variations in words ending in 'er' or 'a' (e.g. 'mother' and 'motha')
RULENAME: a-ending
LEX-ENTRY:
LEXSURF = $Xer
LEXCAT = [scat OR n prep]
ALLO:
ALLOSURF = $Xa
ALLOCAT = LEXCAT, ADD [allo nErB]
ALLO:
ALLOSURF = LEXSURF
ALLOCAT = LEXCAT, ADD [allo nErA]

% for ergative case marker

RULENAME: ergative
LEX-ENTRY:
LEXSURF: -ngku
LEXCAT = [scat case]
ALLO:
ALLOSURF = -ng
ALLOCAT = LEXCAT, ADD [allo 1ngku]
ALLO:
ALLOSURF = -ngu
```

ALLOCAT = LEXCAT, ADD [allo 2ngku]

ALLO:

ALLOSURF = -ng

ALLOCAT = LEXCAT, ADD [allo 3ngku]

ALLO:

ALLOSURF: -ing

ALLOCAT = LEXCAT, ADD [allo 3ngku]

% for variations in 'eep' spelling, e.g. 'sleep' and 'sliip'

RULENAME: eep-ending

LEX-ENTRY:

LEXSURF = $Xeep

LEXCAT = [scat v:intran]

ALLO:

ALLOSURF = $Xiip

ALLOCAT = LEXCAT, ADD [allo eepB]

ALLO:

ALLOSURF = LEXSURF

ALLOCAT = LEXCAT, ADD [allo eepA]

RULENAME: eel-ending

LEX-ENTRY:

LEXSURF = $Xeel

LEXCAT = [scat OR v v:tran v:intran]

ALLO:

ALLOSURF = $Xiil

ALLOCAT = LEXCAT, ADD [allo eel1]

ALLO:

ALLOSURF = LEXSURF

ALLOCAT = LEXCAT, ADD [allo eel2]

RULENAME: ck-k ending

LEX-ENTRY:

LEXSURF = $Xck

LEXCAT = [scat OR n v v:tran v:intran]

ALLO:

ALLOSURF = $Xk

ALLOCAT = LEXCAT, ADD [allo k-ending]

ALLO:

ALLOSURF = LEXSURF

ALLOCAT = LEXCAT, ADD [allo ck-ending]

% default rule- copy input to output

RULENAME: default

LEX-ENTRY:

ALLO: