Two Ojibwe Constraint Grammars: Morphological Disambiguation and Dependency Parsing

Matthias Diederichsen, Christopher Hammerly
University of British Columbia
{matthias.diederichsen, chris.hammerly}@ubc.ca

Abstract

This paper presents the first iteration of two connected Ojibwe constraint grammars, one for morphological disambiguation and one for syntactic parsing. Due to the polysynthetic nature of Ojibwe, along with its status as a low-resource language, the disambiguation grammar proves to be an effective and resource-efficient tool for morphological disambiguation, successfully eliminating 32% of redundant readings and fully resolving 41% of ambiguous tokens. The dependency grammar focuses on assigning dependency relations to model argument structure, where the constraint grammar once again proves to be an effective paradigm, with F1 scores of 0.97 for subject and 0.94 for object relations. The rule-based design of both grammars is linguistically informed, allowing for precise modeling of language-specific phenomena such as animacy, obviation, and verb-argument agreement. Applications of the two constraint grammars include building a disambiguated morphologically-tagged corpus of the Ojibwe language and creating a treebank for the Ojibwe language following the widely adopted CoNLL-U format.

Keywords: Ojibwe, Constraint Grammar, disambiguation, dependency parsing, low-resource language

1. Introduction

The Ojibwe language, also known as Anishinaabemowin, is a Central Algonquian language spoken across the Great Lakes region of Canada and the United States. With an estimated 30,000 speakers (Statistics Canada, 2023), Ojibwe remains one of the most widely-spoken Indigenous languages in North America, though it is considered endangered in many communities (Hermes and King, 2013). Ojibwe has been the focus of revitalization initiatives, where digital resources play a growing role in documentation, pedagogy, and community use. Ojibwe is characterized by rich inflectional morphology, myriad regional dialects, and variation across writing systems. These characteristics present both challenges and opportunities for the tasks of morphological disambiguation and syntactic parsing. This paper aims to introduce two constraint grammars (CGs) as valuable additions to this growing collection of digital resources: (i) a *disambiguation CG*, which focuses on resolving word-form ambiguities in properties such as part-of-speech (POS) and number, and (ii) a *dependency CG*, which focuses on modeling sentence-level dependencies, in particular argument structure relations.

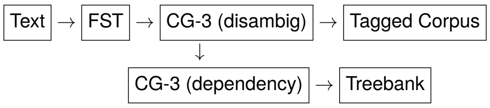
Our CGs are situated in a wider project, which aims to create linguistic resources for Ojibwe, including speech synthesis (Hammerly et al., 2023a; Wang et al., 2025; Chan and Hammerly, 2025), machine translation (Nguyen et al., 2025), and, as is the focus of this paper, tools for morphological and syntactic parsing. We also follow other recent work creating CG systems for under-resourced languages, including the related Central Algonquian language Plains Cree (Schmirler et al., 2018).

Our ultimate goal is to create a text-based tagged corpus and treebank for Ojibwe. This paper sets the stage for the creation of these resources. To process texts, we make use of the *OjibweMorph* finite-state transducer (FST) for morphological parsing and POS tagging (Hammerly et al., 2026). This FST provides robust word-level analyses, but has two shortcomings that we aim to resolve.

First, without knowledge of the wider sentence context, a number of ambiguities are produced by the FST parser. Consider the classic English example *duck*. In “the duck”, *duck* is a noun, while in “I duck” it is a verb. To resolve such ambiguities in our Ojibwe corpus, we use the CG framework to write constraints that can disambiguate FST readings based on the context of surrounding words.

Second, the FST is not well-suited for modeling sentence-level dependencies. Fortunately, the CG framework can also be extended to perform syntactic parsing in addition to disambiguation. By defining constraints that assign grammatical relations such as subject and object, it becomes possible to map the structure of a sentence. We apply this approach in a second CG module that builds on the disambiguated FST output to identify syntactic dependencies in Ojibwe, in particular those related to argument structure and core syntactic relations. These outputs are then converted into the CoNLL-U format from the Universal Dependencies (UD) framework (Nivre et al., 2020). The graphic in (1) illustrates the full pipeline.

(1) **Full pipeline**



Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 9756–9766
11–16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

2. Ojibwe Morphosyntax

This section provides an overview of Ojibwe morphosyntax with a focus on those attributes relevant for disambiguation and dependency parsing. We present examples in a format matching the whole-word analysis produced by the *OjibweMorph* FST rather than a traditional Leipzig-style interlinear gloss.[1] This provides a more straightforward foundation for the exposition of our CG models in the coming sections.

2.1. Animacy

Ojibwe distinguishes two noun classes on the basis of animacy. The animate class includes people and animals, and the inanimate class includes everything else, but the distinction is not always clear-cut (Valentine, 2001, p. 177). Important for our project is that certain words have both an animate and an inanimate variant. For example, the word *mitig* means 'tree' when animate (`NA`), but 'stick' or 'wood' when inanimate (`NI`). This ambiguity is captured by the Ojibwe FST, as shown in (2).

(2) **Animacy ambiguity with *mitig***
`mitig+NA+ProxSg`
`mitig+NI+Sg`

This is a prime example of an ambiguity that we want to resolve in context using the disambiguation CG, using other words such as adjacent demonstratives and verbs that unambiguously indicate an associated noun's animacy.

2.2. Obviation

Ojibwe also makes use of a system called *obviation*, which distinguishes a single animate third-person entity (proximate) from all other animate third-person entities (obviative). This proximate entity is semantically prominent, functioning as the "main character" in a discourse with multiple third persons (Valentine, 2001, p. 183).

In most dialects, the obviative form of nouns is ambiguous in number. The example in (3) shows the FST parse of *ikwewan*, meaning 'woman/women' in the obviative form.

(3) **Number ambiguity with *ikwewan***
`ikwe+NA+ObvSg`
`ikwe+NA+ObvPl`

As we see, both `ObvSg` and `ObvPl` readings are parsed. Moreover, obviative demonstratives and verbal agreement markers also show the same ambiguity, meaning this cannot be disambiguated through sentential context alone. We discuss some implications of this for our CGs in section 6.2.

2.3. Argument realization

2.3.1. Null arguments

Ojibwe is a radical *pro*-drop language, meaning verbal arguments are frequently dropped and not realized as separate words (Grafstein 1984, Hammerly et al. 2022). This means that a single inflected verb can constitute a grammatical sentence. Verbal arguments are interpreted as pronominal when there are no overt nominals to take argument roles. This presents a challenge for our disambiguation CG, as it is not always possible to disambiguate certain verbs due to the lack of surrounding independent argument information.

2.3.2. Nominal phrases

A number of elements, including pronouns, demonstratives, and nouns in different combinations, can appear as overt arguments to verbs. Pronouns tend to be only used for extra emphasis. Demonstratives alone are frequently realized as direct arguments to verbs, or can be paired with a noun to form a nominal phrase. For the purposes of our Ojibwe CGs, we loosely follow the "templatic order" of the Ojibwe nominal phrase as shown in (4), adapted from Sullivan (2016, p. 49).

(4) **Nominal Phrase Template**
(Dem)–(Num)–(N)–(RC)

The elements of the nominal phrase agree in number and obviation with the head noun. An example of how this template is realized is the common *(Dem)-(N)* combination, where a preceding demonstrative attaches to an agreeing noun to form a larger nominal phrase. For example, the phrase *awedi mitig* meaning 'that tree over there' has an animate singular demonstrative *awedi* that pairs with a noun *mitig*. An understanding of the nominal structure allows for a more precise definition of disambiguation and dependency rules, as discussed in section 5.2.

2.4. Argument structure

2.4.1. Transitivity and animacy-based paradigms

Ojibwe verbs are grouped into four main paradigms, which are split in terms of transitivity, and the animacy of subjects (for intransitives) or animacy of objects (for transitives). These paradigms are `VAI` (animate intransitive), `VII` (inanimate intransitive), `VTA` (transitive animate), and `VTI` (transitive inanimate). Importantly, some verb forms are ambiguous. For example, the intransitive verb *bapangigaa* 'it is dripping', is ambiguous between `VAI` and `VII` parses:

---

[1] The full FST tagset is documented at `https://github.com/ELF-Lab/OjibweMorph`.

(5) **Verb class ambiguity with *bapangigaa***

```
bapangigaa+VAI+Ind+Pos+Neu+3SgProxSubj
bapangigaa+VII+Ind+Pos+Neu+0SgSubj
```

Disambiguating such verb paradigm ambiguities is another aim of the present work.

That said, in the general case, inflected verbs unambiguously encode the person, number, animacy, and obviation of their arguments. For our task of mapping argument structure dependencies, this information allows us to determine the features of the subject and/or object of a given verb, and search the surrounding sentential context for any agreeing overt realizations.

2.4.2. Oblique arguments

Obliques are another common type of argument. Unlike core arguments, oblique arguments are not marked on the verb, which means that they must be identified purely through structure rather than verbal agreement. Locatives are one especially common type of oblique argument, which are sometimes headed by a preposition-like locative element, as discussed in section 5.4. We aim to model these arguments within our dependency CG.

2.4.3. Ditransitives

Ditransitive verbs take an additional argument beyond the typical agent and patient. In Ojibwe, only the indirect object (recipient) is marked on the verb. The direct object (theme) is instead realized as an additional, typically inanimate argument.

Ojibwe includes both inherent and derived ditransitives. Inherent ditransitives such as *miizh* ‘give (it) to someone’ are interpreted directly as ditransitives, while derived ditransitives are formed through the benefactive suffix *-amaw*, which adds a recipient or beneficiary role to the verb (Valentine, 2001). Correctly identifying ditransitive verbs and their argument structure is important for improving coverage of the dependency parsing CG.

3. Key Resources

This section provides a brief overview of the key resources used for the creation of the Ojibwe CGs.

3.1. The Ojibwe People’s Dictionary

The Ojibwe People’s Dictionary (OPD) is a large-scale online lexical resource for the Ojibwe language, developed through collaboration between linguists, educators, and Ojibwe speakers. It contains thousands of headwords, example sentences, audio recordings by first-language speakers, and grammatical information (Nichols, 2012).

Important for the current paper is the fact that the OPD includes over 8,000 unique example sentences from 8 speakers, which we used for both development and evaluation of the CGs. Indeed, while we consulted OPD examples during the development of the two CGs presented here, our rules are primarily motivated by more general grammatical documentation, including (Valentine, 2001) and the authors’ personal knowledge from over a decade of language learning and fieldwork. Furthermore, we kept our set of close to 500 test sentences separate, to avoid a problem of “overfitting” the model to the sentences. For details on the use of sentences in evaluation, see section 6.1.

3.2. The OjibweMorph FST

The *OjibweMorph* FST provides a broad-coverage morphological analyzer and generator for Ojibwe (Hammerly et al., 2026). It encodes inflectional, derivational, and lexical information, achieving high coverage and accuracy on a battery of different tests. Lexical information is primarily drawn from a database of entries from the OPD, but also contains additional items from other sources. The FST outputs rich morphological analyses related to part-of-speech, but also higher-order grammatical information. For example, on verbs, the FST can diagnose the person, number, and animacy of the subject and/or object implied by the inflection. This makes it directly useful for downstream disambiguation and syntactic parsing tasks. FST outputs have already been shown in section 2, and we will see them in action again in sections 4 and 5.

3.3. The Constraint Grammar framework

CG, in its original formulation by Karlsson (1990), is a classic computational paradigm for morphosyntactic analysis and disambiguation. CG focuses on building a framework suitable for parsing running text, which utilizes constraints that select, add, and remove readings iteratively over a text. We use the updated third implementation of the CG formalism (abbreviated CG-3), as introduced in Bick and Didriksen (2015).

In the CG formalism, rules for morphological disambiguation and dependency parsing are generally written separately. While these two tasks may have overlapping functionality (e.g. a dependent noun being used to disambiguate a verb), the ultimate uses of morphological disambiguation and dependency parsing are separate enough for us to keep these grammars apart (see the graphic in (1)).

While a detailed comparison with alternative methods is beyond the scope of the current paper, it is still important to mention the choice of CG relative to other approaches. In choosing a framework for morphological disambiguation and dependency parsing for Ojibwe, three desiderata were particularly important: (i) support for incremental development, (ii) suitability for low-resource settings, and (iii) active maintenance and development of the framework.

CG is a rule-based framework, allowing for linguistically-driven implementation, which makes both incremental development possible and allows for the creation of an accurate parser without the need for large amounts of data, satisfying desiderata (i) and (ii). Furthermore, CG is actively maintained, thus making it a good candidate for our current goals. In fact, a growing body of work applies the CG paradigm to under-resourced languages, including Plains Cree (Schmirler et al., 2018) and North Sámi (Wiechetek et al., 2019).

3.4. The Universal Dependencies framework

Universal Dependencies (UD) provides a cross-linguistic framework for representing syntactic annotation in a consistent way across languages. It is currently the most widely used framework for treebanks, supporting hundreds of languages and enabling comparative work across typologically diverse systems (Nivre et al., 2016).

For the current project, dependency parsing outputs are aligned with the UD framework, which contributes a standardized set of dependency relations and part-of-speech tags, and a data format (CoNLL-U) that supports long-term reusability. Moreover, aligning our project with the UD framework at this stage provides a path toward contributing a UD-compatible treebank for Ojibwe in the future. In general, we hope to contribute to a growing literature that applies polysynthetic and low-resource languages to the UD framework (Wagner et al., 2016; Uddin and Zeman, 2025).

4. The disambiguation grammar

The disambiguation CG is organized into four components: (i) set and variable definitions, (ii) initial context-free blanket constraints, (iii) context-dependent constraints, and (iv) final context-free blanket constraints. Each plays a role in systematically removing unwanted analyses and selecting the correct readings.

In this section and the following, rule and sentence examples will be shown using the CG-3 format, so as to illustrate the parallels between the linguistic motivation and CG functionality. The basic format is `"<word>"` followed by indented readings headed by the `"lemma"` tag. Removed readings are marked with a semicolon.

4.1. Set and variable definitions

We begin by defining sets of arguments according to number, animacy, and obviation. These sets provide shorthand references that simplify the later constraints. For example, all third-person animate subjects can be grouped together under a single set `AN_SUBJECT`, which allows rules to refer to the set instead of repeatedly listing individual tags. Lexical lists for special verb classes (such as ditransitives and impersonal verbs) and for derivational morphemes are also defined in this section.

4.2. Initial (context-free) blanket constraints

The next step introduces blanket constraints that apply independently of context. These rules target word-internal ambiguities that are not semantically meaningful and should be filtered out before later constraints are applied. For instance, impersonal verbs often carry spurious plural readings. Consider the verb *waaban* ‘it is tomorrow’, which surfaces in forms such as *waabang* with both `0SgSubj` and `0PlSubj` readings. Because impersonal verbs cannot semantically take a plural argument, the plural analysis is removed at this stage, preventing issues downstream.

4.3. Context-dependent constraints

4.3.1. Verbal disambiguation

Verbal ambiguity is resolved through agreement with nearby nominal phrases. For example, verbs that have a number ambiguity with respect to an inanimate object (`0SgObj` vs. `0PlObj`) are disambiguated by the rule shown in (6). Note, we have simplified this rule for expositional purposes.

(6) **Singular inanimate object rule**

```
SELECT 0SgObj IF ((-1* (NI Sg)
    BARRIER CB) OR (1* (NI Sg)
    BARRIER CB))
```

This is a select-type rule, which selects the singular inanimate object reading of a verb (`0SgObj`) when a nearby singular inanimate nominal is found. Specifically, the rule first searches left (`-1*`) then right (`1*`) for an agreeing inanimate singular noun (`NI Sg`), stopping the search at a clausal boundary in both cases (`BARRIER CB`). Similar rules for disambiguating verbs with animate arguments and subject ambiguities are also implemented.

Another frequent verbal ambiguity involves paradigm tags. As discussed in section 2.4.1, some verb forms are ambiguous between `VAI` and `VII` parses. Such an example is shown in (7), for the sentence *bapangigaa ’aw akik* ‘that pail leaks’.

(7) **Verb paradigm disambiguation**

```
"<bapangigaa>"
   "bapangigaa" VAI Ind Pos Neu
    3SgProxSubj
;  "bapangigaa" VII Ind Pos Neu
    0SgSubj
"<’aw>"
   "’aw" PRONDem NA ProxSg
"<akik>"
   "akik" NA ProxSg
```

Here, the sequence *’aw akik* ‘that pail’ forms an unambiguous animate nominal phrase, which gives context for the grammar to select the VAI reading (indicated by the VII reading being preceded by a semicolon).

In total, there are thirteen rules that handle context-dependent verbal class and verbal argument ambiguities. These handle the vast majority of ambiguous cases.

4.3.2. Nominal disambiguation

Nominal rules use verb agreement features and demonstratives to resolve number and animacy ambiguities. Take the simple phrase *awedi mitig* ‘that tree over there’. As discussed in section 2.3.2, the demonstrative and noun agree, meaning we can use an unambiguous demonstrative to disambiguate the noun as shown in (8).

(8) **Nominal animacy disambiguation**

```
"<awedi>"
   "awedi" PRONDem NA ProxSg
"<mitig>"
   "mitig" NA ProxSg
;  "mitig" NI Sg
```

Since the demonstrative *awedi* is unambiguously singular animate, we can safely disambiguate the noun to also be singular animate. There are twelve total rules used to disambiguate nouns based on context.

4.3.3. Other context-dependent ambiguities

The grammar also handles smaller classes of ambiguous forms. Ambiguous demonstratives are resolved for animacy when followed by an agreeing noun. In addition, rules that handle common idiosyncratic ambiguities are defined. One such example is the interrogative adverb *aaniin* ‘how’, which is ambiguous with the common greeting ‘hello’ (parsed as an interjectional particle). Here, a rule chooses the interrogative reading when a changed conjunct verb (the form used in questions) or adverb follows to the right.

4.4. Final (context-free) blanket constraints

The final set of blanket constraints applies once contextual rules have been run. These rules maximize lexicalization by preferring more specific lexical items over derived analyses. We define these rules at the end to avoid certain edge cases where the maximally lexicalized reading is not the correct choice. One instance when this could happen is a POS ambiguity (e.g. verb vs. noun), where we will want to let the context-dependent constraints first select the correct POS.

An example of a lexicalization rule is selecting the lexicalized diminutive over the derived diminutive reading. For example, *ishkodens* gets two analyses, one that treats the whole word as a lemma, and one that treats it as decomposed into the lemma *ishkode* ‘fire’ in the diminutive form. However, *ishkodens* has an idiosyncratic meaning. Rather than meaning ‘small fire’, it means ‘match’. So our principle is to select the maximal lemma, as shown in (9).

(9) **Lexicalized diminutive selection**

```
"<ishkodens>"
   "ishkodens" NI Sg
;  "ishkode" NI Dim Sg
```

4.5. Feeding the dependency grammar

Before turning to the dependency grammar, it is important to highlight how the two grammars are connected. The disambiguation grammar provides the foundation for the dependency grammar by ensuring that each token has a single, consistent analysis before syntactic rules are applied. One instance where this is crucial is POS disambiguation. If we cannot establish the correct POS, then it is impossible to determine a word’s syntactic function. For example, as discussed in section 4.3.3, the form *aaniin* can either be analyzed as an interrogative pronoun or as an interjectional particle. Without disambiguating the POS, the dependency grammar would not know whether to treat *aaniin* as a question word linking to a clause or a particle modifying another element. In a sense, the disambiguation grammar can be understood as the lexical “pre-processing” stage, while the dependency grammar is responsible for higher-level syntactic interpretation.

5. The dependency grammar

In the dependency grammar, our primary goal is the parsing of verbal arguments and argument-internal structure. Because we focus on arguments first, we only partially model adverbs, discourse particles, and cross-clausal dependencies at this stage.

We divide the grammar into: (i) set definitions and tag additions, (ii) nominal dependency parsing, (iii) verbal dependency parsing (verbal arguments), (iv) other dependencies, and (v) post-processing.

5.1. Set definitions and tag additions

In the dependency grammar, we reuse set definitions from the disambiguation grammar, and introduce mappings from FST POS to UD UPOS (Universal POS) so that labeling remains stable downstream. As with disambiguation, we also define a clause boundary (CB): since we do not yet handle cross-clausal attachments, we treat all verbs as clause boundaries except (a) verbs identified as relative-clause heads and (b) impersonal VII predicates that function as modifiers. This boundary forces attachments to stay local and prevents accidental long-distance dependencies.

5.2. Parsing nominal dependencies

As a first step, we build the nominal phrases. This "smaller constituent first" approach reduces spurious verb–argument links, where we typically want the verb linking to a head noun. In all subsequent constraints, each `SETPARENT` call, which creates a structural edge, is immediately followed by an `ADD` call, which assigns the UD relation, so that the graph always carries both head information and labels.

As discussed in section 2.3.2, demonstratives almost always precede the noun. Accordingly, the first pass attaches a demonstrative to the following noun and adds the `@det` relation tag. This handles cases like *awedi mitig* 'that tree over there', which is shown in (10).

(10) **Dem-Noun example parse**

```
"<awedi>"
   "awedi" PRONDem NA ProxSg @det
    DET #1->2
"<mitig>"
   "mitig" NA ProxSg #2->2
```

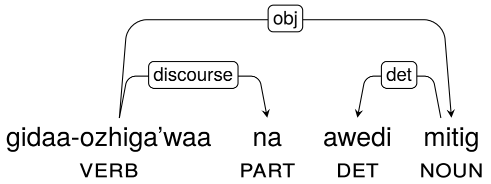
We treat numerals in the same way by attaching them to the noun with `nummod`. For relative clauses (RCs), we attach an `acl:relcl` to the noun when the RC verb shows at least one agreeing argument slot with that noun. At this point, we do not distinguish between subject and object RCs.

5.3. Parsing verbal dependencies

The next step is to link verbs to their arguments. The attachment logic can be understood as agreement-driven: unattached nominal elements seek out verbs with argument slots agreeing in person, number, obviation, and animacy features. Clause boundaries prevent the search from crossing into adjacent clauses. Because Ojibwe allows both preverbal and postverbal surface patterns, we implement left-side and right-side variants for each rule. When two candidate verbs compete, ties are resolved with a left-first preference unless a rule explicitly requires right-only attachment (as with some negatives).

Continuing the dependency example in (10), we now look at a full sentence *gidaa-ozhiga'waa na awedi mitig* 'can you tap that tree over there', where (11) shows how we parse the complete argument structure.

(11)



Here, the question particle *na* is parsed as discourse, and the verb is linked to the agreeing animate singular object *mitig*. The demonstrative contributes inside the nominal phrase as a determiner and does not feed clausal roles.

In other cases, pronouns (including demonstratives without an adjacent noun) serve as subjects or objects if their features align with the verb's agreement. Ojibwe also permits headless RCs as verbal arguments. We approximate these by letting an internal agreement slot on the RC verb stand for the head; the RC then attaches to an agreeing verb. The current constraint might lead to cases of overparsing, and points to an area for future development.

5.4. Parsing other dependencies

Beyond core arguments, we implement constraints to parse obliques, locatives, and certain adverb classes. One area of interest is locative elements, which were mentioned in section 2.4.2. Locatives pattern fairly simply, where any nominal locative attaches to the nearest verb as an oblique argument with the `obl` dependency relation. Some locative adverbs behave like adpositions, therefore, when a locative adverb is immediately followed by a locative noun, we analyze the adverb as a case marker on that noun; this matches UD's treatment of adpositions as dependents of the noun.

We keep the remaining adverb classes simple. Temporal and manner adverbs attach as `advmod` to the closest verb. Degree adverbs prefer to modify a neighboring adverb and otherwise fall back to the closest verb. For negation and prohibitives, we simply attach *gaawiin* as `neg` to the closest negative verb on the right and *gego* as `neg` to the closest prohibitive verb on the right.

5.5. Conversion and Post-processing

After building dependencies, we perform a cleanup by removing helper tags and mapping FST POS tags to CoNLL-U UPOS tags. This ensures that the CoNLL-U conversion is deterministic even when the FST outputs multiple compatible analyses.

The CG output is then converted to CoNLL-U. Heads and relations come directly from the `SETPARENT` and `ADD` passes, while UPOS and XPOS combine the FST tags with the post-processing aids described above. Figure 1 provides a summary of how the CG output maps onto CoNLL-U columns.[2]

CoNLL-U Column	**Source in CG Output**
`ID`	Token index
`FORM`	Surface form
`LEMMA`	FST lemma
`UPOS`	From FST + post-processing
`XPOS`	From FST tags
`HEAD`	From `SETPARENT`
`DEPREL`	From `ADD`

Figure 1: Mapping between CG output fields and CoNLL-U columns.

In summary, the disambiguation grammar resolves lexical and feature ambiguity, the dependency grammar assembles hierarchical relations through agreement-based heuristics, and the CoNLL-U layer serializes the final structure for downstream use.

6. Evaluation

Having introduced the functionality of both grammar modules, we now outline the creation of evaluation sets and the results of both CGs.

6.1. Creating the gold standards

6.1.1. The disambiguation gold standard

The disambiguation gold standard is based on a random sample of 492 examples (1906 total word tokens) from the OPD. We sample examples in parallel for both disambiguation and dependency evaluation, so we do not require every case to have an ambiguity. Also, examples with unparsed words are not included, as we want to keep evaluation focused on the CGs. Missing parses are due to gaps in the FST, not the current system. To keep disambiguation evaluation transparent, we evaluate based on the number of individual ambiguities resolved.

To create the gold standard, we first converted the example sentences into CG-3 input format. Then, a teammate with extensive Ojibwe knowledge manually disambiguated the CG-3 formatted file, which was done by deleting unwanted parses and confirming the remaining parse was correct. All forms were disambiguated to have only one reading, even in cases where this was only possible by looking at the corresponding English translation to get sentential context. This is based on the principle that sentences are almost always uttered with a single intended meaning in mind, and so we aim to approximate how close the CG can approach a truly "gold standard". This requires us to carefully distinguish between cases where the disambiguation grammar fails due to genuine limitations in the rules, and cases where it fails because the sentence is inherently ambiguous without broader context.

6.1.2. The dependency gold standard

The gold standard for the dependency grammar was built by first parsing the system's dependency outputs on the manually disambiguated gold set, and then converting these into a CoNLL-U file. The same 492 examples (1096 word tokens) were then manually corrected by the same teammate with extensive Ojibwe knowledge by fixing incorrectly assigned links and adding missing dependencies.

At present, the gold standard does not attempt to cover every possible dependency relation. Instead, it focuses on the set of thirteen relations that our grammar currently models[3]. While creating a complete gold standard with all dependency relations assigned is a necessary future step, this choice was influenced by time factors, as choosing precise relations requires careful discussion and certain linguistic compromises, as the UD format is not always well-suited for modeling dependencies in morphologically complex Algonquian languages (Wagner et al., 2016). As such, the evaluation is targeted on the dependencies that are actively assigned, with particular emphasis on argument structure and nominal constituents.

6.2. Evaluating the two grammars

6.2.1. Disambiguation evaluation

The disambiguation grammar was evaluated by comparing its output against the gold standard. For each token with multiple FST analyses, we

---

[2] `FEATS`, `DEPS`, and `MISC` columns are excluded.

[3] These thirteen relations are: `nsubj`, `obj`, `iobj`, `det`, `obl`, `nummod`, `discourse`, `case`, `advmod`, `neg`, `acl:relcl`, `csubj`, and `ccomp`.

Table 1: Disambiguation results by class (n ≥ 10)

Contrast	n	✓	Fail	Acc	Cov	Kept
OBJ{0Pl, 0Sg}	54	40	14	74.1	100	1.26
FORM{ChCnj, Pcp}	50	0	50	0	100	2.22
NOUN{ObvPl, ObvSg, Pl}	41	7	34	17.1	100	1.90
OBJ{3PlObv, 3SgObv}	37	0	37	0	97.3	2.00
NOUN{ObvPl, ObvSg}	30	0	30	0	100	2.00
POS{ADVInter, PCInterj}	25	25	0	100	100	1.00
FORM{ChCnj, Cnj}	24	4	20	16.7	100	2.08
SUBJ{0Pl, 0Sg}	23	17	6	73.9	100	1.26
NOUN{NA, NI}	22	21	1	95.5	100	1.05
PV{daa, ga}	22	0	22	0	100	2.00
OBJ{3PlProx, 3SgProx}	20	17	3	85.0	95.0	1.15

recorded whether the grammar produced the correct gold-standard reading. We further grouped results into ambiguity classes (e.g., number, POS, or verbal form contrasts) to identify where the grammar performs well and where it struggles.

Across 1906 tokens, the grammar reduced total readings from 3369 to 2897, eliminating 472 analyses. This corresponds to removing 32% of redundant readings and 41% of token-level ambiguity.

Table 1 summarizes performance by contrast type. The grammar achieves close to exact disambiguation in some cases, but accuracy varies across classes. For instance, the number ambiguity with verbs taking obviative objects has no matches, which is expected, given the fact that agreeing obviative nouns are also ambiguous in number. Additionally, POS distinctions such as adverb (`ADV`) vs. particle (`PC`) were fully resolved, while verbal form and prefix contrasts like *daa-* vs. *ga-* (ambiguously realized as *ji-* in conjunct order) remain challenging.

Importantly, even when exact disambiguation was not achieved, the grammar consistently retained the gold-standard reading. Many remaining ambiguities appear to reflect limits of the local morphosyntactic context available to the grammar, though some may also point to areas for further rule development. Overall, the grammar preserves correct analyses while substantially reducing ambiguity.

6.2.2. Dependency evaluation

To evaluate the dependency grammar, we compared the system's CoNLL-U output against the manually corrected gold standard. Evaluation was restricted to the thirteen dependency relations that our grammar currently attempts to model. For each relation, we counted its gold and system occurrences, the number of exact matches, and computed precision, recall, and F1. In addition, we report overall accuracy across all tokens where at least one side (gold or system) contained one of the focus relations.

Table 2 shows the by-dependency results. The grammar performs strongly on core argument relations: `nsubj` and `obj` both show high precision and recall (0.94 and 0.97 F1). Similarly, the internal nominal `det` dependency reached near-perfect scores, and discourse-level relations like `discourse` and `neg` were also captured accurately. Performance was somewhat lower on adjunct relations such as `obl` and `advmod`, where recall dropped (0.77 and 0.87), indicating that some modifiers are left unattached or assigned to other categories.

More complex clausal relations remain challenging. `acl:relcl` (relative clause) was only identified with a 0.23 recall, and `ccomp` (clausal complement) was not captured at all. These are structures that often require deeper syntactic context and longer-distance linking, which our current grammar does not yet fully handle.

Overall, the grammar effectively models core arguments and simple nominals, but struggles with complex clausal relations, indicating that further rules are needed for higher-level dependencies.

Table 2: Per-relation dependency parsing results

REL	gold	sys	✓	prec	rec	f1
obj	191	182	176	0.97	0.92	0.94
nsubj	180	180	175	0.97	0.97	0.97
obl	164	129	127	0.98	0.77	0.87
advmod	153	133	133	1.00	0.87	0.93
det	143	139	139	1.00	0.97	0.99
discourse	86	86	86	1.00	1.00	1.00
acl:relcl	35	8	8	1.00	0.23	0.37
case	23	22	22	1.00	0.96	0.98
neg	15	15	15	1.00	1.00	1.00
nummod	9	4	4	1.00	0.44	0.62
iobj	8	4	4	1.00	0.50	0.67
ccomp	3	0	0	0.00	0.00	0.00
csubj	0	0	0	0.00	0.00	0.00

7. Discussion

We now turn to a broader discussion of the results and implications of the two CGs.

7.1. The effectiveness of rule-based disambiguation

The results above demonstrate that a CG approach is effective in resolving many types of morphological ambiguity using contextual information. In particular, contrasts involving noun animacy and POS distinctions are often successfully disambiguated, suggesting that the surrounding morphological and syntactic environment provides sufficient cues for the grammar to apply the correct rules.

By contrast, distinctions that rely on discourse-level or semantic information, such as number contrasts in verbs with obviative arguments, remain difficult for the grammar to resolve. This can even be seen in English, where for the sentence “He saw the sheep”, discourse context would be crucial in telling us whether “sheep” is singular or plural. These patterns highlight the limits of rule-based disambiguation when the information is limited to morphological-level analyses.

Overall, the evaluation shows that rule-based disambiguation substantially reduces ambiguity while preserving correct analyses. Remaining ambiguities primarily reflect the inherent need for broader semantic or discourse context, rather than shortcomings in the CG rules themselves.

7.2. Modeling syntactic dependencies

CG is also an effective tool for modeling Ojibwe argument and nominal structure. Although Ojibwe is a “flexible word order” language (Sullivan, 2016), rich morphological information supplied enough context to accurately parse the vast majority of arguments, regardless of their linear position in the sentence string. Nominal structure was also effectively parsed, as a concrete modeling of the nominal domain allowed us to implement a minimal number of constraints to capture these dependencies. One such example is mentioned in section 5.2, where we model the Dem-Noun determiner relationship using linear precedence.

Room for improvement still lies in the parsing of relative clauses, which are subject to dialectal variation, and which are often ambiguous in form with changed conjuncts, verb forms which appear in other constructions such as questions and subordinate clauses (Sullivan, 2016, p. 14).

7.3. When is non-morphological information needed?

Finally, we turn to a question which was gradually answered through the development of both parsers. Namely, we wanted to understand what kind of higher-level information might be needed to address the limits of our current rule-based approach, such as the obviative number ambiguities discussed above. An interesting insight is that such cases might not be as ambiguous as originally predicted, as Sullivan (2016, p. 184-188) explains that many northern dialects of Ojibwe use an unambiguous plural obviative form. This points us to an interesting area for development, which would be to find a way to encode dialect-specific information in the Ojibwe CGs, allowing for more fine-grained treatment of the data. For example, if we know we are parsing text from a northern dialect that has the unambiguous obviative plural form, our parser could make the decision that all forms showing the non-plural variant are simply the singular form, thus removing the ambiguity.

Another case that would benefit from dialect-specific information would be in the recognition of RCs. Specifically, southern varieties of Ojibwe have morphologically distinct word forms for conjunct and RC verbs, while northern varieties are typically ambiguous between the two forms (Sullivan, 2016, p.304). A dialect-specific parser would be able to provide valuable linguistic context, thus improving the quality of both disambiguation and dependency parsing.

Otherwise, there are also cases that will always be ambiguous, regardless of dialect. Here, the clearest solution would be to manually resolve such ambiguities using the semantic context. Making a clear distinction between programmatically and manually solvable ambiguities is an important step in continuing the development of our Ojibwe CGs, and is greatly streamlined by the use of automated tools, including the CGs themselves.

8. Conclusion

This paper marks the first Ojibwe CG parsers, created for the purposes of disambiguation and dependency parsing. This adds to a growing collection of parsers for low-resource languages (Schmirler et al. 2018; Wiechetek et al. 2019). Although an older technology, CGs have proven to be an exceedingly useful tool. The rule-based grammar allowed for linguistically driven development, a key characteristic when working with low-resource languages.

The disambiguation grammar was able to cover most context-dependent cases, and the dependency grammar was highly effective in parsing argument and nominal dependencies. Both grammars remain under active development, with future steps being incorporating dialect-specific rules and beginning to parse clause-level dependencies. These developments will continue to improve the quality of the Ojibwe corpus and allow for the creation of an automated Ojibwe treebank, the first ever for an Algonquian language. These resources further the capabilities of linguistic work, benefiting scholars and students of Ojibwe.

9. Acknowledgments

We would like to thank Minh Nguyen for early contributions to the project, as well as Katherine Tu, Katie Schmirler, and the members of ELF-Lab for detailed feedback and discussion. This work was supported by a SSHRC Insight Grant (435-2023-0474) awarded to Hammerly.

10. Limitations

Several limitations remain in this first iteration of the Ojibwe CGs. First, the grammars have so far been tested only on OPD data, so some results may not generalize fully to other text types or dialectal varieties. In addition, the current dependency grammar focuses mainly on verbal argument structure and argument-internal dependencies, meaning certain clausal relations and long-distance dependencies are not yet fully modeled. The dependency evaluation is therefore targeted rather than a full measure of dependency parsing performance for Ojibwe as a whole.

11. Ethics Statement

The present work focuses on the North American Indigenous language known as Ojibwe. As such, we took care to uphold ethical principles such as data sovereignty. The data from the OPD, which formed the test set for our models, is available under a CC BY-NC-SA 3.0 license. This work falls squarely within the allowable uses under that license, and we are in communication with the team at the OPD to ensure we are properly respecting the data. Our team also includes a member of the Ojibwe community, and we regularly speak with elders, educators, and other community members about project directions and aims. While not the focus of the present work, we are deeply engaged in deploying our tools in language revitalization settings and in ensuring they are accessible to the community.

12. Resources and Reproducibility

Our GitHub repository with code, grammars, evaluation scripts, and documentation is available at `https://github.com/ELF-Lab/OjibweCG`.

13. Bibliographical References

Eckhard Bick and Tino Didriksen. 2015. CG-3 — beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 31–39, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *COLING-02: Grammar Engineering and Evaluation*.

Viann Sum Yat Chan and Christopher Hammerly. 2025. Evaluating indigenous language speech synthesis for education: A participatory design workshop on Ojibwe text-to-speech. In *Proceedings of the Eight Workshop on the Use of Computational Methods in the Study of Endangered Languages (ComputEL-8)*, pages 47–64. Association for Computational Linguistics (ACL).

Ann Grafstein. 1984. *Argument Structure and the Syntax of a Non-Configurational Language*. Ph.D. thesis, McGill University, Montreal, QC.

Christopher Hammerly. 2020. *Person-based Prominence in Ojibwe*. Ph.D. thesis, UMass Amherst.

Christopher Hammerly, Sonja Fougère, Giancarlo Sierra, Scott Parkhill, Harrison Porteous, and Chad Quinn. 2023a. A text-to-speech synthesis system for Border Lakes Ojibwe. In *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 60–65.

Christopher Hammerly, Sonja Fougère, Giancarlo Sierra, Scott Parkhill, Harrison Porteous, and Chad Quinn. 2023b. A text-to-speech synthesis system for Border Lakes Ojibwe. In *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages (ComputEL-6)*, pages 60–65. Association for Computational Linguistics (ACL).

Christopher Hammerly, Nora Livesay, Antti Arppe, Anna Stacey, and Miikka Silfverberg. 2026. Ojibwemorph: An approachable finite-state transducer for Ojibwe (and beyond). *Language Resources and Evaluation*, 60(2):1–44.

Christopher Hammerly, Adrian Staub, and Brian Dillon. 2022. Person-based prominence guides incremental interpretation: Evidence from obviation in Ojibwe. *Cognition*, 225.

Mary Hermes and Kendall A. King. 2013. Ojibwe language revitalization, multimedia technology, and family language learning. *Language Learning & Technology*, 17(1):125–144.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th Conference on Computational Linguistics (COLING 1990), Vol. 3*, pages 168–173, Helsinki, Finland. Association for Computational Linguistics.

Minh Nguyen, Christopher Hammerly, and Miikka Slifverberg. 2025. A hybrid approach to low-resource machine translation for Ojibwe verbs. In *Proceedings of the Fifth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)*, pages 18–26.

Nichols, John D. and Livesay, Nora. 2012. Ojibwe people's dictionary.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D.

Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Katherine Schmirler, Antti Arppe, Trond Trosterud, and Lene Antonsen. 2018. Building a constraint grammar parser for Plains Cree verbs and arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Statistics Canada. 2023. Indigenous languages across Canada (2021 census).

Michael Sullivan. 2016. *Relativization in Ojibwe*. Ph.D. thesis, University of Minnesota, Minneapolis, MN.

Naeem Uddin and Daniel Zeman. 2025. Universal Dependency treebank for a low-resource dardic language: Torwali. In *Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025)*, pages 140–147, Ljubljana, Slovenia. Association for Computational Linguistics.

J. Randolph Valentine. 2001. *Nishnaabemwin reference grammar*. University of Toronto Press, Toronto.

Irina Wagner, Andrew Cowell, and Jena D. Hwang. 2016. Applying Universal Dependency to the Arapaho language. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 171–179, Berlin, Germany. Association for Computational Linguistics.

Shenran Wang, Changbing Yang, Michael l Parkhill, Chad Quinn, Christopher Hammerly, and Jian Zhu. 2025. Developing multilingual speech synthesis system for Ojibwe, mi'kmaq, and maliseet. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 817–826, Albuquerque, New Mexico. Association for Computational Linguistics.

Linda Wiechetek, Sjur Nørstebø Moshagen, and Kevin Brubeck Unhammer. 2019. Seeing more than whitespace — tokenisation and disambiguation in a North Sámi grammar checker. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 46–55, Honolulu. Association for Computational Linguistics.

14. Language Resource References

Didriksen, Tino and Bick, Eckhard. 2025. *VISL CG-3: Constraint Grammar Compiler and Tools*. University of Southern Denmark; GrammarSoft ApS. Software tool; GPL-3.0 license. Accessed 2025-10-05.

Nichols, John D. 2012. *The Ojibwe People's Dictionary*. University of Minnesota, Department of American Indian Studies and University Libraries. University of Minnesota. PID https://ojibwe.lib.umn.edu/. Accessed 2025-10-05.

Universal Dependencies Consortium. 2024. *CoNLL-U Format (Universal Dependencies)*. Accessed 2025-10-05.