2020-03-25

Parsing an American Sign Language Corpus with Combinatory Categorial Grammar

Michael Albert Nix
Brigham Young University

Parsing an American Sign Language Corpus with Combinatory Categorial Grammar

Michael Albert Nix

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Arts

Deryle Lonsdale, Chair
Rob Reynolds
Bryan Eldredge

Department of Linguistics

Brigham Young University

ABSTRACT

Parsing an American Sign Language Corpus with Combinatory Categorial Grammar

Michael Albert Nix
Department of Linguistics, BYU
Master of Arts

Research into parsing sign language corpora is ongoing. Corpora for German Sign Language and Italian Sign Language have been parsed (Bungeroth et al., 2006; Mazzei, 2011, 2012, respectively). However, research into parsing a corpus of American Sign Language is non-existent. Examples of parsed ASL sentences in literature are typically isolated examples used to show a particular type of construction. Apparently no attempt has been made to parse an entire corpus of American Sign Language utterances. This thesis presents a method for constructing a grammar so that a parser implementing Combinatory Categorial Grammar can parse a corpus of American Sign Language. The results are evaluated and presented.

Keywords: American Sign Language, ASL, corpus, Categorial Grammar, computational parsing

ACKNOWLEDGMENTS

I first must acknowledge my wife, Mariah. Her support, encouragement, and love saw me through to the end of this thesis. Next, Dr. Deryle Lonsdale's kind tutelage and patience helped steer this project from the very beginning to its culmination. Without his guidance it is not an exaggeration to say that this thesis would not have been completed. I also need to acknowledge Dr. Carol Neidle. I am indebted to her and her team for building the National Center for Sign Language and Gesture Resources Corpus. I want to also thank my family for their continued support and belief in me that I could undertake and complete such a large task. Lastly, the friends I have made during the course of the time spent on this thesis helped me to continue on. When the stress of having to work out how to complete this thesis was too much they were there to help me keep a good perspective.

Finally, I want to dedicate this thesis to my two lovely babies, Lily Rose and Benji. We had you both here with us for too short of a time, but we are grateful that we will have you both for forever.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational parsing has been explored for many languages, for example: German (Brants et al., 2002), Japanese (Kawahara et al., 2002) and Italian Sign Language (Mazzei, 2011). For these languages, language data exists in the form of corpora that provide resources enabling researchers to study and analyze those languages. Parsing corpus data is useful, for example, in machine translation (Vanderwende et al., 2015). Very few corpora exist for American Sign Language.

American Sign Language (ASL) is becoming more of an interest to researchers as more language data is made available. Research using ASL corpora with relation to parsing is as yet an under-explored endeavor. The research carried out in this thesis parses a well-known ASL corpus implementing a syntactic approach called Combinatory Categorial Grammar. This is apparently the first attempt at parsing a relatively large corpus of ASL to date.

When parsed sentences of ASL are presented in literature, they usually demonstrate a particular construction or point of interest. These examples are typically presented following the X-bar schema for syntactic constituency. Fewer parsed examples exist of ASL sentences being analyzed with Combinatory Categorial Grammar, and only a few researchers have investigated syntactic constructions in ASL using Combinatory Categorial Grammar (Kuhn, 2016; Wright, 2008). In order to parse the ASL corpus for this thesis, the parsing program OpenCCG was utilized, and I constructed a grammar of ASL for OpenCCG.

I will show that Combinatory Categorial Grammar is suitable for parsing a corpus of American Sign Language gloss, which apparently has not been done before. I will also

show that Categorial Grammar can provide analyses for a wide range of interesting ASL constructions.

The benefit of this work can help give researchers another tool to use in describing ASL constructions. It can be used in machine translation efforts, as has been done for Italian Sign Language (Mazzei, 2011, 2012).

This document is organized as follows: Chapter 2 is a review of literature of American Sign Language structure; different syntactic theories and parsing approaches; an in-depth discussion on Combinatory Categorial Grammar; related work; and ASL corpora. Chapter 3 discusses preparing the data; the approach taken to build the grammar; running OpenCCG; and refining the grammar. Chapter 4 presents several generated parse results that contain different syntactic structures and features. Chapter 5 is a discussion of the evaluation process. Finally, Chapter 6 discusses the limitations and proposed future work.

Chapter 2

Literature Review

This chapter begins with a general discussion of American Sign Language (ASL). The structure of ASL along with different types of constructions are presented. After this discussion a review of literature on syntactic theories and parsers is given. The main bulk of the section on syntactic theories and parsers focuses on Combinatory Categorial Grammar. The parsing program used in this thesis, OpenCCG, is also mentioned. The last section is on corpora, in particular sign language corpora.

1 American Sign Language

1.1 Background

Formal linguistic investigations into American Sign Language began in earnest in the 1960s beginning with Stokoe (1960). *Sign Language Structure* was the first publication to give the same type of rigorous study to a signed language as had been given previously to spoken languages. In it, Stokoe proposed that signs are composed of discriminant elements which are analyzable (Stokoe, 1960, p. 33). He proposed that ASL has distinct phonology[1] and morphology. Later, Stokoe (1970) would demonstrate that ASL also has underlying syntactic structure.

The modality in which ASL is transmitted is obviously different than that of English. ASL is a manual-visual language whereas English is an oral-aural language. The visual

[1]Stokoe proposed the term *cherology* in place of phonology.

component of ASL allows for information to be encoded in multiple channels, i.e. non-manual markers such as facial expressions, head tilt, and eye-gaze.

ASL linguistic researchers have developed systems which can encode all the information from non-manual markers and manual signs in an orthographic representation. The common practice for writing ASL is to use a gloss. ASL gloss uses English words but maintains the word order for ASL. The standard convention is to use small capital letters when writing the gloss, as shown in Example (1). Some glosses will have additional tags affixed to them. Some of these include a #, fs-, +, and : followed by a lowercase letter. The # typically indicates that the word whose gloss the symbol is affixed to was finger-spelled. Likewise, the fs- preceding gloss indicates that the word was finger-spelled. + affixed to the end of a sign usually means that sign was reduplicated or somehow continued on.

(1) fs-JOHN fs-MARY LOVE

'As for John, Mary loves him.'

1.2 Structure

The basic constituent order for ASL is Subject-Verb-Object. Example sentence (2) shows this pattern: the subject precedes the verb and the object follows. There is no other possible interpretation for the word order from the example given in (2).

(2) TEACH+AGENT LIKE CHOCOLATE

'The teacher likes chocolate.'

Although ASL is an SVO language, different patterns are possible. As explained hereafter, sentences which have topicalized elements, wh-words, and subject or object pro-drop involve other orders. When this is the case indicators mark the affected constituents. These indicators are realized as various non-manual markers.

Non-Manual Markers

Non-manual markers (NMMs) are important and even necessary for correct grammatical constructions for certain syntactic structures in ASL. Facial expressions, eye-gaze, eyebrow movement, head tilt, body position, and mouth movement all come together with signing to form correct ASL (Herrmann and Steinbach, 2011). Non-manual markers each have their own role in sign language:

> They express a variety of lexical, morphosyntactic, prosodic, semantic, and pragmatic functions such as attributive, adverbial, and aspectual modification, negation, sentence types, reported speech, constructed action, and information structuring (Herrmann and Steinbach, 2011, p. 3).

NMMs give signed languages their multi-channeled character. In aural-oral languages the type of information carried by NMMs is expressed via the syntax, morphology, pitch, tone, and prosody. Sign language discourse participants must pay attention to what the hands are signing as well as to what the face is producing. For example, relative clauses, topicalization, and negation all rely upon NMMs. Subtle movement with lips, eye brows, and head tilt are strongly associated with relative clause constructions in ASL (Liddell, 1980).

NMMs are indicated in the gloss by a line and a subscript label above the lexical items which they have scope over. Table 2.1 contains the names of NNMs along with their reference labels and a description.

NNM	Label	Description
Relative Clause	rc	Indicates the scope of the relative clause.
Rhetorical Question/Wh-word	rhq/wh	Indicates the scope of a wh-question. Used in constructions where the signer provides an answer to the question just asked.
Topic	t	Indicates that a sign has been topicalized.
Negation	Neg	Indicates the scope of negation.

Table 2.1: Non-manual marker labels

Relative Clauses

Relative clauses are identifiable by the associated NMMs produced in conjunction with the manual signs. The NMMs scope over the portion of the utterance which composes the relative clause and help to disambiguate the type of relative clause. ASL has both internally and externally headed relative clauses (Wilbur, 2017). Aside from NMMs identifying relative clauses, two manual markers help in identifying relative clauses: complementizers and relative pronouns (Wilbur, 2017, p. 5-6).

The complementizer THAT may be used within a relative clause. However, sentences with an overt THAT as a complementizer are considered to be "Englishy", so it is not usually signed. The second way to identify relative clauses with manual markers is with relative pronouns. Relative pronouns may be identifiable by their agreement morphology (Wilbur, 2017). Person is typically notated by using '1' for first, '2' for second, and '3' for third, e.g. IX-1p, IX-2p, IX-3p. IX is an index marker indicating the referent being pointed to by the index finger. In addition to person numbering, letters may be affixed to help disambiguate referents, e.g. IX-3p:i, IX-3p:j, IX-3p:k, IX-3p:l, etc. The example in Figure 2.1 comes from the ASL corpus.

$\overline{\text{CAT CHASE \#DOG}}^{rc}$ IX-3p:i EAT MOUSE/FICTION

'The cat that chased the dog ate the mouse.'

Figure 2.1: A sentence with a relative clause

The line above CAT CHASE #DOG indicates the scope of the NMMs distinguishing that part of the utterance as the relative clause. CAT is the nominal head of the relative clause. IX-3p:i serves as a type of relative pronoun which links the relative clause to the matrix clause. IX-3p:i serves as the subject of the matrix clause but it also stands in relation to CAT. The pronoun in this instance may be acting like a resumptive subject, present to fill in for CAT.

Topicalization

ASL is a topic-comment language. The process of topicalization may affect the underlying SVO syntactic ordering. Aarons (1994) presents evidence for three different types of topicalization in ASL. Her evidence includes examples of two types of based-generated topics and moved topics. The effect of the different types of topicalizations upon word ordering makes ASL look very different from English.

Base-Generated Topics[2]

The primary characteristic that identifies a based-generated topic is that the topic "does not constitute an argument of the main verb" (Aarons, 1994, p. 152). The topicalized element in Example (3) is not an argument of the verb LIKE since the argument position is already occupied by CORN. However, there is a type of semantic relationship between the topic and the verb's argument CORN: class-element

(3) $\overline{\text{VEGETABLE}}^{t}$, JOHN LIKE CORN

'As for vegetables, John likes corn.'

Similarly, (4) has an object overtly expressed in the argument position of the verb. In this example though, the topic is coreferential with the object pronoun IX-3RD (indicated by the coindexed $_i$):

(4) $\overline{\text{MARY}}^{t}_{i}$, JOHN LIKE IX-3RD$_i$

'As for Mary, John likes her.'

Aarons (1994) argues that the topics in (3) and (4) are not arguments of the main clause verb and thereby could not have moved from an object position to the topic position. The conclusion is then that these topics were base-generated in a sentence-initial position.

[2]The examples in this section come from Aarons (1994).

Moved Topics

Example (5) shows a topicalized noun that has moved from its original argument position.

(5) $\overline{\text{MARY}}^{t}$, JOHN LOVE

'Mary$_i$, John loves e_i.'

The topicalization movement operation changes the SVO ordering to OSV. MARY, which is the object of LOVE, moved from its argument position into the topic position.

Negation

Negation in American Sign Language is formed from two parts: a non-manual marker, i.e. a headshake, and an optional manual sign. The NMM accompaniment is obligatory. When negation is used, a headshake occurs coextensively with the manual negation sign. The NMM has scope over the part that is being negated. This allows for only the NMM ($+neg$) to be overtly expressed with no negative manual sign being produced and still have the same interpretable effect. Examples (6), (7) and (8) all come from Neidle et al. (2000, p. 45).

(6) *JOHN $[\text{NOT}]_{Neg}$ BUY HOUSE

(7) *JOHN $\overline{[+neg]}_{Neg}$ BUY HOUSE

(8) JOHN $\overline{[+neg]_{Neg}\text{BUY HOUSE}}^{Neg}$

Example (6) shows that when *only* the manual sign is produced the utterance is ungrammatical. Example (7) shows that when the NMM does not spread across the entirety of what is being negated, the utterance is, again, ungrammatical. When the NMM scopes over the entire constituent being negated, as in (8), then the manual sign is not mandatory, and the utterance is grammatical.

pro-drop

American Sign Language is a pro-drop language. Lillo-Martin (1986) shows that ASL can have null subjects and null objects in tensed finite clauses. Example (9) is a subject pro-drop and (10) is an object pro-drop construction.

(9) SHOOT:i fs-FRANK

'(He/She) shoots Frank.'

(10) fs-JOHN SHOOT:j

'John shot him/her/it.'

Wh-words

Wh-words and their associated movement in ASL have been studied exclusively using X-bar theory (Aarons, 1994; Neidle et al., 1998b; Petronio and Lillo-Martin, 1997). There is debate as to whether wh-words undergo rightward or leftward movement in ASL. Petronio and Lillo-Martin (1997) argue that wh-words undergo leftward movement whereas Neidle et al. (1998a) argue that wh-words undergo rightward movement. Petronio and Lillo-Martin also propose that in the process of movement a wh-word can have a copy of itself, dubbed a "wh-twin", which can either be overtly expressed or elided. The appearance of wh-words in various locations in sentences, along with ASL allowing pro-drop , can give rise to different surface structures that diverge from an SVO ordering.

To illustrate various positioning of wh-words in ASL sentences[3], (11), (12) and (13) illustrate when the wh-word can remain *in situ*. The sentences in Examples (11) and (13) have wh-words in object position, and the sentence in (12) has the wh-word in subject position.

(11) fs-JOHN LOVE WHAT

'What does John love?'

[3]The examples in this section come from the NCSLGR Corpus unless otherwise stated.

(12) WHO LOVE fs-JOHN

'Who loves John?'

(13) fs-JOHN SEE WHO YESTERDAY

'Who did John see yesterday?'

Word re-ordering occurs when a wh-word is topicalized or placed into focus position. Example (14) is composed of two utterances. The first utterance is a question; the second is the answer. The NMMs associated with this utterance have been annotated in the NCSLGR Corpus as reflecting a rhetorical question. This changes the SVO order to OSV.

(14) $\overline{\text{WHO fs-JOHN SEE PART:INDEF}}^{rhq/wh}$ fs-MARY

'Who did John see? Mary.'

Wh-words may also appear in sentence-final position. Example (15) comes from the NCSLGR Corpus and (16) comes from Petronio and Lillo-Martin (1997). Neidle et al. (1998b) argue that the wh-word moved rightward to the sentence-final position. Petronio and Lillo-Martin (1997) argue that (16) is a double construction wh-element where the overt wh-word is in focus position with its twin null, a pro-drop subject. The end result is that (15) has VOS word order and (16) has VO wh-twin.

(15) LIKE CHOCOLATE WHO

'Who likes chocolate?'

(16) pro_e BUY CAR WHO

'Who bought the car?'

Another operation called doubling can cause two wh-words to be present in the same utterance. Example (16) is, according to Petronio and Lillo-Martin (1997), an instance of wh-word

doubling with the *in situ* wh-word nully expressed. Example (17) is an example of wh-word doubling where the wh-word appears sentence-initially and sentence-finally.

(17) WHO fs-JOHN LOVE WHO

'Who, who does John love?'

Passives

Janzen et al. (2001) present arguments for—and evidence of—the passive construction in ASL. They provide an outline for identifying passives based upon the evidence that ASL forms passive constructions via spatial morphology and non-manual marker cues rather than word order. Sentences which are active can be made to be passive by one of two ways: (1) placing focus on the patient by taking the point of view of the "patient rather than the agent" (Janzen et al., 2001, p. 283), and (2) demoting the subject-agent by omitting it. A transitive verb that begins at the spatial locus of an agent can start from an empty locus, thereby the verb is not being associated with an agent. Example (18) is a passive construction found in the NCSLGR Corpus.

(18) SOMETHING/ONE CAR STEAL

'Someone's car was stolen.'

2 Syntactic Theories and Parsers

Parsing maps linear sequences of input tokens (usually words and sentences) into a model of the hierarchical structure that underlies language. Different parsers implement analyses proposed by different syntactic theories. Chomsky (1956) introduced phrase structure rules as a component of context-free grammars (CFGs). Phrase structure rules define how words and constituents combine. CFGs are defined by a four-tuple specification: N, Σ, R, and S. Table 2.2 sets out the formal definition of the four-tuple (Jurafsky and Martin, 2009, p. 391)

N	is a set of non-terminal symbols
Σ	is a set of terminal symbols
R	is a set of production rules; each rule has the form of $A \rightarrow \beta$
	A is a non-terminal
	β is a string of symbols composed from the set $(\Sigma \bigcup N)$
S	is a start symbol

Table 2.2: Definition of the CFG four-tuple

2.1 Parsing Strategies

There are many different formal approaches to syntactic parsing. The X-bar schema is very prevalent in linguistic literature due to its effectiveness at representing syntactic structure (Chomsky, 1970; Jackendoff, 1977). X-bar theory utilizes phrase structure rules in generating a parse, i.e. syntax trees. This theory allows constituent heads to project up to immediate dominating nodes. The nodes formed from projections may themselves combine to form another constituent projection. This continues until the entire parse tree is formed.

Two other parsing strategies are shallow parsing and chunk parsing. Shallow parsing is used to capture only specific parts of a string that is of interest instead of parsing out the entire string. For tasks that do not require fully generated parses, shallow parsing is a good alternative. Li and Roth (2001) demonstrate that under certain conditions for specific tasks shallow parsing performs as well as a full parser. Chunk parsing is similar to shallow parsing in that parts of the string are grouped, or chunked, together to form a larger constituent. Unlike shallow parsing, however, the entire string is parsed. The parser will chunk together the tokens in the string which form a unit (Abney, 1991).

The next few sections cover different syntactic theories that have been used in studying signed languages. Each theory has its own way of modeling language. Within each section is also a brief mention of the types of parsers that were used[4].

[4]The Natural Language Toolkit (NLTK) package for Python includes many of the parsers discussed. Bird et al. (2009) is a great resource for learning how to use NLTK. `https://www.nltk.org/book/`

2.2 Government Binding/Principles & Parameters

Government binding (GB), or Principles and Parameters (P&P), is the most prevalent theory adopted to date for doing linguistic research into ASL. Papers where authors followed the structural pattern for GB, the X-bar schema, include Braze (2004); Fischer (1996); Neidle et al. (1998a,b) and Petronio and Lillo-Martin (1997).

Wh-words studied under the lens of Government Binding and Principles & Parameters employing the X-bar schema are associated with movement operations. Example (19) shows the steps undertaken to form a wh-question in English. If we are curious about John's eating habits, a natural question could be "What did John eat?". To arrive at this question, specific steps are followed.

(19) (a) 'John ate an apple.'

(b) 'John ate (what)?'

(c) '$What_i$ did John eat $\_\_\__i$?'

To get to the final step (c), the question "What did John eat?", one must change the direct object of the verb into a wh-word, in this case *what*. The newly inserted wh-word is then fronted in the sentence, and for English the word *did* is inserted into the sentence. Figure 2.2 is an X-bar syntax tree showing the wh-movement.

Several movement operations are undertaken in Figure 2.2. *What* undergoes the highest movement in the tree, moving from deep within the tree to then being nearly at the very top.

Figure 2.3 shows an X-bar syntax tree based upon the position adopted by Neidle et al. (1998b). The sentence is HATE JOHN WHO, where WHO is the subject of the sentence that has been moved to the right of CP.

A somewhat undesirable feature of this approach is that unlike other SVO languages that are right binary branching, this diagram assumes an initial left binary branching process,

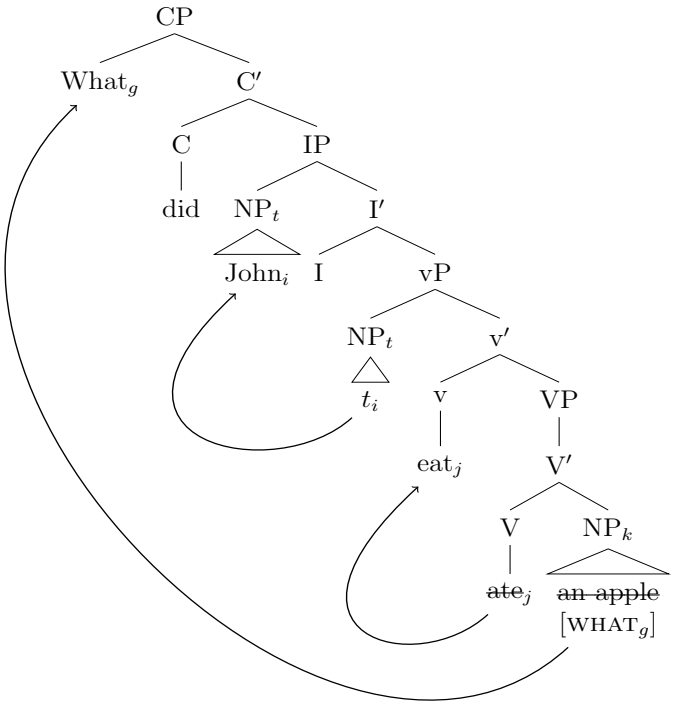


Figure 2.2: Syntax tree showing wh-movement in English

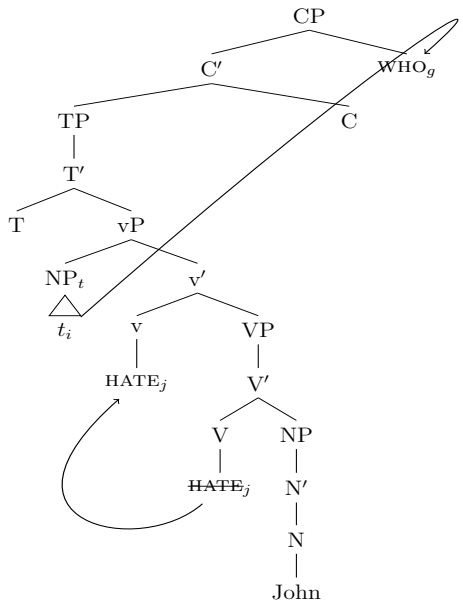


Figure 2.3: Rightward wh-movement: ASL

but only above the TP node, i.e. the Tensed Phrase—this is equivalent to the IP node in Figure 2.2 (see also Neidle et al. (2000, p. 3)). Figure 2.4 is a close depiction of what is found in Petronio and Lillo-Martin (1997, p. 27). This is an abridged syntax tree for the purpose of showing that the landing site for wh-words is to the left of CP. The sentence which is diagramed has two wh-words. The first WHAT is the moved wh-word and the second WHAT is a base generated wh-word. The parsed sentence is: WHAT NANCY BUY YESTERDAY WHAT.

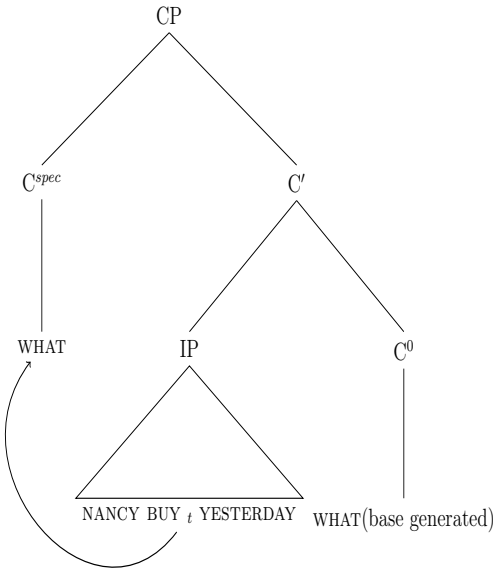


Figure 2.4: Leftward wh-movement: ASL and wh-double

The argument that Petronio and Lillo-Martin (1997) make is that wh-words in ASL are leftward moving because "true WH-movement is leftward" and furthermore "the specifier of CP is universally on the left, even in languages that have been argued to have other specifier positions to the right. The source of this generalization remains a mystery, but its strength is striking."

2.3 Lexical-Functional Grammar

Lexical-Functional Grammar (LFG) has been used for sign language research into ASL (Speers, 2001) and Indian Sign Language (Dasgupta et al., 2008). LFG represents sentences

in two ways via constituent structure (c-structure) and functional structure (f-structure). C-structure is rendered as a traditional syntax tree showing word order and constituency. F-structure is represented in attribute value matrices (AVM). Grammatical and morphosyntactic information may be shown in the AVM. Speers (2001) employed a third structure, phonetic transcription (p-structure). This p-structure is where he encoded NMMs. His work produced a mono-directional machine translation from English into ASL (Speers, 2001, p. 11). He achieved this by creating the *ASL Workbench* software program that implemented an LFG grammar for ASL.

2.4 Head-driven Phrase Structure Grammar

Marshall and Safar (2004) created a grammar based upon Head-driven Phrase Structure Grammar (HPSG) principles for British Sign Language (BSL). They used the Attribute Logic Engine (ALE)[5] to build a parser implementing HPSG. The generated grammar modeled a signing avatar for BSL. HPSG explicitly specifies the sub-categorization for each lexical item required. The specification is laid out in an attribute value matrix. Each lexical item has an attribute (at least one) with a value (at least one). Attributes are specified by their values. Taking this approach, Marshall and Safar (2004) were able to utilize properties pertaining to a sign and any accompanied NMM. For any sign, they list the attributes belonging to that sign and specify the values pertinent to that sign. It is an ingenious way to capture and encode both manual and non-manual information.

2.5 Dependency Grammar

Dependency Grammar directly maps the relationships which exist among the lexical items in a string (Osborn, 2014). The direct mapping from heads to dependents does away with constituency mapping where relationships are projected to any intermediary dominating node. This framework directly shows how lexical items relate to one another.

[5]The ALE can be found at: `https://www.cs.toronto.edu/~gpenn/ale.html` (Carpenter and Penn, 2001).

Östling et al. (2017) were the first to demonstrate a dependency grammar annotation for any signed language. They produced dependency parses for Swedish Sign Language (SSL) sentences from the SSL Corpus (Mesch and Wallin, 2015). The parser they used is part of the UDPipe toolkit[6] developed by Straka et al. (2016). Östling et al. (2017) parsed a SSL utterance containing both a verb sandwich and a verb chain.

3 Categorial Grammar

Categorial Grammar (CG) is a syntactic theory which is founded in the logic and philosophy of Frege, the functional categories presented by Bar-Hillel (1953), Lambek calculus (Lambek, 1958), and directly from the logic of Ajdukiewicz (Wood, 1993). As the name implies, Categorial Grammar is built upon grammatical categories that are closely associated with the lexicon. CG employs a scarce number of atomic categories: (**n**)oun, (**s**)entence, and noun phrase (**np**). The atomic categories themselves are used as functions seeking arguments when applied together with directional slashes. Functions may seek either other functions or atomic categories as their arguments. For example, a determiner can combine with a noun to make a noun phrase. To do this the determiner will have a complex category of **np/n**, and the noun will have an atomic category of **n**. When a noun follows a determiner, the function may consume its argument. The directional slashes indicate which argument the function is searching for in order to form a constituent.

Pure CG had limited functionality for constituent composition. Bar-Hillel (1953) laid out the foundational principle of what would later become known as functional application, which allows for composition between a function and its immediately adjacent argument (see Figure 2.5).

CCG has several application labels that delineate the different composition processes. Additional rules with their unique application labels have been included in the grammar.

[6] `https://github.com/bnosac/udpipe`

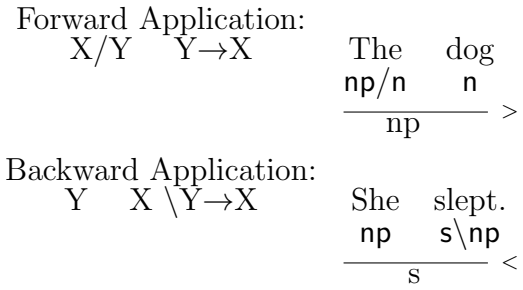


Figure 2.5: Functional application schemas

Table 2.3 includes the label types, the associated rules to which they belong, and a brief description, which will be further described hereafter.

Categorial Grammar has been extended to include more ways of composing categories into larger constituents. One such extension is Combinatory Categorial Grammar which is the parsing strategy implemented in this thesis.

3.1 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG) emerged when Ades and Steedman (1982) developed the process of allowing functions to compose with other functions. This extended the power of functions from merely consuming arguments to being able to combine to form a new function. For example, the functions of modal auxiliary verbs and main verbs may combine to yield a new complex functional category. Other extensions were adopted into CCG, such as type raising (Steedman, 1985) which "turn arguments into functions over functions-over-such-arguments"(Steedman, 2014, p. 675). One practical application of type raising is when subjects need to be functions over predicates.

Combinatory Categorial Grammar abides by a few axioms. The first rule, Principle of Adjacency, states that only immediately adjacent lexical items may combine. This rule is foundational in all categorial grammars. The second rule, Principle of Consistency, was specifically introduced in CCG when functions were allowed to compose with other functions. This rule ensures that when one function is seeking out the result of a subordinate function, that the position of what is being searched for is in its proper argument place. For example,

$>$	Forward Application	Allows functions to consume arguments to the right.
$<$	Backward Application	Allows functions to consume arguments to the left.
$>_B$	Forward Composition	Allows two functions to compose together.
$>_T$	Forward Type-Raising	Raises an argument type to be a function over a function that would have taken the unraised type as an argument.
$<_{B_x}$	Backward Crossed Composition	Allows two functions to compose when they have different slash type directions.
$_{subj\ pro-drop}$	Subject pro-drop	Allows a predicate category to drop its subject requirement when no subject is present.
$_{obj\ pro-drop}$	Object pro-drop	Allows a predicate category to drop its object requirement when no object is present.
$_{topic}$	Topic Raising	Changes a topicalized object to be a function over a predicate.
$_{n-to-np}$	Noun-to-Noun-Phrase	Changes a noun from an atomic **n** to a noun phrase atomic category **np**.
$_{twin}$	Noun to Predicate Function	Changes an **np** to a complex function that may compose with a predicate.

Table 2.3: Application labels for CCG rules utilized in the grammar

Y/Z and (X\Y)/Z may compose a new function of X/Z because the Y is to the left of X and would be the result of Y/Z once Z was consumed. The rule Principle of Inheritance enforces that the proper slash direction is maintained. In the example of Y/Z (X\Y)/Z the new resulting functional category is X/Z because the forward slash is present in both inputs.

Additional compositional functions developed in Curry and Feys (1958) were subsumed by Steedman (1993). The first rule which extends CG into CCG is composition. This rule allows for the argument of a function and a head of another function to merge into a new compositional category. This is useful, for example, when a modal precedes a verb, as in Figure 2.6.

Forward Composition:
$X/Y \quad Y/Z \rightarrow_{>B} X/Z$

$$\frac{\begin{array}{cc}\text{might} & \text{eat} \\ (\mathsf{s}\backslash\mathsf{np})/(\mathsf{s}\backslash\mathsf{np}) & (\mathsf{s}\backslash\mathsf{np})/\mathsf{np}\end{array}}{(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}}{>}B$$

Figure 2.6: Forward Composition

A second rule is forward type raising. This rule changes an atomic category to be a function, essentially it "makes [a] subject NP into a function over predicates"(Steedman, 1993, p. 234). This rule is productive in constructions that have topicalized elements, as the sentence in Figure 2.7.

Forward Type-Raising:
$Y \rightarrow_{>T} X/(X\backslash Y)$

$$\begin{array}{ccccc}\text{Harry} & & & \text{Mary} & \\ \mathsf{np} & \text{cooked and} & & \mathsf{np} & \text{ate some apples.} \\ \overline{\mathsf{s}/(\mathsf{s}\backslash\mathsf{np})}{>}T & & & \overline{(\mathsf{s}/(\mathsf{s}\backslash\mathsf{np}))}{>}T & \end{array}$$

Figure 2.7: Type raising—example from (Steedman, 1993, p. 234)

Another rule is backward crossed composition. This rule is applied whenever a verb, or a lexical item with a complex functional category, is modified by a following lexical item. For example, in French, verbs are often modified by post-adverbials. *Embrasse* (to kiss) is modified by the post-adverbial *souvent* (often), as seen in Example (20).

(20) Kim lui embrasse souvent.

'Kim often kisses her.'

Backward crossed composition allows for disparate slash types to compose. The type categories for *embrasse* and *souvent* are **(s\np)/np** and **(s\np)\(s\np)** respectively. The main slash type in the functional category for *embrasse* is a forward slash and *souvent* has a back slash in its functional category. These two slash types are in opposite direction. Backward crossed composition allows for the functional category of *souvent* to compose with the functional category of *ebrasse*, see Figure 2.8.

Backward Crossed Composition:
$Y/Z \quad X\backslash Y \rightarrow_{<B_x} X\backslash Z$

$$\frac{\begin{array}{cc}\text{embrasse} & \text{souvent} \\ (\mathsf{s}\backslash\mathsf{np})/\mathsf{np} & (\mathsf{s}\backslash\mathsf{np})\backslash(\mathsf{s}\backslash\mathsf{np})\end{array}}{(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}}{<}B_x$$

Figure 2.8: Backward Crossed Composition

Other permissible combinatory operators are defined in Steedman (2014, p. 679). These extended rules make CCG more powerful than pure Categorial Grammar: complex syntactic constructions can be derived such as composing modals with main verbs, topicalization, and parasitic gap constructions (Steedman, 2014).

The relative simplicity that CCG offers for parsing a sentence makes it an attractive option for implementing a computational analysis for any language. The following section examines what work has been done with regard to CCG and Signed languages specifically.

3.2 Related Work

The first paper that analyzed American Sign Language using Combinatory Categorial Grammar comes from Wright (2008). He showed that CCG is a viable syntactic theory for doing linguistic research into ASL. He demonstrated that certain syntactic structures such as topicalization, relative clauses, sentential complements and coordination can be effectively modeled by using CCG. The limitation of this paper comes from the examples:

"The sentences used as data for this project are of a well-attested, high-occurrence type which are well-known and documented..." (Wright, 2008, p. 139). Wright used a few examples to demonstrate that CCG can model a few instances of ASL constructions. He did not do an in-depth analysis of various types of constructions that can be found in a corpus.

Wright shows how CCG is capable of generating parses for sentences with fronted objects. He proposes a rule that changes an **np** category into a complex function that mirrors Forward Type-Raising. Wright also suggests using a vertical bar (|) in the argument function (see Figure 3.5, the example comes from (Wright, 2008, p. 41)). The vertical bar allows for the input argument to be any directional slash, i.e. either forwards (/) or backwards (\).

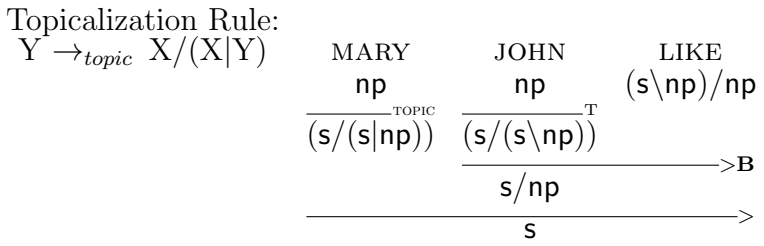


Figure 2.9: Topicalization
Mary, John likes. (i.e., You know Mary? John likes her.)

The Topicalization Rule proposed in Wright (2008) must work in conjunction with forward type-raising. When the subject **np** is type-raised to be a function over the predicate, it may combine via forward composition. Finally, the topicalized object can consume the predicate as its argument since the vertical bar does not discriminate between directional slashes.

Table 2.4 lists several papers which employed CCG to analyze various signed languages.

4 Corpora

Corpora are collections of language data. The data may be either textual, spoken or, in the case of sign language corpora, visual.

Sign language corpora are relatively new, beginning in the 1990s (Neidle et al., 1997; Segouat and Braffort, 2009). The reason for such a late adoption into corpus research is

Researcher	Language
Wright (2008)	American Sign Language
Kuhn (2016)	American Sign Language
Mazzei (2011)	Italian Sign Language
Mazzei (2012)	Italian Sign Language
Geraci et al. (2014)	Italian Sign Language
Chung and Park (2011)	Korean Sign Language
Sevinç (2006)	Turkish Sign Language

Table 2.4: Research using CCG\OpenCCG

primarily due to technological limitations. As interest in sign language research has increased, so too has the development of more sign language corpora (Bono et al., 2014; Bungeroth et al., 2008, 2006; Forster et al., 2012; Mesch and Wallin, 2015; Neidle et al., 2012; Neidle and Vogler, 2012). The National Center for Sign Language and Gesture Resources (NCSLGR) Corpus (Neidle and Vogler, 2012) is the corpus from which all the data contained in this thesis comes. A more comprehensive discussion of the NCSLGR Corpus is found in the next chapter.

5 Hypothesis

This thesis addresses two main hypotheses: 1. Combinatory Categorial Grammar is well suited for use in parsing a sizable corpus of American Sign Language gloss, which has apparently not been done yet; 2. The theoretical framework of Categorial Grammar can provide analyses for a wide range of interesting constructions in ASL, whereas only a few constructions have been addressed to date.

Chapter 3

Methodology

This chapter begins with an explanation of how the data used for the grammar was obtained. Statistical information about the NCSLGR Corpus is also provided. Next, the parsing program OpenCCG is explained in more detail. The necessary input file types are described: how the grammar was constructed and which steps are necessary to build a grammar. Specific rules to the ASL grammar are listed and justified with examples in the Unary Rules section. How to run OpenCCG with different available commands is discussed. A discussion of the process of developing and refining the grammar concludes the chapter.

1 Input Data Preparation

The American Sign Language Linguistic Research Project (ASLLRP) began as a way to make ASL language data examples publicly available. The ASLLRP team has made ASL corpora publicly available for research, including the National Center for Sign Language and Gesture Resources (NCSLGR) Corpus. The NCSLGR Corpus is a video-based corpus of recordings of ASL utterances.

Since the ASLLRP is an ongoing project, it produces periodic updates. The NCSLGR Corpus is not the most currently available corpus provided by the ASLLRP. Since the beginning of this thesis work, it has been merged with a new data set to create the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus (Neidle et al., 2012). The ASLLVD Corpus is still in its last stages of completion and a cautionary note warns that the data set

is still evolving. Due to the transitory nature of the ASLLVD, I opted to adopt the NCSLGR Corpus for this thesis work.

The NCSLGR Corpus is housed at Boston Univeristy[1] under the supervision of Carol Neidle and Stan Sclaroff. The corpus itself is a collection of short narrative and elicited utterance videos from deaf signers using American Sign Language (ASL). A total of 8 participants provided the language examples. Each video in the corpus has been annotated with an ASL gloss transcription, non-manual markers, English translation, and metadata.

The corpus contains both narrative stories and intentionally elicited utterances. The elicited utterances demonstrate specific syntactic constructions that may otherwise not be provided from spontaneous narrative utterances. The elicited utterances are labeled in the XML files and in the DAI by NCSLGR in the file name. For example, NCSLGR10j.xml contains elicited utterances. Any of the files that do not contain NCSLGR in the name are recordings of narratives.

The corpus consists of XML files which contain all the language data that is searchable via the DAI, which supports online search queries over the entire corpus. All 38 XML files are freely accessible by downloading a zip file[2]. The annotations are all contained in these XML files. Additionally, another zip file is available containing Python scripts which can parse the XML files for data extraction[3].

I used Python code to extract data from the 38 XML files, to provide statistical counts for the corpus, and to generate input XML files for OpenCCG. The XML file structure within all 38 files is fairly consistent. Only a few XML tags contain all the data necessary for parsing. These few XML tags include all the part-of-speech tags, the glosses, and the English translations. Separate tags contain the dominant and non-dominant hand information.

[1] `http://www.bu.edu/asllrp/ncslgr.html`—the website works best if accessed with Internet Explorer.

[2] `http://secrets.rutgers.edu/dai/xml/ncslgr-xml.zip`

[3] `http://www.bu.edu/asllrp/ncslgr-for-download/signstream-xmlparser.zip`

Some annotation errors in the XML files became apparent. These include missing `A` tags and having the incorrect `VID` value. Appendix A contains a table of changes I made to the XML files necessary to ensure correctness.

The data I extracted was saved into several .txt files. These .txt files served as the input for when I began constructing the grammar. The gloss strings from the dominant and non-dominant hand were saved in the form of the dominant hand gloss followed by the non-dominant hand. There is no way that I am aware of to extract all the gloss in its correct linear order. This means that I had to go through all the utterances which contain non-dominant hand gloss data and incorporate those into the linear order of the dominant hand gloss. This process required looking up each utterance in the NCSLGR Corpus that contained non-dominant hand gloss data. I watched the associated video clip and looked at the transcription to identify the proper placement of the non-dominant hand gloss within the utterance. Once the proper place was identified I would insert the non-dominant hand gloss into its proper placement. Complicating integration of the non-dominant and dominant hand glosses was when both hands simultaneously produced signs which overlapped. The total number of utterances in the NCSLGR Corpus containing non-dominant hand gloss data is 834. When there was no way to integrate the non-dominant and dominant hand glosses that entire utterance was omitted for this thesis. In total, 21 utterances had to be omitted; these are listed in Appendix B. The total number of utterances with non-dominant hand gloss used in the final data set for this thesis is 813.

General statistical information about the NCSLGR Corpus is provided by Neidle and Vogler (2012). The total number of utterances is given as 1,888[4] (the total is actually 1,887). The type and token counts provided in Neidle et al. (2012) are: 1,920 types, 11,861 tokens. The type and token count totals which I arrived at using AntConc[5] and Python (provided in Table 3.1) are slightly different:

[4]This is an error in reporting. The article states that 19 videos contain 885 utterances and that there are 1,002 utterances contained in other videos. Summing 885 and 1,002 totals 1,887.

[5]`https://www.laurenceanthony.net/software/antconc`

AntConc	Type count total:	2,415
	Token count total:	11,864
Python	Type count total:	2,416
	Token count total:	11,864

Table 3.1: Type and token counts

The discrepancy in the type counts is due to the fact that I tallied all the glossed data, whereas Neidle and Vogler (2012) grouped together close variants of canonical signs.

Table 3.2 is organized by the number of words in each utterance. It includes both non-dominant and dominant hand gloss counts for each utterance. In the process of integrating both glosses, some non-dominant hand gloss data had to be omitted. The process of omitting certain non-dominant hand gloss data affected the sentence length counts for the data set used in this work as referenced in the two right-hand columns of Table 3.2.

2 The Parser Components

OpenCCG is a free parsing program that implements Combinatory Categorial Grammar. It is housed at GitHub[6] and SourceForge[7] and written in Java. The installation process for OpenCCG is fairly straightforward as explained in Bozşahin et al. (2013).

Input files for OpenCCG are written in XML[8]. Four files are necessary for OpenCCG: grammar.xml, lexicon.xml, morph.xml, and rules.xml. Two other files are optional: testbed.xml and types.xml. I did not utilize the types.xml file for this thesis.

In order to populate the lexicon.xml, morph.xml, and testbed.xml files, I wrote Python scripts that read in data from .txt files containing the corpus data. Generating the XML files programmatically helped to ensure that no errors occurred if I had to manually input all the part-of-speech tags, glosses, and sentences into the XML files.

[6] `https://github.com/OpenCCG/openccg`
[7] `https://sourceforge.net/projects/openccg`
[8] Pertinent files are downloadable at: `http://linguistics.byu.edu/thesisdata/ASL-CCG.zip`.

Number of Words	NCSLGR Corpus	Thesis Data Set
1	7	7
2	49	50
3	145	168
4	303	341
5	330	372
6	251	244
7	164	184
8	159	128
9	91	78
10	90	74
11	61	67
12	48	33
13	45	20
14	22	17
15	18	20
16	19	16
17	16	13
18	15	8
19	10	5
20	10	8
21	11	3
22	6	2
23	2	1
24	2	1
25	3	2
26	2	4
27	1	0
28	2	0
29	2	0
30	2	0
31	0	0
32	0	0
33	1	0
Total Num. of Utterances	1887	1866

Table 3.2: Table of utterances by number of words

grammar.xml

This file is where the name of the grammar is specified. It also references the other file names created as part of the grammar, i.e. lexicon.xml, morph.xml, rules.xml, and testbed.xml. The references are necessary so that OpenCCG knows to use those particular files.

lexicon.xml

The lexical family types along with their syntactic functions are contained in this file. The lexicon is grouped together by lexical families. This allows a grammar to be constructed without having to assign a syntactic function to each lexical item individually. For example, all lexical items that are nouns can be assigned a syntactic function of an **n**. Lexical families can also be assigned multiple syntactic functions: the Adjective lexical family can take both categories, **np/n** and **np\n** (see Figure 3.1).

Each lexicon entry is built using XML hierarchical structure (see Figure 3.1). The top-level `<family>` tag has a `pos` attribute whose value links the lexical family to the morph.xml file containing all the lexical items belonging to this family. Nested inside is the `<entry>` tag. The `<atomcat>` tag assigns an atomic syntactic category. A lexical family can have a complex category, like that of a transitive verb. This requires using a `<complexcat>` tag along with a directional `<slash>` tag embedded, as shown in Figure 3.2.

morph.xml

The morph.xml file contains all the individual gloss lexemes, listed as in Figure 3.3. The `pos` attribute's value references the word's part-of-speech linking it to its syntactic function as defined in the lexicon.xml file.

rules.xml

The rules.xml file contains the rules which specify how syntactic elements are allowed to combine. Four rules that we have previously discussed are already built into the OpenCCG distribution: functional application, harmonic composition, backward crossed composition, and type raising. Any unary type rule required by the grammar can be specified here.

```
<family name="ADJECTIVE" pos="Adjective">
  <entry name="np/n">
    <complexcat>
      <atomcat type="np"/>
      <slash dir="/" mode="&gt;"/>
      <atomcat type="n"/>
    </complexcat>
  </entry>

<family name="ADJECTIVE" pos="Adjective">
  <entry name="np\n">
    <complexcat>
      <atomcat type="np"/>
      <slash dir="\" mode="&lt;"/>
      <atomcat type="n"/>
    </complexcat>
  </entry>
</family>
```

Figure 3.1: An example of a lexical family having two syntactic functions

```
<family name="VERB-TRANSITIVE" pos="Verb-trans">
  <entry name="(s\np)/np">
    <complexcat>
      <atomcat type="s"/>
      <slash dir="\" mode="&lt;"/>
      <atomcat type="np"/>
      <slash dir="/" mode="&gt;"/>
      <atomcat type="np"/>
    </complexcat>
  </entry>
</family>
```

Figure 3.2: An example of a complex category

```
<!-- ========== NOUN ========== -->
<entry pos="Noun" word="#BANKS"/>
<entry pos="Noun" word="#BUS"/>
<entry pos="Noun" word="#CAR"/>
<entry pos="Noun" word="#CLUB"/>
<entry pos="Noun" word="#CO"/>
<entry pos="Noun" word="#DOG"/>
<entry pos="Noun" word="#DRUGS"/>
<entry pos="Noun" word="#EMAIL"/>

<!-- ========= VERB-TRANSITIVE ========= -->
<entry pos="Verb-trans" word="#DO"/>
<entry pos="Verb-trans" word="#DO+++"/>
<entry pos="Verb-trans" word="#FIX"/>
<entry pos="Verb-trans" word="i*BLAME*j"/>
```

Figure 3.3: Example of entries in the morph.xml file

testbed.xml

The testbed.xml file is an optional file. It contains all the sentences that a grammar is supposed to be able to parse. This file is needed if all the sentences are to be parsed in batch mode using OpenCCG. The testbed.xml file used for this thesis contains the 1,866 utterances extracted from the NCSLGR Corpus.

3 Unary Rules

The following list are rules specific to American Sign Language that I created for the grammar.

1. Noun-to-Noun-Phrase
2. Topic Raising
3. Subject and Object pro-drop
4. Noun Phrase to Predicate Function
5. Intransitive Topicalized Verb
6. Sentential Conjunction

Noun-to-Noun-Phrase

The Noun-to-Noun-Phrase rule changes the atomic category of an **n** to create an **np**. Essen-

tially this states that a noun that by itself cannot combine with any other lexical item can change its category to become a noun phrase. This allows singleton nouns to be arguments of predicates. Figure 3.4 is an example that shows application of this rule. WOLF is an unmodified noun whose atomic category is **n**. ARRIVE is an intransitive verb whose category is a complex function, **s\np**. WOLF with its atomic **n** category cannot serve as the argument for ARRIVE because WOLF does not have the appropriate argument the predicate requires. In this instance, the unary Noun-to-Noun-Phrase[9] rule changes the **n** to **np** thus allowing WOLF to be the argument of the predicate.

$$\dfrac{\dfrac{\dfrac{\text{WOLF}}{\mathsf{n}}}{\mathsf{np}}\downarrow \quad \dfrac{\text{ARRIVE}}{\mathsf{s\backslash np}}}{\mathsf{s}}<$$

"Wolf arrived."

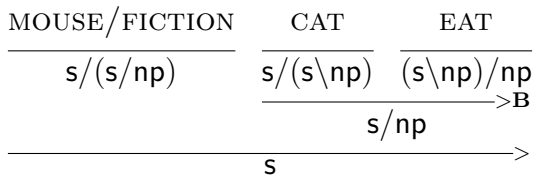
Figure 3.4: A noun changed to a noun phrase.

Topic Raising

Topic Raising is for when objects are topicalized. This rule is similar to the Type-Raising rule in that a noun becomes a functor over a predicate. Figure 3.5 is an example of a parse of a sentence containing a topicalized object. MOUSE/FICTION is the direct object of the verb EAT. It has moved from its canonical position after the verb to the front of the sentence. This operation changes the underlying syntactic order from SVO to OSV. When this pattern occurs, the topicalized object has its canonical atomic category type-changed from an **np** to **s/(s/np)**. This functional category is used so that the topicalized object may combine with the parts of the parse which have already composed. The subject and the predicate first compose via forward composition. This leaves the predicate still searching for an object. The example in Figure 3.5 shows that CAT and EAT have composed to form a constituent with

[9]The ↓ indicates a type-change operation. The operations indicated by ↓ are listed in Table 2.3. They are the unary rules specific to this grammar.

the category of **s/np**. The subject-verb constituent category serves as the argument to the now topic-raised object, thereby a proper parse is obtained.



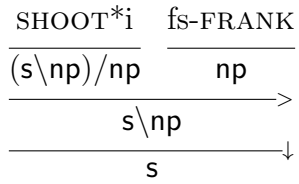
"As for the mouse, the cat ate it."

Figure 3.5: Topicalization Example

Subject and Object pro-drop

ASL is a pro-drop language. Either a subject or an object pronoun may be dropped. In order to properly obtain a parse when either subject pro-drop or object pro-drop occurs, the predicate function category is affected.

Figure 3.6 is a sentence which contains a dropped subject pronoun. The transitive verb, SHOOT*i, consumes its argument, fs-FRANK, to produce the predicate category **s\np**. The predicate is searching for a subject to consume as its argument. The subject, however, in this instance has been dropped. The Subject pro-drop rule changes the predicate category from a complex function seeking a subject argument to an atomic category of **s** when a subject with a category of **np** is not present.

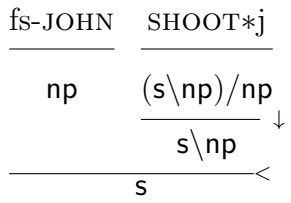


"(He/She) shoots Frank."

Figure 3.6: A parsed sentence with subject pro-drop.

The Object pro-drop rule functions very similarly to the Subject pro-drop rule. However, instead of a dropped subject, the object has been dropped. The transitive verb's

category changes from **s\np/np** to **s\np**. This rule takes effect when an object with a type category of **np** does not follow the transitive verb (see Figure 3.7).

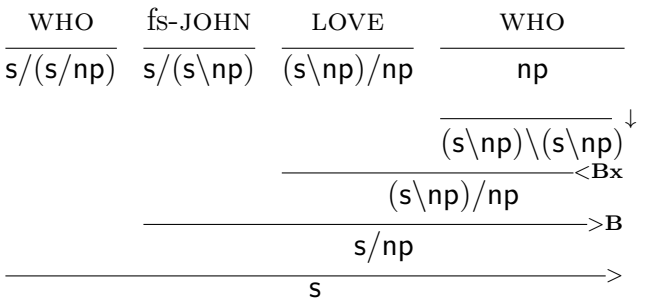


"John shot him/her/it."

Figure 3.7: A parsed sentence with object pro-drop.

Noun Phrase to Predicate Function—a.k.a. twin

The Noun Phrase to Predicate Function, also the *twin*, rule is operative for when wh-words are duplicated at the end of a sentence. Figure 3.8 shows how the WHO at the end of the sentence is type-changed from its atomic category **np** into a complex functional category of **s\np\(s\np)**. The changed functional category allows the latter wh-word to compose with the predicate via the Backwards Crossed Composition rule (see Section 3.1).

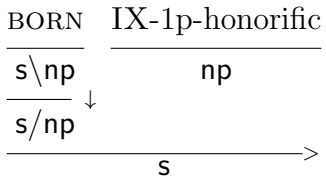


"Who, who does John love?"

Figure 3.8: A parsed sentence with a wh-twin.

Intransitive Topicalized Verb

Topicalized intransitive verbs alter the syntactic ordering of sentences from SV to VS. To reflect this change the functional category of the predicate must flip its slash direction for its accepting argument. Instead of a backslash indicating that the subject is to the left of the verb, the slash changes direction to allow for the subject to its right (Figure 3.9).

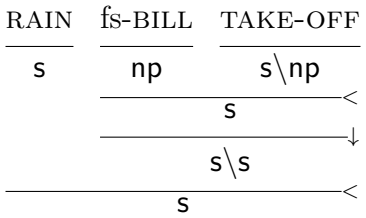


"And then I was born."

Figure 3.9: A parsed sentence with a topicalized intransitive verb.

Sentential Conjunction

When two clauses derive to an atomic constituent **s** but a final composite constituent has yet to be obtained, one of the **s** constituents may change its category to a functional category **s\s** (see Figure 3.10). There are essentially two clauses in this one sentence: RAIN and fs-BILL TAKE-OFF. In ASL, RAIN does not need a pleonastic subject like it does in the English sentence "It will rain."; it can stand as an independent clause. In the context of the sentence in Figure 3.10 RAIN has an atomic category of **s**, and fs-BILL TAKE-OFF comes together to form a constituent of category **s**. The constituent category then changes into a function which searches leftward for an **s**. Once the argument is identified the final derivation creates a constituent of category **s**.



"If it rains, Bill will leave."

Figure 3.10: Two sentential clauses joined by Sentential Conjunction.

4 Running OpenCCG

The documentation provided from downloading OpenCCG provides detailed steps on how to run the program (Bozşahin et al., 2013). In order to run OpenCCG, the grammar.xml file is loaded from the command prompt. The **tccg>** prompt signals that the program is ready

to accept strings for parsing. Two other commands that I used extensively are: **ccg-test** and **wccg**. The **ccg-test** command will run all the test sentences in the testbed.xml file and return the number of parses generated for each sentence. If no parse was generated, then FAIL is returned for that sentence. The **wccg** command also iterates over the testbed.xml file. This command, however, returns the generated derivations of each parse for each sentence in the testbed.xml file. The outputs from **ccg-test** and **wccg** can be saved to .txt files.

5 Development and Refinement

During the build, I initially relied upon **ccg-test** to determine coverage. I wanted to ensure that I was able to obtain as many parses as possible. Once I began to plateau on the number of sentences parsed, I then went back and refined the lexical-syntactic categories. I would look at sentences that were not already parsed to determine why those would not parse. I would often diagram parses for those sentences by hand. Once to my satisfaction I was able to assign proper lexical-syntactic categories to obtain a correct parse, I would then add the syntactic lexical category to the lexical family to which a particular lexical item belonged.

For example, compound nouns are problematic because of one noun being a modifier of another. The head and its dependent need to combine as one constituent noun phrase. The compound noun object TURKEY SANDWICH in Example (1) ought to behave as a single unit instead of two independent nouns. In order to have TURKEY modify SANDWICH, it needs to be listed in a closed class lexical family in the lexicon.xml file.

(1) 'TURKEY SANDWICH fs-JOHN DEVOUR.

'As for the turkey sandwich, John devoured it.'

As I perused the generated output and came across any composition error, I would attempt to rectify that error. I would either have to add additional lexical-syntactic categories to the

particular lexical family a word belonged to, add the lexical item to the morph.xml file, or build a unary rule appropriate for the grammar in the rules.xml file.

Chapter 4

Results

This chapter shows examples of sentences parsed by the system as reported during the evaluation. The first three sections look at the parses of sentence types (intransitive, transitive, and ditransitive). The later sections examine types of constructions found within sentences, such as wh-words, relative clauses, topicalization, and passives. The examples in this chapter are all parsed sentences by OpenCCG of sentences taken from the NCSLGR Corpus unless specifically otherwise stated.

1 Intransitives

Basic intransitive sentences having only a subject and a verb are the easiest to parse. The subject's **np** category will always serve as the input argument for the predicate (see Figure 4.1).

$$\dfrac{\dfrac{\text{WOLF}}{\mathsf{np}} \quad \dfrac{\text{ARRIVE}}{\mathsf{s\backslash np}}}{\mathsf{s}}\,{<}$$

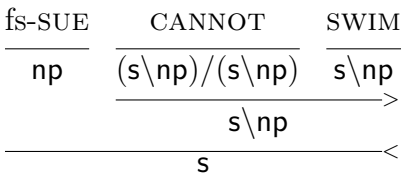
"Wolf arrived."

Figure 4.1: Sentence 1735: Intransitive and Backward Application

The subject, WOLF, has the atomic category **np** and the verb, ARRIVE, has a functional category of **s\np**. The verb consumes its argument via backward application.

The next example, Figure 4.2, is a bit more complicated than the last. In this example, the intransitive verb is being negated by CANNOT. The negative and the verb must first

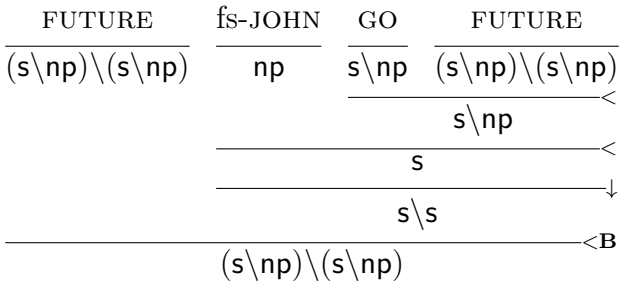
combine to form a predicate category that then seeks for its argument. CANNOT has a complex functional category that allows it to consume the lexical-syntactic category of SWIM. This process is accomplished by forward application. Once all the arguments of each function have been consumed, the resulting final derived category is **s**.



"Sue can't swim."

Figure 4.2: Sentence 1154: Intransitive with negation

The next example, Figure 4.3, is a construction where two grammaticalized tense lexemes are present. FUTURE occurs at the beginning and at the end of the sentence, showing how complex sentence constructions can be in ASL. The parse generated by the system is not accurate. The category assigned to FUTURE is not correct for the first occurrence but is correct for the second. In order to obtain a proper parse, it would be necessary to somehow link the sentence initial FUTURE with the predicate.

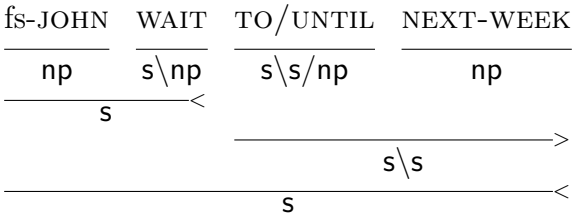


"In the future, John will go."

Figure 4.3: Sentence 931: Intransitive with tense (incorrect)

The parse for sentence 1230 in Figure 4.4 shows how a preposition may compose during the derivational process. The subject and the verb first compose resulting in a final constituent category of **s**. This serves as the input to the prepositional phrase constituent. Before this category is derived though, TO/UNTIL and NEXT-WEEK must first combine.

The lexical-syntactic category assigned to TO/UNTIL is **s\(s/np)** and NEXT-WEEK is **np**. Once these two combine by forward application the composed constituent receives category **s\s**, allowing the constituent fs-JOHN WAIT to compose with the constituent TO/UNTIL NEXT-WEEK.

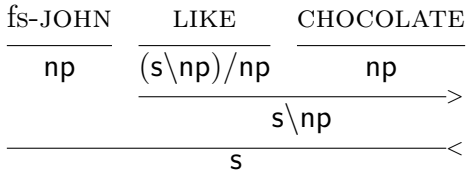


"John is waiting until next week."

Figure 4.4: Sentence 1230: Intransitive with preposition

2 Transitives

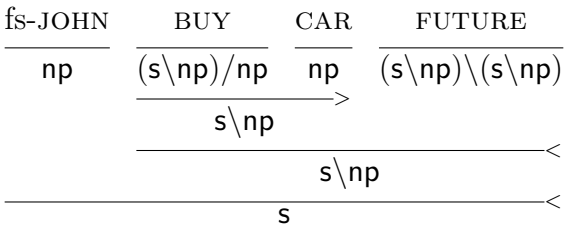
When a simple transitive sentence has a subject, verb, and object the parse is straightforward (see Figure 4.5).



"John likes chocolate."

Figure 4.5: Sentence 1320: Transitive

More complicated structures involving tense, negation, aspect and modals increase the complexity of a correct parse (see Figure 4.6). FUTURE occurs sentence finally and has the lexical-syntactic type category of **(s\np)\(s\np)**. This function is seeking as its argument a predicate. The verb and the direct object in the sentence compose by forward application to form the constituent predicate category **s\np** which serves as input for assigned category to FUTURE. The final derived constituent **s** is obtained once the constituent predicate category consumes its subject **np** argument.

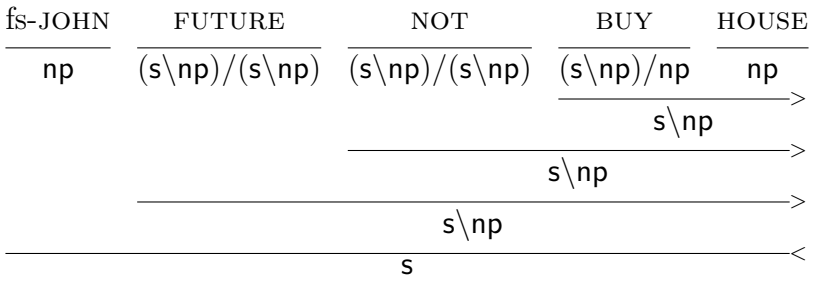


"John will buy a car."

Figure 4.6: Sentence 918: Transitive with tense

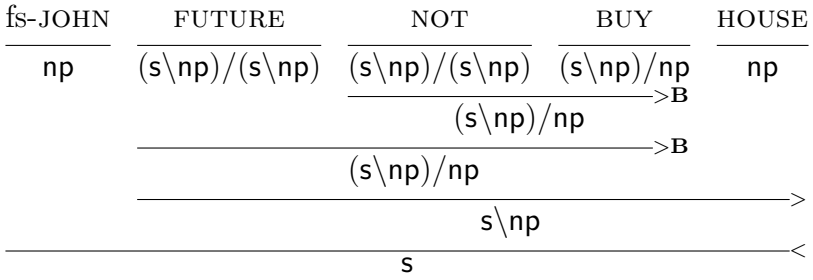
Figure 4.7 is an example of more complex sentence structures. The parsed sentence has tense and negation lexical items. FUTURE and NOT both have assigned functions searching for predicates as their argument. Both functions are looking to consume the same predicate argument category type, **(s\np)**. Each lexeme is able to consume this type of argument but not necessarily in the proper manner. BUY and HOUSE first combine forming a constituent with a predicate category, then NOT combines by forward application with that constituent, and finally FUTURE accepts the composed predicate constituents as its argument. This step-by-step process in turn yields a predicate category which takes as its argument the subject **np**.

The case can be made that the derivational process is out of order for this sentence. The sequence of composition could be that NOT and BUY compose via forward composition to derive a constituent category of **(s\np)/np**. FUTURE could then compose with that constituent category via forward composition yielding the same complex constituent category. The predicate would then consume its direct object argument by forward composition and then its subject argument by Backward Composition. Figure 4.8 shows this proposed derivation.



"John, in the future, is not buying a house."

Figure 4.7: Sentence 917: Transitive with tense and negation (incorrect)

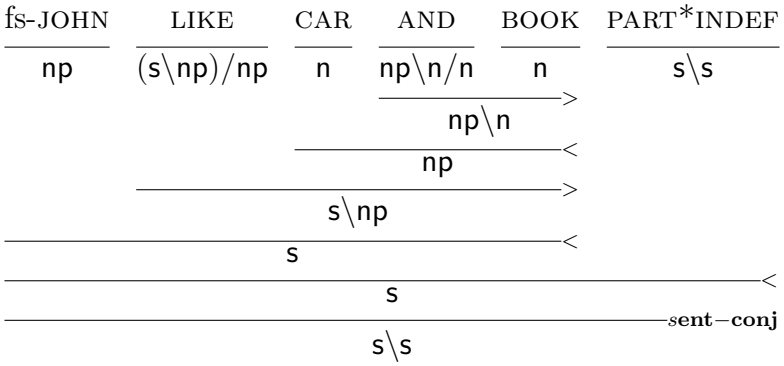


"John, in the future, is not buying a house."

Figure 4.8: Alternative parse for sentence 917: Transitive with tense and negation

The last transitive example that will be analyzed contains a conjunct. Conjunction is labeled as (X\X)/X in the literature (Steedman, 2014). In this rule the categories to the left and right of the conjunct are mapped into the appropriate conjunction category. Unfortunately, this option is not available in OpenCCG. In order for conjunction to properly apply, one first needs to know what categories are going to be conjoined. The parsed sentence in Figure 4.9 conjoins two noun phrases. Steedman (2014) presents different types of coordination that are far more complicated than the example provided here. In the sentence fs-JOHN LIKE CAR AND BOOK PART*INDEF both CAR and BOOK are the arguments for LIKE. The functional category for AND allows it to combine both nouns together to form a noun

phrase. The resulting constituent serves as the argument to LIKE. The derivation continues as other transitive constructions until a final **s** is obtained for the parse[12].



"John likes cars and books."

Figure 4.9: Sentence 643: Transitive with conjunction

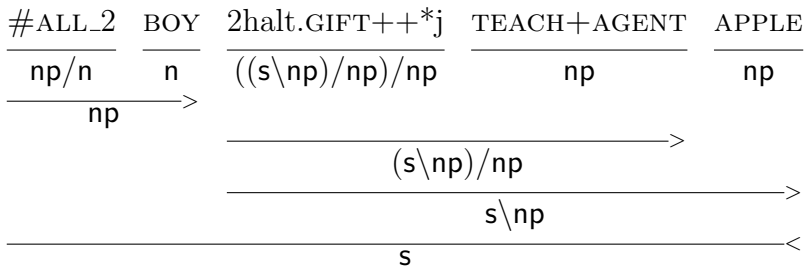
3 Ditransitives

Ditransitive sentences behave similarly to transitive sentences. Ditransitive verbs, however, require two object complements. The lexical-syntactic type category for ditransitive verbs is **(s\np)/np/np**. This functional category allows for the indirect object to first compose with the verb via forward application and then the direct object by the same process. Once the predicate category **s\np** is formed, the subject is consumed as the predicate's last argument (see Figure 4.10).

The ditransitive sentence in Figure 4.11 contains a topicalized direct object, which is not fronted to the beginning of the sentence, but rather just before the verb. The position of the direct object causes OpenCCG to misinterpret CHOCOLATE as being the subject and fs-JOHN as the direct object. When a subject is type-raised from an **np** to a category that functions over a predicate, the now category is **s/(s\np)** and the application label of $_T$ is shown. Topicalized objects raise from having an atomic category of **np** to having a complex

[1]Due to a spurious subsequent step, the final derived constituent category in Figure 4.9 is **s\s**. This is because OpenCCG applied a type-changing rule inappropriately. Apparently this was done because another parse yields a final **s** and OpenCCG only permits one parse to have a final **s** category.

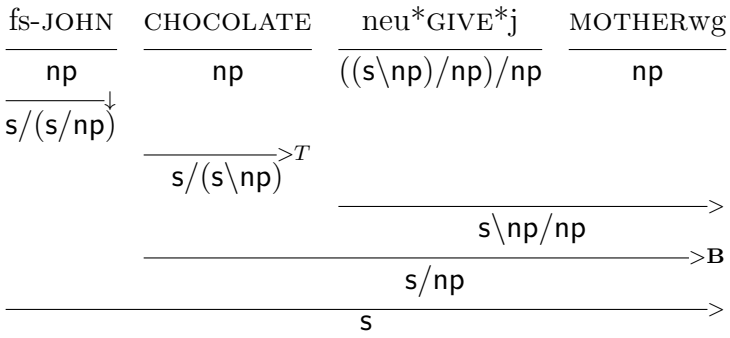
[2]The category for AND can also be **np\np/np**.



"All the boys each give the teacher an apple."

Figure 4.10: Sentence 894: Ditransitive

category of **s/(s/np)**. The down arrow indicates that the application of the appropriate rule has taken effect. The subject and topicalized object receive the other's expected raised category types, but this does not affect the constituent composition. Once the verb consumes its indirect object argument, the direct object composes with the predicate by way of forward composition. The subject is then able to form with the larger constituent to form a final derived constituent category of **s**.



"He, John, gave his mother chocolate."

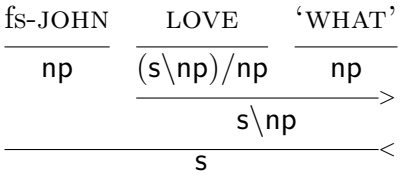
Figure 4.11: Sentence 1095: Ditransitive

4 Wh-words

One axiom true to all categorial grammars is that they "[avoid] destructive devices such as movement or deletion rules which characterize transformational grammars"(Wood, 1993, p. 4). The linear order in which the string appears is the way in which it is to be parsed. This

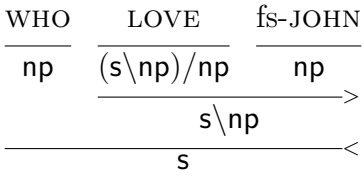
means that the positions taken by Neidle et al. (1998b) and Petronio and Lillo-Martin (1997) are irrelevant in analyzing wh-words in the context of CCG. Therefore, if a wh-word appears sentence initially, finally, or even as a double the approach to parsing them is the same.

When wh-words function as either a subject or as an object of the sentence, the derived parses for those sentences is straightforward. Each of the wh-words in the sentences shown in Figures 4.12, 4.13, and 4.14 all have the lexical-syntactic category **np**.



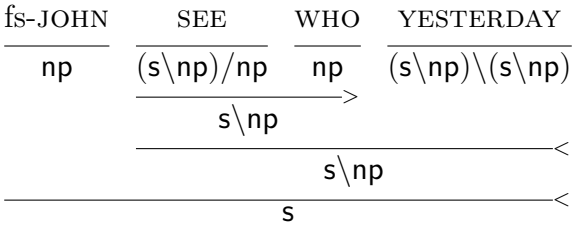
"What does John love?"

Figure 4.12: Sentence 1109 with wh-word object



"Who loves John?"

Figure 4.13: Sentence 1142 with wh-word subject

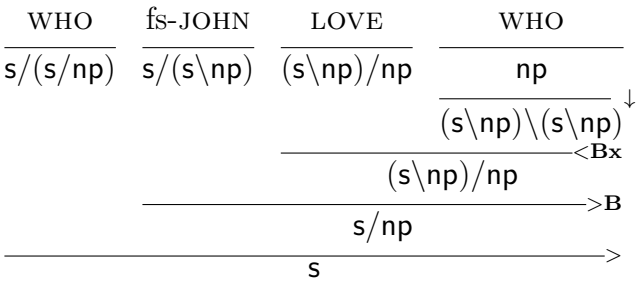


"Who did John see yesterday?"

Figure 4.14: Sentence 1108 with wh-word object

More interesting cases of wh-words involve wh-word doubles. The parsed sentence in Figure 4.15 has a wh-word double construction. WHO appears in the sentence's initial and

final positions. The difficulty with a construction of this type is managing to incorporate the wh-words. This example not only has wh-doubling but also has a topicalized object. The first WHO is a topicalized object and the second WHO is the double.



"Who, who does John love?"

Figure 4.15: Sentence 963 with wh-word double

Like any topicalized object, it undergoes a category change[3], changing from an atomic **np** into a complex category that may function over predicates. The specific *twin* rule that is applied to the wh-double allows it to functionally compose with the predicate. The complex category type of **(s\np)\(s\np)** looks for a predicate type category to its left to consume as its argument. Since LOVE has a lexical-syntactic category of **(s\np)/np** it can functionally compose with WHO. The predicate and the *twin* complex category compose via the Backwards Crossed Composition rule. Once the two functional categories compose the then type-raised subject composes via forward composition; finally, the topicalized WHO consumes its argument resulting in a final derivation of an **s**.

5 Relative Clauses

Figure 4.16 is an example of how an English sentence containing an externally headed relative clause can be represented in pure categorial grammar. The lexeme **that** introduces the relative clause.

[3]The PDF output generated by OpenCCG often omits category changes because of rule operations. The complex functional categories seen in Figure 4.15 assigned to the lexical items are the result of unseen rule change operations.

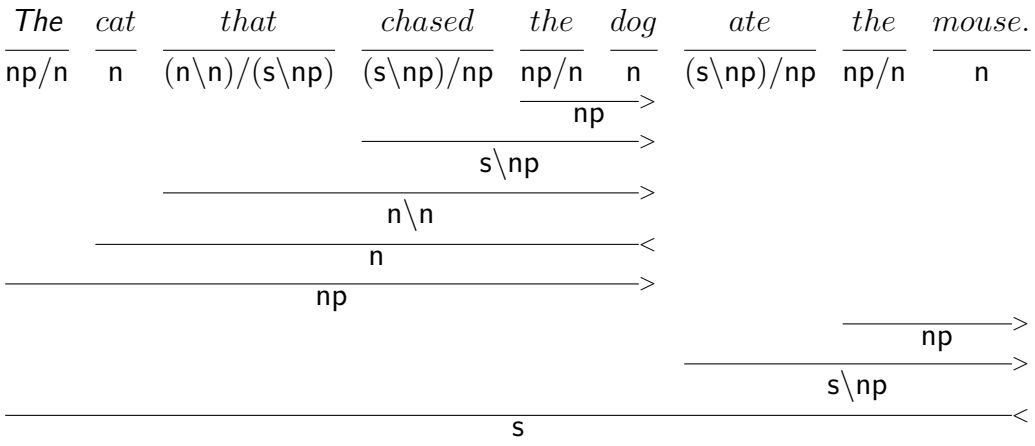
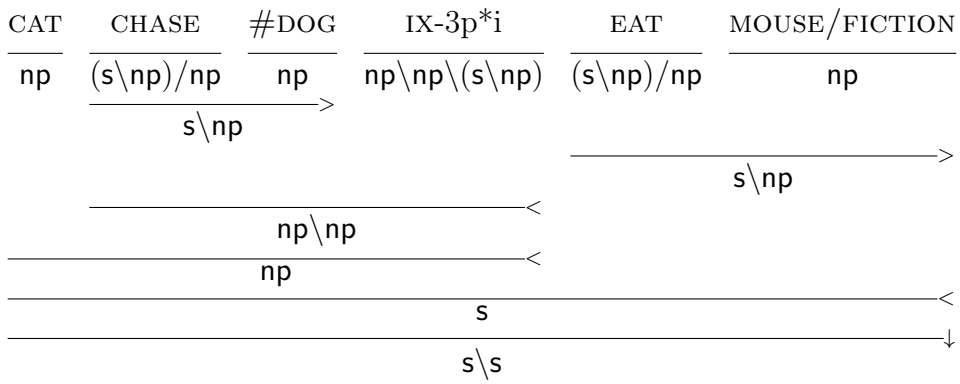


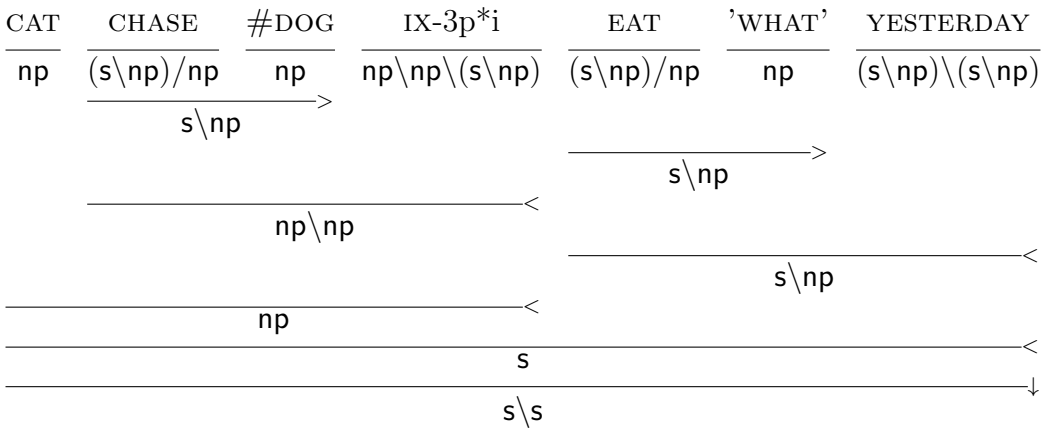
Figure 4.16: English Relative Clause

Relative clauses for ASL can be represented just as straightforwardly as relative clauses in English. Figures 4.17 and 4.18 show the derivational parse for two relative clauses in ASL. They each have the same relative pronoun IX-3p*i. The lexical-syntactic category for each pronoun is **np\np\(s\np)**. First the relative clause's predicate forms yielding **s\np** before that constituent is consumed as the argument of the relative pronoun. Once these two combine, the new constituent is a **np\np**. This category combines with the subject, CAT, to form a subject **np**. Though IX-3p*i is a pronoun that could serve independently as the pronominal subject of the matrix clause, treating it as a relative pronoun enables the matrix clause to combine with its dependent relative clause.



"The cat that chased the dog ate the mouse."

Figure 4.17: Parse for sentence 1270: Relative Clause



"What did the cat that chased the dog eat yesterday?"

Figure 4.18: Parse for sentence 1287: Relative Clause

6 Topicalization

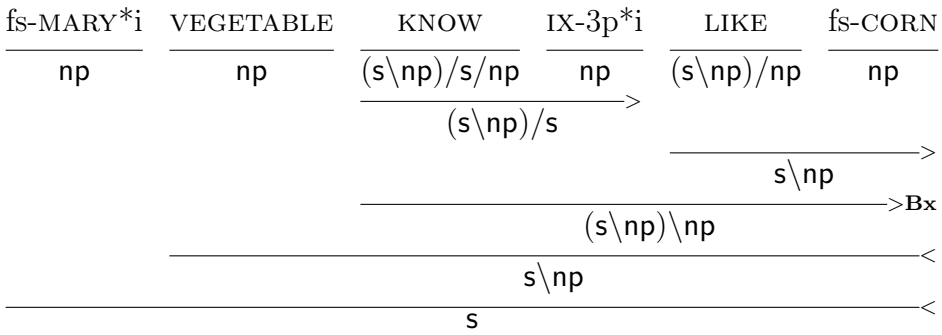
As described earlier, ASL has two types of topicalized constituents: topics that are base-generated and topics that have been moved.

6.1 Base-Generated Topics

Sentence 932 (Figure 4.19)shows an incorrect parse generated by the system. The two base-generated topics do not reflect topicalization. Also, KNOW was assigned the incorrect category; it should be **(s\np)/s**. In order to obtain a correct derivational parse for this sentence, the system would need additional rules. First, the base-generated topics will need a category change from **np** to **s/s**, declaring that the noun can be part of a sentence if it can find a sentence to its right. Figure 4.20 is a proposed way to parse this particular sentence.

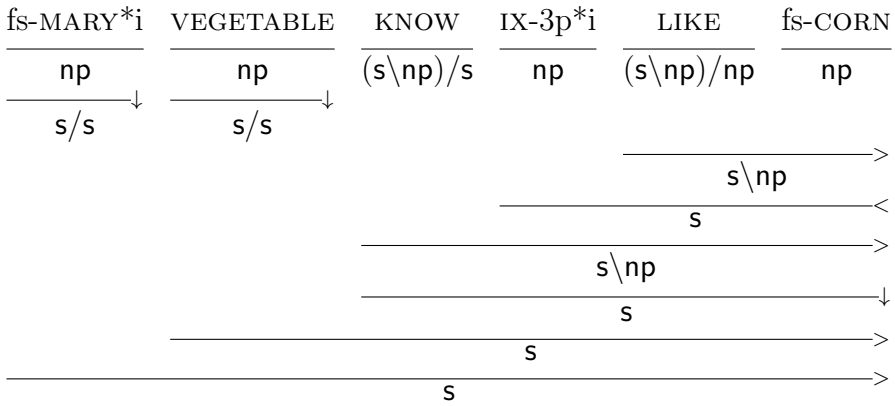
The parsed sentence in Figure 4.21 contains a base-generated topic and an overtly expressed direct object. This example does not fit into either pattern for base-generated topics: it contains a duplication of the same item.

The constituency composition for the parse in Figure 4.21 is fine. The correct constituents compose together in the order that they are meant to. However, lexical-syntactic categories are not correctly assigned. TEACH+AGENT should have been assigned an atomic



"As for Mary and vegetables, (I) know she likes corn."

Figure 4.19: Sentence 932: Topicalization (incorrect)

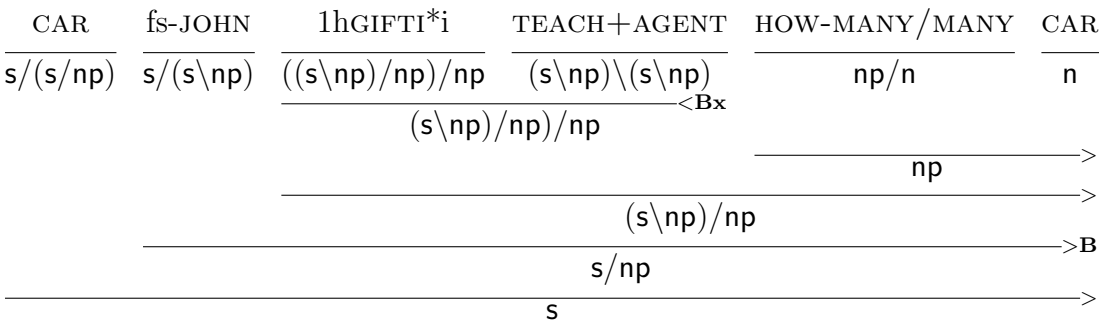


"As for Mary and vegetables, (I) know she likes corn."

Figure 4.20: Alternative parse for sentence 932: Topicalization

np. Instead, the complex function that it was assigned by OpenCCG allowed it to compose with the verb's lexical-syntactic category by Backwards Crossed Composition. The composed constituent category though is the same as a category assigned to a ditransitive verb, **((s\np)/np)/np**. This complex category consumes the direct object noun phrase resulting in the constituent category of **s\np/np**. The constituent category serves as input for the type-raised subject by forward application. Finally, the base-generated topic composes with the rest of the string also by forward application.

Figure 4.22 is the same sentence but this time parsed by hand utilizing the proposed rule that changes the base-generated topic **np** category to **s/s**. Each constituent composes correctly by only forward application. The base-generated topic, CAR is able to join with the rest of the string by accepting the derived constituent category **s** as its argument input.



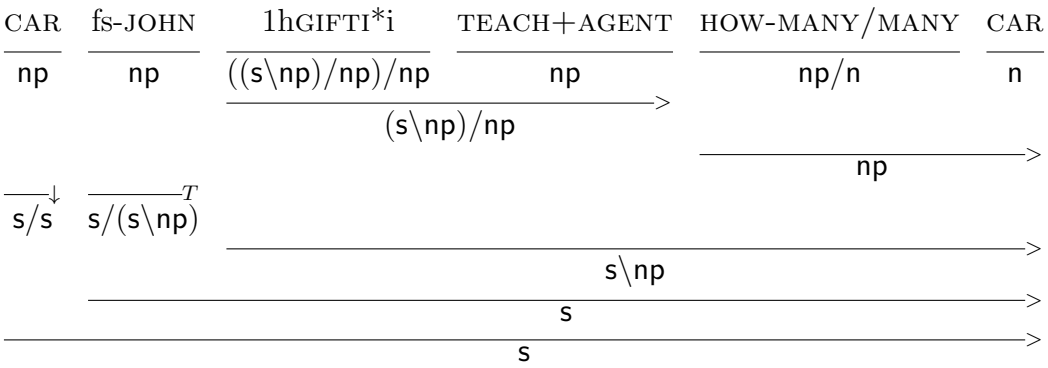
"The cars Jon gave the teacher, how many cars was it?"

Figure 4.21: Sentence 724: Topicalization (incorrect)

6.2 Moved Topics

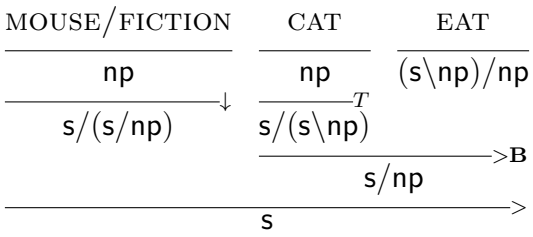
Sentences with sentential-initial moved topics are easier to parse than sentences having base-generated topics. Figure 4.23 is an example of a sentence with a topicalized object as parsed by the system.

The object, MOUSE/FICTION has been fronted. Its category has been type-changed from an **np** to a complex function of **s/(s/np)**. CAT has had its category type-raised from an **np** to **s/(s\np)**. CAT composes with the verb by forward composition. The topicalized



"The cars Jon gave the teacher, how many cars was it?"

Figure 4.22: Alternative parse for sentence 724: Topicalization

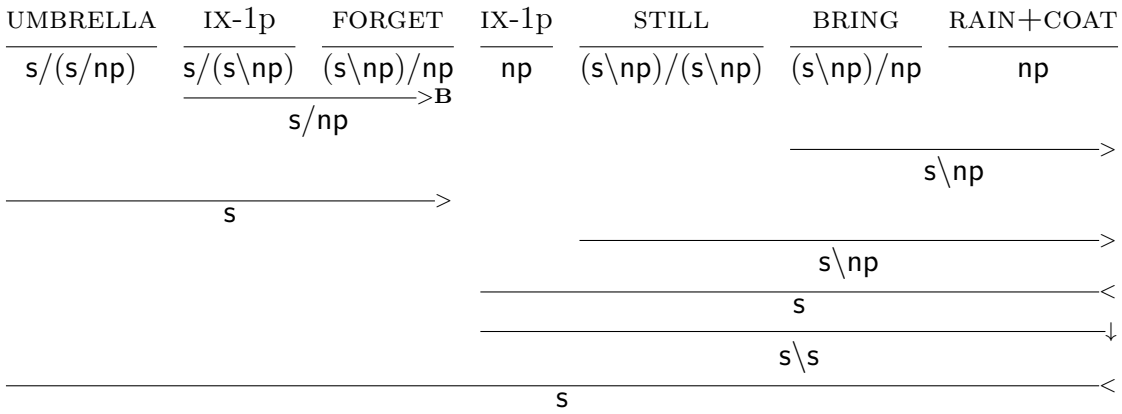


"As for the mouse, the cat ate it."

Figure 4.23: Sentence 1184: Transitive with topicalization

object takes as input the constituent category from CAT EAT to yield a final sentence with category **s**.

Figures 4.24 and 4.25 also contain topicalized objects[4]. Sentence 1238 (Figure 4.24) has two independent clauses that are conjoined even though there is not an explicit conjunction. The object of the first clause is what has been topicalized. The composition for this clause follows the same process as the parse in Figure 4.23. The second clause composes together by forward application until a final **s** is derived. This **s** then changes to **s\s** by a rule specific to this grammar allowing it to compose with the first clause, thereby producing a final derived constituent **s** for the entire sentence.

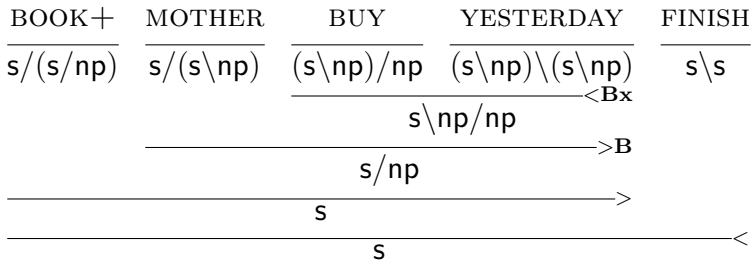


"As for my umbrella, I forgot it, but I still brought my raincoat."

Figure 4.24: Sentence 1238: Sentential conjunction with topicalization

The topicalized object in Sentence 768 (Figure 4.25)has been type-changed from its atomic **np** category to the complex functional category shown. The subject has been type-raised to its complex category. Derivation for this parse is as follows: (1) the verb and adverb compose by backward crossed composition; (2) the subject then composes by forward composition, (3) the topicalized object then combines by forward application; (4) the aspect lexeme composes by backward application.

[4]The categories for the topicalized object and for the subject are type-raised categories from an atomic **np**. OpenCCG does not always show the steps taken to get to these categories.



"As for the book, mother bought it yesterday; that's done."

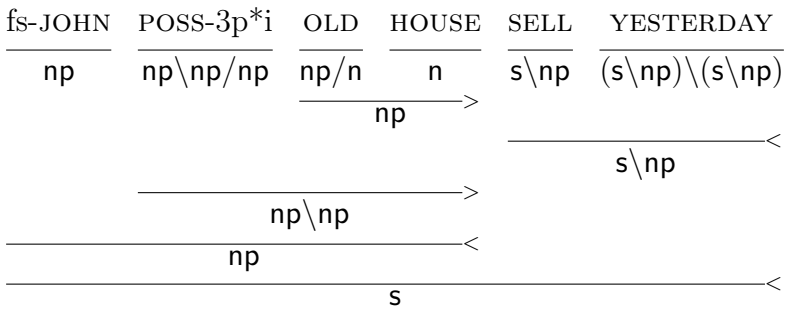
Figure 4.25: Sentence 768: Topicalization with tense

7 Passives

Figures 4.26, 4.27, and 4.28 show parses for passive sentences. One requirement for passive constructions in ASL is that "the agent is demoted, which often means that the agent is not mentioned"(Janzen et al., 2001). In the first two examples there are no subject-agents but the third has a demoted subject-agent. The demoted subject is introduced with fs-BY.

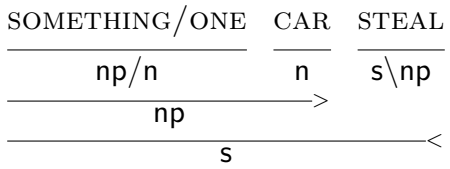
The parsed sentences in these examples each follow the rules for constituent composition. Sentence 969 (Figure 4.26) has a complex noun phrase in subject position. The ASL possessor, POSS-3p*i, has the category **np\np/np** which requires two arguments, a possessor and a possessee (in this case, fs-JOHN and OLD HOUSE respectively). The **np** subject of the sentence does not fulfill the role of the agent of the sentence, but is instead the theme. There is no mention of who sold the house. The lack of an overt subject-agent in sentence 985 (Figure 4.27) is also what makes that sentence passive. Sentence 586 (Figure 4.28) is different from the previous passive examples.

Sentence 586 has more of an "English-like"' construction to it than pure ASL. The English influence upon the sentence can be seen with the use of fs-BY. The subject-agent, COP+, has been demoted to an oblique prepositional phrase introduced with fs-BY. ASL does not have a sign for BY having a prepositional function as in this example. That is why the gloss is affixed with an fs- tag indicating that it was finger-spelled. The oblique prepositional



"John's old house was sold yesterday."

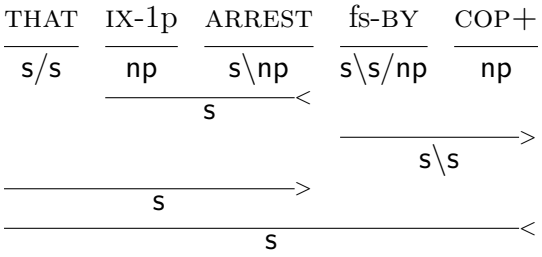
Figure 4.26: Sentence 969: Passive



"Someone's car was stolen."

Figure 4.27: Sentence 985: Passive

phrase glossed as fs-BY has a lexical-syntactic category which allows it to consume COP+ as its **np** argument and then combine with the independent clause IX-1p ARREST.



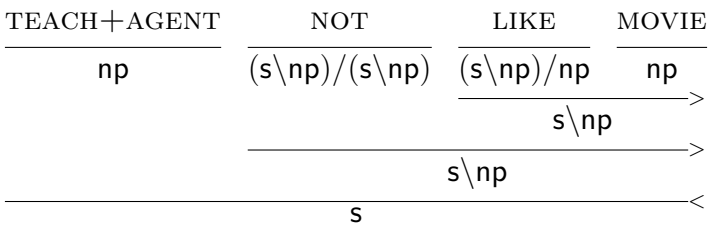
"...that I had been arrested by the police."

Figure 4.28: Sentence 586: Passive with an oblique

8 Negation

Figures 4.29 are 4.30 are examples of sentences containing negation. The categories assigned to negation are: **s\np/(s\np)** and **s\np\(s\np)**. These allow for the negative lexical items to compose with predicates by either forward or backward application.

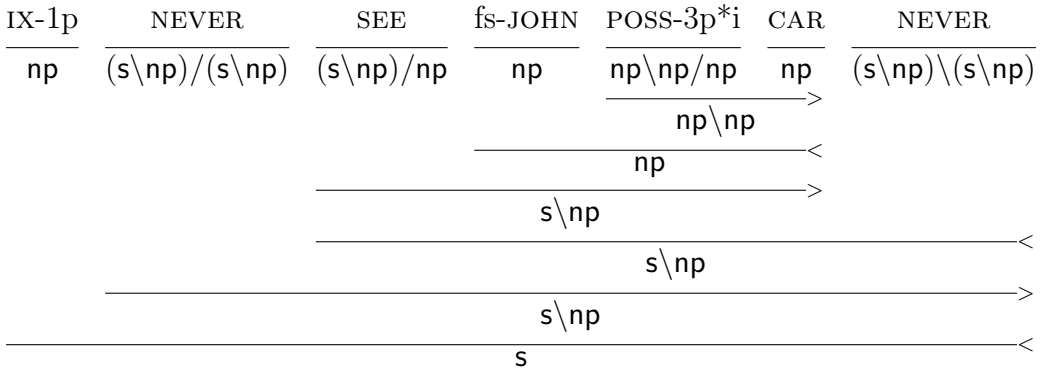
The negative lexeme in sentence 646 (Figure 4.29) negates the predicate LIKE MOVIE. NOT accepts the predicate constituent category **s\np** as its input argument. The newly formed constituent predicate has the same functional category of **s\np** which allows it to consume the subject as its argument to yield a final derivation for the sentence of **s**.



"The teacher does not like movies."

Figure 4.29: Sentence 646: Negation

Sentence 794 (Figure 4.30) has two negation lexical items. The double negative surrounds the sentence's predicate, which is why negation also has the **s\np\(s\np)** functional category. In the sentence, the predicate SEE fs-JOHN POSS-3p*i CAR composes to yield a predicate **s\np** that serves as the input for sentence final NEVER. The first negative lexical item accepts the predicate as its argument by way of forward application and then the subject via backward application.



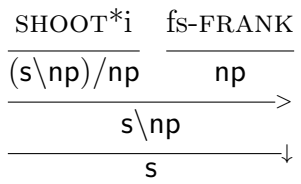
"I've never seen John's car."

Figure 4.30: Sentence 794: Double negation

9 pro-drop

The output sentences in Figures 4.31, and 4.32 will serve as examples to illustrate pro-drop in ASL as discussed in section Chapter 2. Both sentences have the same verb.

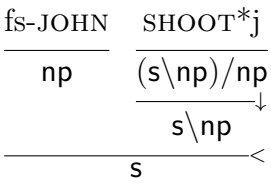
In sentence 922 (Figure 4.31), the subject has been dropped. SHOOT*i first combines with its direct object argument, forming a predicate category **s\np**. However, there is no available subject argument to consume. In such cases, the Subject pro-drop rule takes effect, changing **s\np** to an **s** with the indication that the rule has applied.



"(He/She) shoots Frank."

Figure 4.31: Sentence 922: Subject pro-drop

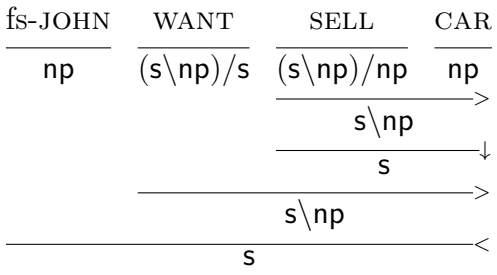
When the object of the sentence is dropped, a change to the verb's category occurs: the **(s\np)/np** will drop its direct object **np** requirement to become a predicate of category **s\np**. The output parse in Figure 4.32 illustrates the steps undertaken to obtain a final derived constituent category of **s**.



"John shot him/her/it."

Figure 4.32: Sentence 1307: Object pro-drop

The parsed sentence in Figure 4.33 contains a subordinate clause that has a PRO subject. The subordinate clause composes with the matrix clause after the Subject pro-drop rule changes the predicate category to an **s**. The matrix verb, WANT, has a lexical-syntactic category of **s\np/s**. This category type requires a sentential complement as its argument.



"John wants to sell (his/a) car."

Figure 4.33: Sentence 944: pro-drop with subject control

Understanding how the null pronominal gets interpreted requires a brief discussion on the topic of control. Certain verbs require a clausal complement; *want* is one of them in certain contexts. The subject of the clausal complement may be omitted giving rise to PRO. The subject of the matrix clause determines the interpretation as to whom or what the PRO is referring to[5]. This is easier to see in a different notation schema than derivational parses in CCG. The same sentence is in Example (1) but this time in bracketed notation.

(1) [fs-JOHN$_i$ WANT [PRO$_i$ SELL CAR]]

This example shows that there are two clauses as indicated by brackets. The matrix clause surrounds the entire sentence by the outermost brackets, with the interior brackets surrounding the subordinate clause. The subscript on both fs-JOHN and PRO indicates that they are coreferential. fs-JOHN determines the interpretation for PRO.

Following the derivation for Sentence 944 (Figure 4.33), SELL CAR forms a predicate that has no available subject as input. The Subject-pro drop rule changes the category to an **s** that serves as the input for WANT. The new constituent predicate looks leftward for fs-JOHN as the subject argument. Through the derivational process, fs-JOHN is the only subject available for the matrix and clausal complement clauses.

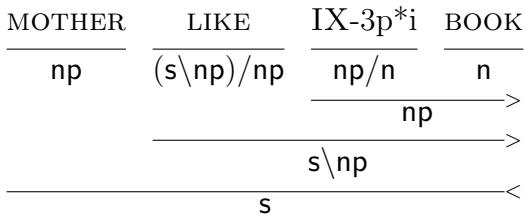
[5]Adger (2012) provides a discussion on the subject of PRO and control clauses in chapter 8.

10 Compositional Rules

We now illustrate instances where the parser executed rules of Combinatory Categorial Grammar during derivations. A more comprehensive discussion about the individual rules is found in Section 3.1.

10.1 Forward Application

Figure 4.34 illustrates a parsed sentence with two instances of forward application. The first is when the article IX-3p*i accepts its input argument BOOK to form a constituent noun phrase. The verb is then able to take as its argument the newly formed noun phrase also by forward application.



"Mother likes that book."

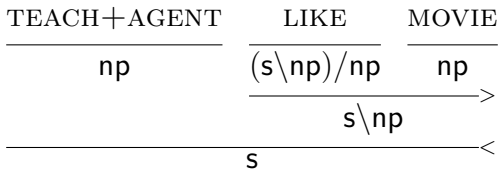
Figure 4.34: Sentence 691: Forward Application

10.2 Backward Application

Backward application allows for functions to consume their left-adjacent argument, such as when a predicate consumes its subject argument. The predicate in sentence 683 (Figure 4.35) consumes its subject argument.

10.3 Forward Composition

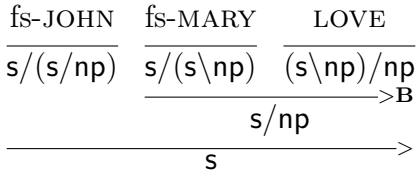
Forward composition permits two functions to compose to form a single composite function. Functional composition is permissible when two functions are adjacent to one another and where the head of one function is the same category type as the argument of the other



"The teacher likes movies."

Figure 4.35: Sentence 683: Backward Application

function. Sentence 1158 (Figure 4.36) has a type-raised subject with a functional category of **s/(s\np)** that composes with the verb's functional category of **(s\np)/np**. The argument of the type-raised subject is **s\np** which is the same as the head of the verb's function **s\np**. Since the head of one function is the same category as the argument of the other function, they may compose together to form a single constituent.



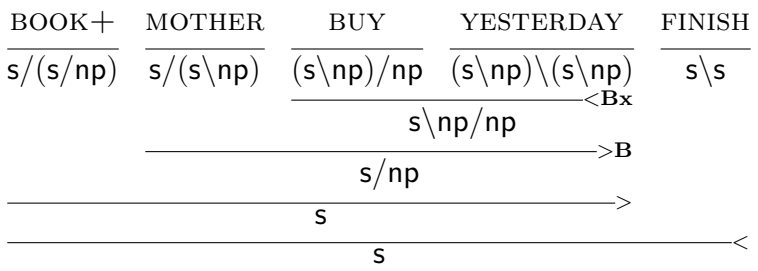
"As for John, Mary loves him."

Figure 4.36: Sentence 1158: Forward Composition

10.4 Backward Crossed Composition

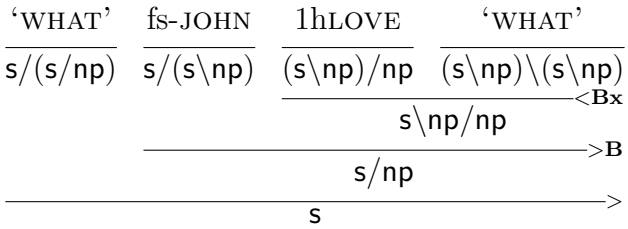
The following examples in Figures 4.37 and 4.38 are of constituents that compose by backward crossed composition. Backward crossed composition is a rule that allows constituents with different directional slash types to compose in order to form a single constituent.

Sentence 768 (Figure 4.37) and 999 (Figure 4.38) have different constituent types that utilize this rule. In sentence 768, the adverb composes with the verb; in sentence 999, the object that is part of a base-generated topic sentence composes with the verb. The composition can be seen in the underlines directly below BUY YESTERDAY and 1hLOVE 'WHAT' labeled with a $_{<\mathbf{Bx}}$. In each example, the composed constituents form a new constituent category that is a composite of the two original categories.



"As for the book, mother bought it yesterday; that's done."

Figure 4.37: Sentence 768: Backward Crossed Composition



"What is it John loves, what?"

Figure 4.38: Sentence 999: Backward Crossed Composition

Chapter 5

Evaluation

Various methods exist for evaluating a parser and/or grammar (Carroll et al., 1998). Current practice is to evaluate a parser's output by comparing it against an annotated gold standard. A gold standard for a corpus consists of correct annotations provided by human or automated means that provide "additional structured information" to data (Wissler et al., 2014, p. 1). Some annotated corpora that have been employed to evaluate parsers include the Penn Treebank, the British National Corpus, CCGbank and the Chinese CCGbank (Baldwin et al., 2004; de Marneffe et al., 2006; Hockenmaier and Steedman, 2002a,b; Tse and Curran, 2012). Different approaches can be taken when evaluating parser output; PARSEVAL is one of the most well-known. It evaluates output based upon bracketed notations (Black et al., 1991). Bracketed notation is equivalent to a parse tree whose words have been grouped by brackets. PARSEVAL evaluates precision and recall, but it is not the only method for evaluating a grammar.

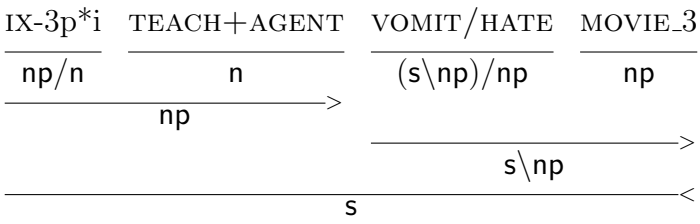
Another parser evaluation method is Leaf-Ancestor, which compares the paths from terminal nodes to root nodes of an output parse and compares those to a gold standard tree parse (Rehbein and van Genabith, 2007). A variety of dependency-based evaluations are also found in literature (Rimell et al., 2009). Semantic evaluation metrics also exist for parsers (Bisk et al., 2016; Kübler et al., 2011), though this thesis does not address semantic parsing.

Typically, computer output is evaluated by precision and recall measures. The evaluation undertaken in this thesis only examined precision. Precision is the proportion of correct guesses made by the computer to the total number of guesses that the computer

made. Recall is the proportion of correct guesses made by the computer to the total number of correct guesses. I did not evaluate recall because when developing a grammar for a low-resource language it is often not evaluated until further refinement.

1 Evaluation Method

As explained in Chapter 2, the derived parses for categorial grammars are akin to syntax trees, only inverted. The 'leaf' nodes, i.e. lexical items, are at the top of the tree with the root at the bottom (see Figure 5.1).



"The teacher doesn't like the movie.'

Figure 5.1: CCG parse for a sentence

The first line in the derived parse structure contains the lexical items. The levels below the lexical items are composed of the lexical-syntactic categories assigned to each lexical item and any derived constituent. The final level for all derived parses is a final constituent composed of all the lexical items in the string.

Since no gold standard exists for parsing the NCSLGR Corpus, and quantitative scoring for large-scale parsing of ASL has not been done before, this thesis introduces a method to quantify the parsing accuracy. The evaluation undertaken here provides a better measure than only reporting coverage, the "[c]alculation of the percentage of sentences from a given, unannotated corpus that are assigned one or more analyses by a parser/grammar..."(Carroll et al., 1998, p. 1). By itself, coverage is not a reliable measure for how a parser performs because it does not address whether any of the generated parses are correct.

Instead, I propose evaluating the parses based upon two metrics. The first metric looks at whether the assigned category for each lexical item and each composed constituent is correct. The second metric scores each constituent and whether it has been combined correctly. As discussed in Section 3.1, constituents may compose together via Combinatory Categorial Grammar rules in various ways. If given constituents are appropriately composed, then the result is considered correct.

The overall scoring algorithm is as follows:

1. Select a parsed sentence from the corpus.
2. Select the most correct parse for that sentence.
3. Determine the total number of derivations in the parse.
4. For each derivation, identify whether the correct category has been assigned.
5. For each derivation, determine whether its constituency combination is correct.

2 Evaluation Examples

Consider the scoring method for the following example sentences. Figure 5.2 shows the derivational parse for utterance 1221. The sentence contains a subject, a grammaticalized tense lexeme, and an intransitive verb. Each lexeme has the correct lexical-syntactic category assigned, and each compositional constituent derivations also has the correct category. Hence the output has a score of five out of five for the Syntactic Category Score.

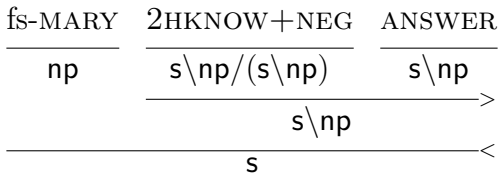
$$\dfrac{\dfrac{\text{fs-JOHN}}{\mathsf{np}} \quad \dfrac{\dfrac{\text{FUTURE}}{(\mathsf{s}\backslash\mathsf{np})/(\mathsf{s}\backslash\mathsf{np})} \quad \dfrac{\text{GO-OUT}}{\mathsf{s}\backslash\mathsf{np}}}{\mathsf{s}\backslash\mathsf{np}}\,{>}}{\mathsf{s}}\,{<}$$

"John will go."

Figure 5.2: Parse for sentence 1221

To obtain the total possible score for any parse, each constituent combination is also evaluated. The parse in Figure 5.2 has a total of five distinct line combinations. The composition of each is correct, so the total Constituency Combination score is five out of five.

Figure 5.3 is an example of a parse where two of the type categories are incorrect, but where the constituency combination is correct.



"Mary doesn't know the answer."

Figure 5.3: Parse for sentence 1188

In this sentence, fs-MARY has the correct category, whereas 2hKNOW+NEG and ANSWER do not. Within the grammar the lexeme 2hKNOW+NEG has available to it the transitive predicate category of **s\np/np**. Instead of applying that category in this context the parser applied the complex functional category of **(s\np)/(s\np)**. ANSWER was also assigned an incorrect lexical-syntactic category. Instead of **np** it was assigned **s\np**. Although the individual categories are incorrect, in this case the constituent content is correct when 2hKNOW+NEG and ANSWER do combine. The predicate category is able to compose with the subject **np** category so as to yield a final derivational category of **s**. For this parse the final score for the Syntactic Category Score is three and the Constituency Combination Score is five, and the total possible score is five for each. Table 5.1 reports sample scores from the first ten parsed sentences.

3 Final Results

The evaluation for the parser was done by hand. In total, 1,735 utterances were evaluated out of a possible total of 1,866 utterances; the reason is that for 131 sentences no parse was generated by the system. If a parse was not obtained for a sentence it was either due to a

Utterance	Syntactic Category Score	Constituency Combination Score	Possible
789	9	10	12
1197	6	6	7
1319	5	5	5
342	5	5	5
1222	9	9	9
1221	5	5	5
1550	22	22	25
914	7	7	7
1626	11	15	22
1285	14	12	16
Total score	93	96	113
	82%	85%	100%

Table 5.1: First 10 evaluated sentences

missing lexical item in the *morph.xml* file, or the available lexical-syntactic category types could not compose given the current rule base. Each successfully parsed sentence has at least one generated parse. The most correct parse for each sentence is the one that was included in the final evaluation. The process for selecting which parse to evaluate depended upon the parse having the fewest number of constituent compositions along with having the most correct number of lexical-syntactic categories present. The evaluation process was followed for all 1,735 utterances, and took approximately 40 hours. This process was carried out by one person. The evaluation results for the entire corpus are listed in Table 5.2.

Syntactic Category Score	Constituency Combination Score	Possible
12,862	17,901	22,124
58.14%	80.91%	100%

Table 5.2: Final evaluation results

The percentages in Figure 5.2 reflect the precision score based upon a modified precision evaluation method. The modification comes from only evaluating the parse that was the most correct for any given output sentence. The average number of generated parses per sentence is 15.65. Only one parse for each sentence was selected out of an average of 15.65 parses per sentence.

The scores presented[1] stand as a benchmark since there is no gold standard of parsed ASL data to compare to, especially one for Combinatory Categorial Grammar. The Constituency Combination Score is greater than the Syntactic Category Score probably due to CCG's compositional flexibility. It is built to be able to derive a parse. Finally, the total number of sentences which obtained a parse is 1,735 out of 1,866, yielding a coverage for this grammar of 92.9%.

[1]Pertinent files are downloadable at: `http://linguistics.byu.edu/thesisdata/ASL-CCG.zip`.

Chapter 6

Discussion and Conclusion

The two hypotheses I explored in this thesis are: 1. Combinatory Categorial Grammar is well suited for use in parsing a sizable corpus of American Sign Language gloss, which has apparently not been done yet; 2. The theoretical framework of Categorial Grammar can provide analyses for a wide range of interesting constructions in ASL, whereas only a few constructions have been addressed to date. In order to prove these hypotheses, I constructed a grammar for ASL using OpenCCG implementing Combinatory Categorial Grammar. The National Center for Sign Language and Gesture Resources Corpus was parsed using CCG. Several ASL sentence constructions were analyzed and presented as evidence that Categorial Grammar can model interesting ASL constructions.

Several limitations and opportunities for future work need to be mentioned in connection with this work. The first is with the corpus itself. Some glosses in the corpus were assigned incorrect part-of-speech tags. For example, occasionally classifiers were tagged incorrectly, which required preprocessing before the parse (see Appendix A). They were labeled incorrectly as either verbs or nouns.

Second, the multi-modal extensions (Baldridge, 2002; Baldridge and Kruijff, 2003) available in OpenCCG were not employed which is beyond the scope of this thesis. Had these extensions been used it is possible that the accuracy of the generated parses would be higher, because they allow for hierarchically ordered constraints on rule applications.

Third, non-manual markers can possibly be added as features. The scope of the NNMs can indicate constituency among the lexical items, thereby potentially increasing the Constituency Combination Score.

Fourth, semantics was not built into this grammar. Categorial Grammar and OpenCCG are ideally suited for integrating syntactic and semantic parsing. If semantics is added to the grammar it should be able to increase system accuracy because tandem parsing would constrain parser performance. This should decrease spurious derivations from occurring.

A fifth point to mention is that recall was not used as a metric for evaluation. As is often the case with new parser/grammar implementations for emerging languages without a gold standard, scoring the parses for recall would have been too involved for this thesis.

Finally, the evaluation process needs to be addressed. It was done by hand by one subject-matter expert. Having an individual evaluate all the results does not invalidate those results, it just means that human error is possible given the limitations of fatigue, intuition, and time constraints which all can affect how correctly an evaluation was carried out. It would have been better to have had several people work on evaluating the generated parses. That way results from many could be compared and a single evaluation could then be agreed upon.

However, the limitations listed do not compromise the importance of this work. The constructed grammar implemented in OpenCCG was able to parse the NCSLGR Corpus. The work carried out also demonstrates the importance of having corpus data of ASL available to researchers. If the NCSLGR Corpus did not exist this thesis would not have happened.

This thesis will hopefully serve as a spring-board for other researchers to study ASL. Work can be started almost immediately to improve the generated results. Implementing the multi-modal extensions can be a project easily undertaken because the bulk of the work has already been done in constructing an ASL grammar. Adding semantic features would help in the constituency composition because it could restrict spurious combinations. Lastly, there is

an updated corpus of American Sign Language from the American Sign Language Linguistic Research Project. The constructed grammar from this thesis could be applied to that corpus. The work carried out in this thesis should be a starting point not a terminal destination.

References

Aarons, D. (1994). *Aspects of the Syntax of American Sign Language*. PhD thesis, Boston University, Boston, Massachusetts.

Abney, S. (1991). Parsing by Chunks. In Berwick, R. C., Abney, S. P., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, chapter 10, pages 257–278. Kluwer Academic Publishers.

Ades, A. E. and Steedman, M. (1982). On the Order of Words. *Linguistics and Philosophy*, 4(4):517–558.

Adger, D. (2012). *Core Syntax*. Oxford University Press.

Baldridge, J. (2002). *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, Edinburgh, Scotland.

Baldridge, J. and Kruijff, G.-J. M. (2003). Multi-Modal Combinatory Categorial Grammar. In Copestake, A. and Hajič, J., editors, *10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 211–218, Budapest, Hungary. Association for Computational Linguistics.

Baldwin, T., Bender, E. M., Flickinger, D., Kim, A., and Oepen, S. (2004). Road-testing the English Resource Grammar over the British National Corpus. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., and Silva, R., editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 2047–2050, Lisbon, Portugal. European Language Resources Association (ELRA).

Bar-Hillel, Y. (1953). A Quasi-Arithmetical Notation for Syntactic Description. *Language*, 29(1):47–58.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Bisk, Y., Reddy, S., Blitzer, J., Hockenmaier, J., and Steedman, M. (2016). Evaluating Induced CCG Parsers on Grounded Semantic Parsing. In Su, J., Duh, K., and Carreras,

X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2022–2027, Austin, Texas. Association for Computational Linguistics.

Black, E. W., Abney, S., Flickenger, S., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R. J., Jelinek, F., Klavans, J. L., Liberman, M., Marcus, M. P., Roukos, S., Santorini, B., and Strzalkowski, T. (1991). A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In *HLT'91: Proceedings of the Workshop on Speech and Natural Langauge*, pages 306–311, Pacific Grove, California. Association for Computational Linguistics.

Bono, M., Kikuchi, K., Cibulka, P., and Osugi, Y. (2014). A Colloquial Corpus of Japanese Sign Language: Linguistic Resources for Observing Sign Language Conversations. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1898–1904, Reykjavik, Iceland. Language Resources and Evaluation Conference, European Language Resources Association (ELRA).

Bozşahin, C., Geert-Jan M. Kruijff, and White, M. (2013). *Specifying Grammars for OpenCCG: A Rough Guide*.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT2002)*, volume 168, Sozopol, Bulgaria.

Braze, D. (2004). Aspectual Inflection, Verb Raising and Object Fronting in American Sign Language. *Lingua*, 114:29–58.

Bungeroth, J., Stein, D., Dreuw, P., Ney, H., Morrissey, S., Way, A., and van Zijl, L. (2008). The ATIS Sign Language Corpus. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 2943–2946, Marrakech, Morocco. Language Resources and Evaluation Conference, European Language Resources Association (ELRA).

Bungeroth, J., Stein, D., Dreuw, P., Zahedi, M., and Ney, H. (2006). A German Sign Language Corpus of the Domain Weather Report. In Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J., and Tapias, D., editors, *Proceedings of the Fifth*

International Conference on Language Resources and Evaluation (LREC'06), pages 2000–2003, Genoa, Italy. Language Resources and Evaluation Conference, European Language Resources Association (ELRA).

Carpenter, B. and Penn, G. (2001). *The Attribute Logic Engine User's Guide Version 3.2.1*. Bell Labs.

Carroll, J., Briscoe, T., and Sanfilippo, A. (1998). Parser Evaluation: a Survey and a New Proposal. In *Proceedings of the First International Conference on Language Resources and Evaluation*, Granada, Spain. European Language Resources Association (ELRA).

Chomsky, N. (1956). Three Models for the Description of Language. *IRE Transactions on Information Theory*, 2(3):113–124.

Chomsky, N. (1970). Remarks on Nominalization. In Jacobs, R. A. and Rosenbaum, P. S., editors, *Readings in English Transformational Grammar*, chapter 12, pages 184–221. Ginn and Company.

Chung, J.-W. and Park, J. C. (2011). Text Parsing for Sign Language Generation with Combinatory Categorial Grammar. In *Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2011)*, University of Dundee, UK.

Curry, H. B. and Feys, R. (1958). *Combinatory Logic*, volume 1. North-Holland Publishing Company.

Dasgupta, T., Dandpat, S., and Basu, A. (2008). Prototype Machine Translation System from Text-toIndian Sign Language. In Singh, A. K., editor, *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 19–26, Hyderabad, India. Asian Federation of Natural Language Processing.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J., and Tapias, D., editors, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 449–454, Genoa, Italy. European Language Resources Association (ELRA).

Fischer, S. D. (1996). The Role of Agreement and Auxiliaries in Sign Language. *Lingua*, 98:103–119.

Forster, J., Schmidt, C., Hoyoux, T., Koller, O., Zelle, U., Piater, J., and Ney, H. (2012). Rwth-phoenix-weather: A Large Vocabulary Sign Language Recognition and Translation

Corpus. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Morena, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation(LREC'12)*, pages 3785–3789, Istanbul, Turkey. European Language Resources Association (ELRA).

Geraci, C., Mazzei, A., and Angster, M. (2014). Some Issues on Italian to LIS Automatic Translation. The Case of Train Announcements. In Basili, R., Lenci, A., and Magnini, B., editors, *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 the Fourth International Workshop EVALITA 2014*, pages 191–196, Pisa, Italy. Pisa University Press.

Herrmann, A. and Steinbach, M. (2011). Nonmanuals in Sign Languages. *Sign Language & Linguistics*, 14(1):3–8.

Hockenmaier, J. and Steedman, M. (2002a). Acquiring Compact Lexicalized Grammars from a Cleaner Treebank. In Rodríguez, M. G. and Araujo, C. P. S., editors, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, pages 1974–1981, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Hockenmaier, J. and Steedman, M. (2002b). Generative Models for Statisitcal Parsing with Combinatory Categorial Grammar. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 335–342, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jackendoff, R. (1977). *X Syntax:A Study of Phrase Structure*. MIT Press.

Janzen, T., O'Dea, B., and Shaffer, B. (2001). The Construal of Events: Passives in American Sign Language. *Sign Language Studies*, 1(3):281–310.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing*. Pearson.

Kawahara, D., Kurohashi, S., and Hasida, K. (2002). Construction of a Japanese Relevance-tagged Corpus. In Rodríguez, M. G. and Araujo, C. P. S., editors, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, pages 2008–2013, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Kübler, S., Cantrell, R., and Scheutz, M. (2011). Actions Speak Louder than Words: Evaluating Parsers in the Context of Natural Language Understanding Systems for Human-Robot Interaction. In Mitkov, R. and Angelova, G., editors, *Proceedings of the International*

Conference Recent Advances in Natural Language Processing 2011, pages 56–62, Hissar, Bulgaria. Association for Computational Linguistics.

Kuhn, J. (2016). ASL Loci: Variables or Features? *Journal of Semantics*, 33:449–491.

Lambek, J. (1958). The Mathematics of Sentence Structure. *The American Mathematical Monthly*, 65(3):154–170.

Li, X. and Roth, D. (2001). Exploring Evidence for Shallow Parsing. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (CoNLL)*, Toulouse, France.

Liddell, S. K. (1980). *American Sign Language Syntax*. Mouton Publishers.

Lillo-Martin, D. (1986). Two Kinds of Null Arguments in American Sign Language. *Natural Language & Linguistic Theory*, 4(4):415–444.

Marshall, I. and Safar, E. (2004). Sign Language Generation in an ALE HPSG. In Müller, S., editor, *Proceedings of the 11th International Conference on Head-Driven Phrase Structure Grammar*, pages 189–201, Flanders, Belgium. Center for Computational Linguistics, Katholieke Universiteit Leuven, CSLI Publications.

Mazzei, A. (2011). Building a Generator for Italian Sign Language. In Gardent, C. and Striegnitz, K., editors, *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 170–175, Nancy, France. Association for Computational Linguistics.

Mazzei, A. (2012). Sign Language Generation with Expert Systems and CCG. In Eugenio, B. D. and McRoy, S., editors, *Proceedings of the 7th International Natural Language Generation Conference*, pages 105–109. Association for Computational Linguistics.

Mesch, J. and Wallin, L. (2015). Gloss Annotations in the Swedish Sign Language Corpus. *International Journal of Corpus Linguistics*, 20(1):102–120.

Neidle, C., Bahan, B., MacLaughlin, D., Lee, R. G., and Kegl, J. (1998a). Realizations of Syntactic Agreement in American Sign Language: Similarities Between the Clause and the Noun Phrase. *Studia Linguistica*, 52:191–226.

Neidle, C., Kegl, J., MacLaughlin, D., Bahan, B., and Lee, R. G. (2000). *The Syntax of American Sign Language*. MIT Press.

Neidle, C., MacLaughlin, D., Bahan, B., Lee, R. G., and Kegl, J. (1997). The Sign Stream Project. Technical Report 5, Boston University.

Neidle, C., MacLaughlin, D., Lee, R. G., Bahan, B., and Kegl, J. (1998b). The Rightward Analysis of wh-Movement in ASL: A Reply to Petronio and Lillo-Martin. *Language*, 74(4):819–831.

Neidle, C., Thangali, A., and Sclaroff, S. (2012). Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus. In Crasborn, O., Efthimiou, E., Fotinea, E., Hanke, T., Kristoffersen, J., and Mesch, J., editors, *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*, pages 143–150, Istanbul, Turkey. Language Resources and Evaluation Conference (LREC).

Neidle, C. and Vogler, C. (2012). A New Web Interface to Facilitate Access to Corpora: Development of the ASLLRP Data Access Interface (DAI). In Crasborn, O., Efthimiou, E., Fotinea, E., Hanke, T., Kristoffersen, J., and Mesch, J., editors, *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*, pages 137–142, Istanbul, Turkey. Language Resources and Evaluation Conference (LREC).

Osborn, T. (2014). Dependency Grammar. In Carnie, A., Sato, Y., and Siddiqi, D., editors, *The Routledge Handbook of Syntax*, chapter 29, pages 604–626. Routledge.

Östling, R., Börstell, C., Gärdenfors, M., and Wirén, M. (2017). Universal Dependencies for Swedish Sign Language. In Tiedemann, J. and Tahmasebi, N., editors, *Proceedings of the 21st Nordic Conference of Computational Linguistics*, pages 303–308, Gothenburg, Sweden. Association for Computational Linguistics.

Petronio, K. and Lillo-Martin, D. (1997). WH-Movement and the Position of Spec-CP: Evidence from American Sign Language. *Language*, 73(1):18–57.

Rehbein, I. and van Genabith, J. (2007). Evaluating Evaluation Measures. In Nivre, J., Kaalep, H.-J., Muischnek, K., and Koit, M., editors, *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 372–379, Tartu, Estonia. University of Tartu, Estonia.

Rimell, L., Clark, S., and Steedman, M. (2009). Unbounded Dependency Recovery for Parser Evaluation. In Koehn, P. and Mihalcea, R., editors, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, pages 813–821, Singapore. Association for Computational Linguistics.

Segouat, J. and Braffort, A. (2009). Toward Categorization of Sign Language Corpora. In Fung, P., Zweigenbaum, P., and Rapp, R., editors, *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: From Parallel to Non-parallel Corpora (BUCC)*, pages 65–67. Association for Computational Linguistics (ACL), Association for Computational Linguistics.

Sevinç, A. M. (2006). Grammatical Relations and Word Order in Turkish Sign Language (TİD).

Speers, d'A. L. (2001). *Representation of American Sign Language for Machine Translation*. PhD thesis, Georgetown University, Washington D.C.

Steedman, M. (1985). Dependency and Coördination in the Grammar of Dutch and English. *Language*, 61(3):523–568.

Steedman, M. (1993). Categorial Grammar. *Lingua*, 90:221–258.

Steedman, M. (2014). Categorial Grammar. In Carnie, A., Sato, Y., and Siddiqi, D., editors, *The Routledge Handbook of Syntax*, chapter 32, pages 670–701. Routledge.

Stokoe, W. C. (1960). Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Studies in Linguistics*, Occasional Papers(8):1–78.

Stokoe, W. C. (1970). The Study of Sign Language. *ERIC Clearinghouse for Linguistics*, pages 1–38.

Straka, M., Hajič, J., and Straková, J. (2016). Udpipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Tse, D. and Curran, J. R. (2012). The Challenges of Parsing Chinese with Combinatory Categorial Grammar. In Fosler-Lussier, E., Riloff, E., and Bangalore, S., editors, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 295–304, Montréal, Canada. Association for Computational Linguistics.

Vanderwende, L., Menezes, A., and Quirk, C. (2015). An AMR Parser for English, Frence, German, Spanish and Japanese and a New AMR-Annotated Corpus. In Gerber, M., Havasi,

C., and Lacatusu, F., editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 26–30, Denver, Colorado. Association for Computational Linguistics.

Wilbur, R. (2017). Internally-headed Relative Clauses in Sign Languages. *Glossa*, 2(1):1–34.

Wissler, L., Almashraee, M., Monett, D., and Paschke, A. (2014). The Gold Standard in Corpus Annotation. In *5th Institute of Electrical and Electronics Engineers (IEEE) Germany Student Conference*, Passau, Germany. Institute of Electrical and Electronics Engineers.

Wood, M. M. (1993). *Categorial Grammars*. Routledge.

Wright, T. (2008). A Combinatory Categorial Grammar of a Fragment of American Sign Language. In Gaylord, N., Hilderbrand, S., Lyu, H., Palmer, A., and Ponvert, E., editors, *Texas Linguistic Society 10: Computational Linguistics for Less-Studied Languages*, pages 138–149, Austin, Texas. TLSX Texas Linguistic Society, CSLI Publications.

Appendix A

Preprocessing to the NCSLGR XML

Filename and line number	Modification
lapd.xml–line 3358	added: <A E="1100" S="834" VID="8"/>
ncslgr10g.xml–line 480	added: <A E="1633" S="1433" VID="8"/>
ncslgr10l.xml–line 5632	added: <A E="767" S="600" VID="12"/>
speeding.xml–line 3565	added: <A E="2066" S="1666" VID="10"/>
three pigs.xml–line 2718	added: <A E="2333" S="2166" VID="8"/>
three pigs.xml–line 2719	added: <A E="2533" S="2366" VID="8"/>
three pigs.xml–line 6410	added: <A E="67" S="0" VID="0"/>
three pigs.xml–line 1821	changed: The VID value from 8 to 14; corrected from Verb to Classifier.
three pigs.xml–line 6492	changed: The VID value from 0 to 14; corrected from Noun to Classifier.
ali.xml–line 1351	added: <A E="5267" S="5233" VID="9"/>
boston-la.xml–line 2039	added: <A E="2700" S="2400" VID="16"/>
dorm prank.xml–line 5932	changed: The VID value from 0 to 15; corrected from Noun to Classifier.
DSP Dead Dog Story–line 809	added: <A E="1000" S="0" VID="2"/>
ncslgr10n.xml–line 1730	changed: The VID value from 5 to 16; corrected from Wh-word to Particle.
scarystory.xml–line 2058	added: The TRACK tag did not have a closing forward slash, so I added one.

Appendix B

21 NCSLGR utterances not included

The utterances listed here were not included in the final data set used in this thesis. These were dismissed due to not being able to incorporate non-dominant hand gloss with the dominant hand gloss.

1. LOOK:i BCL'holding_and_examining_hand' 1h5'wave_no'+ IX-1p SUMMON FOR SECOND OPINION **NON-DOM** FINGER SUMMON
2. FACE+SAME IX-loc:i REALLY+WORK SCL:B'finger_being_cut_by_machine' IX-1p 5'that's the way it is' **NON-DOM** FINGER FINGER
3. #SO START fs-TO BPCL:V'stand_up'-inceptive REALLY SHAKE ANSWER+AGENT 5'you_know' LITTLE-BIT AFRAID NOT KNOW HOW TO/UNTIL CONFRONT THAT SITUATION **NON-DOM** LCL:B-L'ground' CONFRONT
4. FINE OTHER THAN EAT IX-1p CAN COMPARE POSS-3p:i ENVIRONMENT **NON-DOM** POSS-3p:j
5. IX-1p LOOK:j DCL:5'headlights_on' FINISH **NON-DOM** DRIVE
6. IX-1p 1hTEND SCL:1'sneaking' CAUSE PROBLEM FOR OTHER LIVE **NON-DOM** MAKE+AGENT DCL:5'roof'
7. ICL:S'holding_carrier_from_the_top' WOMAN LOOK:t NOT POSS-1p DOG IX-3p:t **NON-DOM** ICL:B-L'holding_carrier_from_the_bottom' fs-OH
8. 5MAN ICL:S'holding_carrier_from_the_top' **NON-DOM** fs-OH ICL:B-L'holding_carrier_from_the_bo
9. ICL:S'holding_carrier_from_the_top' **NON-DOM** i:1hCOME:h ICL:B-L'holding_carrier_from_the_bo
10. ICL:S'hold_carrier_from_top' DOG 1hDIE **NON-DOM** ICL:B-L'hold_carrier_from_bottom'
11. ICL:S'holding_carrier_from_top' COLLAR DCL'tags_swinging' ICL:S'holding_carrier_from_top' 1hHAVE ICL:S'holding_carrier_from_top' **NON-DOM** ICL:B-L'holding_carrier_from_bottom' 1hHAVE IX-3p:t ICL:B-L'holding_carrier_from_bottom'
12. ICL:S'holding_carrier_from_top' 5WOMAN KNOW+ IX-3p:t NOT POSS-1p DOG part:indef **NON-DOM** ICL:B-L'holding_carrier_from_bottom'

13. 2hPCL:5wg'people_rushing_left_to_right':m ICL:S'pick_up_carrier_from_top' **NON-DOM** ICL:B-L'pick_up_carrier_from_bottom'
14. SPECIAL/EXCEPT+fs-LY WHEN IX-3p:m FOOTBALL REALLY ONLY LAW COMPETITION IX-2p CAN BPCL:bent-B'body_hitting' SOMETHING/ONE #SO HARD AND NOT GET PRICE FOR fs-IT **NON-DOM** SCL:1'person'
15. IX-1p REALLY SAD COW CANNOT MOVE REALLY SEE 5'wow' 1hWOW/AWFUL SCL:3'vehicle_moving' REALLY END fs-OF THAT PRISON SCL:5'enclosed_fence' IX-loc:t 1hHAVE fs-BUTCHER-SHOP REALLY ICL'cutting_head_off' SHOW DIE COW BPCL:2'cow_hanging' **NON-DOM** LOOK:t MOVE DRIVE SCL:5'enclosed_fence' ICL:C'holding_neck' IX-1p LOOK:t
16. REALLY POSS-1p IX-1p ALONE 1hHAVE FRIEND DRIVE IX-1p DRIVE++++++ **NON-DOM** DRIVE REALLY TWO FRONT
17. ICL'heart_beating_in_the_hand' BCL'walking_while_carrying_person' **NON-DOM** ICL'carrying_victim'
18. 2hSCL:3'vehicle_following_car' SIREN SCL:3'vehicle_following_car' HEARING BPCL:F'eyes_moving' IX-loc:l **NON-DOM** SCL:3'vehicle_driving' SCL:3'vehicle_driving'
19. REALLY ENOUGH IX-3p:i COP SCL:3'vehicle_pulls_behind_vehicle' **NON-DOM** ENOUGH SCL:3'vehicle_driving'
20. MISTAKE REALLY+WORK ENOUGH COP+ SCL:3'vehicle_pulls_behind_other_car' **NON-DOM** DRIVE IX-3p:i SCL:3'vehicle_driving'
21. COP OH-I-SEE POSS-1p MOTHER+FATHER DEAF **NON-DOM** IX-3p:m