A hybrid approach to low-resource machine translation for Ojibwe verbs

Tran Minh Nguyen, Christopher Hammerly
Department of Linguistics
University of British Columbia
Vancouver, B.C., Canada
minhngca@student.ubc.ca, chris.hammerly@ubc.ca

Miikka Silfverberg
Independent
mpsilfve@iki.fi

Abstract

Machine translation is a tool that can help teachers, learners, and users of low-resourced languages. However, there are significant challenges in developing these tools, such as the lack of large-scale parallel corpora and complex morphology. We propose a novel hybrid system that combines LLM and rule-based methods in two distinct stages to translate inflected Ojibwe verbs into English. We use an LLM to automatically annotate dictionary data to build translation templates. Then, our rule-based module performs translation using inflection and slot-filling processes built on top of an FST-based analyzer. We test the system with a set of automated tests. Thanks to the ahead-of-time nature of the template-building process and the light-weight rule-based translation module, the end-to-end translation process has an average translation speed of 70 milliseconds per word. The system achieved an average ChrF score of 0.82 and a semantic similarity score of 0.93 among the successfully translated verbs in a test set. The approach has the potential to be extended to other low-resource Indigenous languages with dictionary data.

1 Introduction

Ojibwe is an Indigenous language of North America in the Algonquian family spoken in both the US and Canada. There are approximately 25,440 (Statistics Canada, 2023) in Canada, and likely not more than a few thousand speakers in the US. It is important to document and revitalize the language for the benefit of the Indigenous community and the learners. As recently discussed by (Littell et al., 2018), machine translation has the potential to help learners and reduce the workload of teachers.

However, it is a difficult task, because Ojibwe is a morphologically complex language, and there is not enough parallel data for modern neural machine translation. Similar in spirit to recent work by (Zhang et al., 2024), we propose a novel combination of advanced neural architecture such as LLM (Large Language Model) to annotate the dictionary data of Ojibwe to create translation templates, and from that, using rule-based translation computer program, to construct good English translations of inflected Ojibwe verbs. The present work was designed to overcome the challenges of not having enough data to build neural translation systems, while keeping the precision and speed of rule-based translations. The purpose is to help learners, teachers, and researchers.

There are currently no machine translation systems available for the Ojibwe language. Many of the current translation projects for lower-resourced languages like Ojibwe are rule-based (Littell et al., 2018), though there are exceptions such as the recent translation system developed by Google for Inuktitut (Caswell, 2024) and Meta's NLLB (Koishekenov et al., 2022). We know of one rule-based system for machine translation of an Algonquian language – Plains Cree – which has been integrated into the *itwêwina* dictionary (Arppe et al., 2022).

One important type of rule-based system, which can provide at least a partial solution for machine translation, are finite-state transducers (FSTs) or morphological parsers more generally (Zhang et al., 2024). Like all rule-based systems, FSTs have the advantage of only requiring meta-linguistic knowledge of morphophonological forms and rules and a dictionary of stems to get off the ground — there is no need for large collections of training data.[1] As such, FSTs are now relatively commonplace for

[1] It should be noted that, for some languages, even meta-linguistic descriptions in the form of grammars and dictionaries is uncommon. At the extreme, such languages could be seen not just as low-resourced, but unresourced when it comes to documentation and description. For these languages, it is still true that the task of creating a set of rules and collecting word lists is a far more tractable task than creating parallel corpora on the order of millions of tokens.

Proceedings of the Fifth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP), pages 18–26
May 4, 2025 ©2025 Association for Computational Linguistics

North American Languages (e.g. Harrigan et al., 2017; Bowers et al., 2017; Forbes et al., 2021; Hammerly et al., 2025). However, these systems generally produce abstract tags, rather than direct translations to another language such as English. In this paper, we show how these tags can be used as an intermediary form to guide rule-based translations.

2 Translation Approach

The system contains two main components: the Template Building module and the Translation module. The code for this project is publicly available in the OjibweTranslation repository (ELF-Lab, 2025).

2.1 Template Building module

The Template Building module has the main task of analyzing Ojibwe dictionary data, which is based on the Ojibwe People's Dictionary (OPD; Nichols, 2012). This data is openly available for use and adaptation by researchers and educators for non-commercial use under a Creative Commons license (Attribution-NonCommercial-ShareAlike 3.0 Unported License), with the explicit goal "to make the dictionary content available as a tool for Ojibwe language revitalization, academic scholarship and cultural awareness". Note, we have only released a limited set of verbs in the public version of the source code at the request of the editors of the OPD.

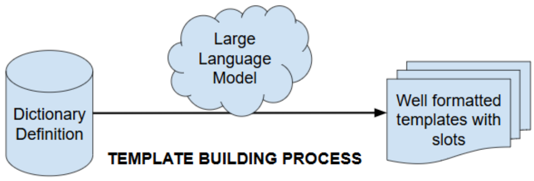

Figure 1: Template Building Process

Our basic process is schematized in Figure 1. We took dictionary data including the English-language definition and used an LLM to build templates with relevant slots. For example, the Ojibwe verb *waabam* defined in English as "**see h/**" (where "h/" means "him/her") becomes "**{{subject}} see {{object}}**". The purpose of building templates is to make it easier for the Translation module to replace these slots with appropriate pronouns or other information, according to the inflected verb.

Verbs in Ojibwe are separated into four basic types based on valency and animacy. Valency refers to whether a verb is intransitive (only a subject) or transitive (both a subject and object). Animacy restricts certain arguments of the verb based on grammatical noun class. All nouns in Ojibwe are grammatically categorized as "animate" or "inanimate", a roughly conceptual split that puts humans, animals, and most plants into one class (animate), and everything else into the other (inanimate). Animate Intransitive (AI) verbs have an animate subject, Inanimate Intransitive (II) verbs have an inanimate subject, Transitive Animate (TA) verbs have an animate object (but the subject can be any animacy), and finally transitive inanimate (TI) verbs have an inanimate object (and again, subjects can have any animacy).

In an early stage of the project, we attempted to use a rule-based approach to create templates. However, we quickly found that the significant inconsistencies in the way dictionary entries were formatted made such an approach untenable. While such inconsistencies do not at all get in the way of normal use – this is not a critique of the dictionary in general – this was a barrier for creating a simple set of rules that could work across all 15,000 verbs in our set. We therefore opted for an LLM approach, which allowed for more flexibility by creating examples and prompts, rather than hard-and-fast rules.

Our ultimate implementation used the Groq API provider, with a model named "**llama3-70b-8192**" based on Meta's Llama3. This particular approach also has the advantage of ensuring data is not passed on to a third party such as Meta (Groq does not use or retain data from prompts), which could potentially violate the license of the dictionary, or more generally afoul of Indigenous data sovereignty. In our case, the LLM is nothing more than a tool to get a specific job done: the annotation of thousands of dictionary entries. This job is not possible to complete with a purely rule-based approach (see above), and discussed later in the section, would be multiple of orders of magnitude less efficient if completed via purely human annotation.

We used the few-shot prompt strategy. The prompt included: (i) The initial instruction to ask the LLM analyze the context, subject and object; (ii) 10 to 20 human written examples; and (iii) A command to process new data. A sample prompt used for processing VTA verbs is in Appendix 5.

For example, with the same definition "see h/", we produce a transitive template "{{subject}} see {{object}}". Using slots such as {{subject}} and {{object}} makes it possible to build more complex sentences in the subsequent steps.

If the lemma definition has multiple meanings or glosses we instructed the LLM to split the definition into multiple templates. For example, with the word **niimaakwa'**, which have the definition "**pick it (animate) up or hold it (animate) out with something stick-like**", the system will produce the following templates:

- verbs: ['pick', 'hold'],
- templates:
  - "{{subject}} pick {{object}} up"
  - "{{subject}} hold {{object}} out with something stick-like"

It is important to emphasize that, while our Template Building module requires an LLM to extract and build templates, it is an ahead-of-time operation, meaning that we need to build the dictionary templates only once and export the templates to a computer-readable data format (such as csv). We do not need to run the template building process every time we do translation. We only need the exported data, which is stored locally, for translation in the subsequent steps. This increases the efficiency of translation.

Template building took about 3 seconds per example, which means about 12 hours of processing time for about 15,000 verbs. In comparison, if the task is to be done with a human annotator, it would take about 5 minutes per example, or about 1,250 hours of working time—a process that would also lead to high numbers of typos and other errors and inconsistencies. The LLM-based template building therefore resulted in an efficiency ratio of about 100 times, while maintaining favorable output quality.

2.2 Translation Module

The Translation module is a pipeline to transform the input (an inflected Ojibwe verb) through several steps to complete the final English translations. The process is schematized in Figure 2.

Important to note is that verbs in Ojibwe are morphologically marked for the person, animacy, and number of all arguments (using up to four distinct morphological slots), whether the predicate has a positive or negative polarity, and an aspectual distinction known as mode. The verb complex also contains certain morphologically dependent tense prefixes. All of these elements are part of the target

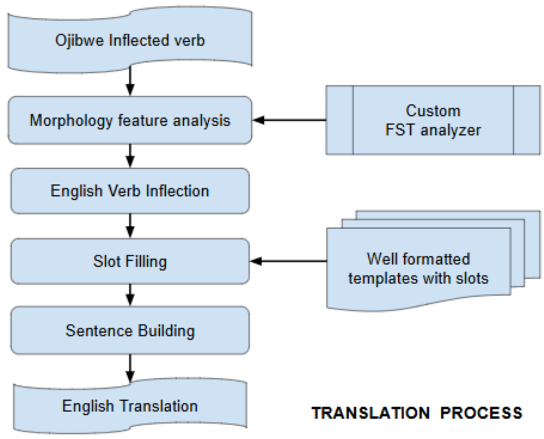

Figure 2: Translation Process

Paradigms	VTA, VTI, VAI(O), VII
Order	Independent, Conjunct, Imperative
Mode	Neutral, Preterit, Dubitative
Tense	Present, Definitive future, Past, Future/wish
Negation	Positive, Negative

Table 1: Supported verb properties. For each paradigm, all possible argument combinations are supported.

for our translation. A summary of the verb properties that can be handled by the translation model is given in Table 1.

Our translation module uses the following data sources:

- The dictionary and template data, built from previously mentioned Template Building Module.
- The FST binary file (in ".att" or ".fomabin" format) contains the rules for Ojibwe inflection, so that a FST parser can analyze the inflected input.

At the core of the Translation module are then the following operations:

- **Morphological feature analysis**: the Ojibwe verb is parsed by the FST to analyze and extract morphological features. It returns all important linguistics information such as the lemma, order, mode, subject, object, tense, negation, etc. in the list-of-tags format. For example, for the verb "giwaabamin" ("I see you" in

English), the FST parser returns the tag `waabam+VTA+Ind+Pos+Neu+1SgSubj+2SgObj`, which indicates the lemma "waabam" ("to see somebody" in English), the verb paradigm "VTA", the order "Independent", the polarity "Positive", the mode "Neutral", the subject "1st Singular", the object "2nd Singular". The FST parser is integrated into the translation system through a Python library called "fst_runtime"(CultureFoundry, 2025) made by CultureFoundry. The fst_runtime library uses compiled binary data of OjibweMorph (Hammerly et al., 2025) to process the Ojibwe input word and returns the analyses back to the translation system.

- **Verb inflection**: the inflection step considers the main English verb (of the English definition) in the infinitive form and the input Ojibwe FST context, which contains the subject, the mode, the tense and polarity. Then a set of custom rules is implemented in Python code to convert the infinitive English verb to the corresponding inflected English verb, which will be used in the subsequent slot-filling step. The sequence of FST tags to process English verb inflection is generally tense, then mode, then polarity (negation), then subject. To transform an infinitive verb, including irregular verbs, into different tenses such as past or perfect tense, it is done through a Python package called "pyInflect"(Jascob, 2023). Some examples of how a verb might be transformed depending on the context are given in Appendix C, Table 4.

- **Slot filling**: based on the subject and object of the sentence structure, it will replace the slots with relevant information, for example {{subject}} → "he/she" for 3SgSubj, and {{object}} → "me" for 1SgObj . The slot-filling process is illustrated in Figure 3.

- **Sentence building**: This builds a complete sentence from the template, using verb inflection and slot-filling operations. For example, the template "{{subject}} see {{object}}" → "He/she will not see me" for 3rd Singular subject, 1st Singular object, future tense, negative polarity, neutral mode, independent order.

Again, the translation pipeline is entirely rule-based, so it does not require direct use of LLMs.

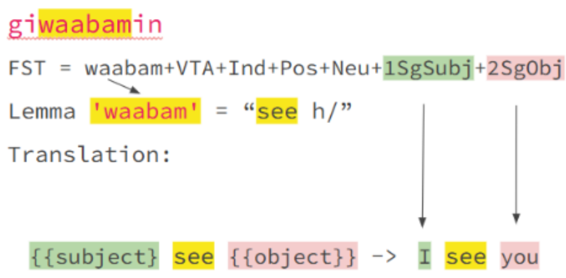

Figure 3: Slot-Filling Illustration

As such, the system produces transparent and predictable results and we can easily modify the rules to suit specific needs. A sample of translations is provided in Appendix B.

3 Evaluation

We completed two types of tests: Speed tests and translation accuracy tests.

3.1 Speed tests

For speed, our system processed a batch of 10 words for translation with an average speed of around 700 millisecond / batch, which means about 70 millisecond / word. The hardware used is a laptop with 24GB RAM and AMD Ryzen 7 5800H CPU, without using GPU (Graphical Processing Unit). That our system can run on standard hardware with limited computing power is a major benefit to making the tool accessible.

Because Ojibwe is a morphologically complex language and a single verb in Ojibwe can be translated into a full sentence in English. If the definition has multiple meanings, the output can contain several sentences or phrases in English. Therefore, the processing speed implies one Ojibwe verb input and one or more English sentences output, rather than one Ojibwe input word and one corresponding English output word.

3.2 Translation accuracy tests

We created a test set of inflected Ojibwe verbs, along with gold translations, from the University of Toronto Ojibwe Textbook (Meltzer et al., 2022-2023), available under the BY-NC-SA 2.5 CA. We selected only inflected verbs from the provided word list and performed simple data cleaning and normalization on the gold translations, including:

- changing abbreviations such as "s/he" to "he/she", etc.

Number of verbs in test set	214
Number of successful translated verbs	200
Percentage of successful translation	93%
Mean ChrF score	0.82
Mean Semantic Similarity score	0.93

Table 2: Evaluation scores

- removing punctuation
- removing extra information inside parentheses, such as "(ani.)", "(inc.)"
- keeping only one translation if there are multiple translations.

There are 214 inflected verbs included in the test set—a small, but reasonable, number due to the low-resource nature of Ojibwe. Our system was able to provide a translation for 200 of the 214 verbs (93%). Some verbs cannot be translated because of missing definition or stem in the database from the dictionary. Examples of comparisons between system and gold translations are illustrated in Appendix D, Table 5

We first calculated the ChrF score (Popović, 2015). The score is a real number between 0.0 (no overlap between translations) and 1.0 (perfectly matched translations). We used NLTK sentence_chrf function with parameters min_length=1 (unigram) and max_length=3 (3-gram) to calculate ChrF score between system and gold translation. If the system generates multiple translations, the translation with highest score was selected. Among the verbs that were successfully translated, the average score is 0.82, as summarized in Table 2.

We also performed a semantic similarity comparison between the system and the gold translations through the Sentence-BERT package (Reimers and Gurevych, 2019). and the LaBSE (Language-agnostic BERT Sentence Embedding) (Feng et al., 2020) embeddings model. Semantic similarity is useful in scenarios where the system and gold translations use synonyms, for example, "we will **enjoy** the taste of **things**" versus "we will **like** the taste of **something**". In this case, the semantic similarity score would be high, while the ChrF score could be considerably lower.

The semantic similarity score between two sentences is a real number between 0.0 (completely unrelated meanings) to 1.0 (perfectly aligned meanings). If the system produces multiple translations, the highest score was selected. Out of the successfully translated verbs, the average semantic similarity score is 0.93, as summarized in Table 2.

4 Applications

The translation package will be used in various settings and purposes, which include:

- Ojibwe language learners, teachers, and schools via a free web interface to analyze and understand complex inflected verbs.
- Researchers to produce an automated translation of Ojibwe verbs for downstream tasks, such as neural machine translation.

In addition to a ready-to-use Python package that can be easily integrated into current popular NLP pipelines, we also included a web application (see Figure 4 in Appendix A) built on the NiceGUI framework, so users such as teachers and students can use it easily without coding, making it more approachable to the general audience. We have yet to widely and systematically test this interface, but such testing is an aim of future work.

5 Future directions

There are a number of avenues for future work. First, the current system only works at the individual word level, so cannot yet handle full sentences. One potential rule-based way to augment the current system to handle full sentences is through the use of a constraint grammar to identify overt subject and object nouns, which could be fed to our rule-based translation module. Second, we are not yet able to translate from English to Ojibwe, nor from Ojibwe to a language other than English. Expanding the system for rules that work in the other direction, or for other languages, is another priority. Third, there is a small set of low-frequency verb forms not yet handled by the system, as well as the more general system of so-called lexical preverbs (which behave much like adverbs) that are not yet handled. Adding support for some of the most common lexical preverbs and expanding remaining tenses and modes in the functional domain is another direction for our future work. Finally, while the data from the Ojibwe People's Dictionary is robust, adding more words and definitions to improve coverage will be an ongoing task.

6 Ethics Statement

The present work was conducted in the context of a larger body of work by our research group to build computational tools relevant to language revitalization of Ojibwe. Our team includes a member of the Ojibwe community with linguistic training, and we have engaged in both formal and informal community consultation about our tools, including elders and teachers. We are committed to striking the balance between practicing open science and generating work that may find uses beyond the immediate community we are serving on one hand, while ensuring the integrity of the data and respecting the elders and community members who have created resources such as the Ojibwe People's Dictionary.

7 Limitations

The current system, although covered a wide range of Ojibwe paradigms and various grammar aspects such as order, mode, tense, etc., it still has some notable limitations such as:

- It works at word level, in particular Ojibwe verbs only. It does not yet have capability to translate other word types such as nouns, adjectives, etc. It is also not able to translate at sentence level, i.e. a full Ojibwe sentence to a full English sentence.

- Because of the diverse and potentially inconsistent format of the Ojibwe People's Dictionary definitions, some of the templates might not be extracted and built properly. We have not yet performed an exhaustive check on all template data. Some unusual definitions can lead to unusual templates, and in extreme case, we can not rule out templates that are not grammatically correct or do not make sense. It has the potential to produce inaccurate or ungrammatical translations in these cases. However, it is still likely to yield some meaningful text in the translations in these cases.

- Because of rule-base translation process, and it is not a neural translation model, therefore, it does not remember or learn all dictionary definitions. It requires external template data to do translation.

- The current system can translate Ojibwe to English, but is not yet able to translate English to Ojibwe.

- It can translate Ojibwe to English as the target language, but another target language, such as French, is not yet supported.

- Although the system supports an extensive grammar range of Ojibwe verbs, it does not fully cover all aspects of the verbs yet. For example, such as preterit-dubitative, which means uncertainty about a past completed event (Valentine, 2001) is not yet supported.

- Due to the low-resource nature of the Ojibwe language, we have not yet built a larger gold-standard test set to better evaluate the performance and quality of the system.

References

Antti Arppe, Jolene Poulin, Eddie Antonio Santos, Andrew Neitsch, Atticus Harrigan, Katherine Schmirler, Daniel Hieber, Ansh Dubey, and Arok Wolvengrey. 2022. Towards a morphologically intelligent and user-friendly on-line dictionary of Plains Cree – next next round. In *Presentation at the 54th Algonquian Conference, Boulder, CO.*

Dustin Bowers, Antti Arppe, Jordan Lachler, Sjur Moshagen, and Trond Trosterud. 2017. A morphological parser for Odawa. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 1–9.

Isaac Caswell. 2024. Google translate learns inuktut. Accessed: 2024-11-05.

CultureFoundry. 2025. fst-runtime. `https://github.com/CultureFoundryCA/fst-runtime`. Accessed: 2025-03-01.

ELF-Lab. 2025. Ojibwe translation repository. Accessed: 2025-03-28.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Clarissa Forbes, Garrett Nicolai, and Miikka Silfverberg. 2021. An FST morphological analyzer for the Gitksan language. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 188–197.

Christopher Hammerly, Nora Livesay, Antti Arppe, Anna Stacey, and Miikka Silfverberg. 2025. Ojibwe-morph: An approachable finite-state transducer for Ojibwe (and beyond). *Preprint Submitted to Language Resources and Evaluation.*

Atticus G Harrigan, Katherine Schmirler, Antti Arppe, Lene Antonsen, Trond Trosterud, and Arok Wolvengrey. 2017. Learning from the computational modelling of Plains Cree verbs. *Morphology*, 27:565–598.

Brad Jascob. 2023. pyinflect: A python module for word inflections designed for use with spacy. `https://github.com/bjascob/pyinflect`.

Yeskendir Koishekenov, Alexandre Berard, and Vassilina Nikoulina. 2022. Memory-efficient nllb-200: Language-specific expert pruning of a massively multilingual machine translation model. *arXiv preprint arXiv:2212.09811*.

Patrick Littell, Anna Kazantseva, Roland Kuhn, Aidan Pine, Antti Arppe, Christopher Cox, and Marie-Odile Junker. 2018. Indigenous language technologies in Canada: Assessment, challenges, and successes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2620–2632.

Jed Meltzer, Michael Sullivan Juvenal Ndayiragije Lindsay Morcom Marie-Odile Junker Inge Genee John-Paul Chalykoff Callie Hill Maureen Buchanan Perry Bebamash, with contributions from Gordon Jourdain, and Conor Quinn. 2022-2023. Ojibwe textbook: Downloadable resources. Funded by SSHRC Partnership Development Grant between Rotman Research Institute at Baycrest Hospital, University of Toronto, and Kingston Indigenous Languages Nest. Accessed: 2025-03-27.

John D. Nichols. 2012. Ojibwe People's Dictionary.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084*.

Statistics Canada. 2023. Indigenous languages in Canada, 2021.

Randy Valentine. 2001. *Nishnaabemwin reference grammar*. University of Toronto Press.

Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15654–15669.

Acknowledgments

We acknowledge Scott Parkhill from Culture-Foundry for advice and guidance on code quality and organization, and Antti Arppe from the University of Alberta for comments on early versions of this project. This work was supported by a SSHRC Insight Grant (435-2023-0474) awarded to Hammerly and Silfverberg.

A Web Application interface

A screenshot of the built-in Web Application interface Figure 4.

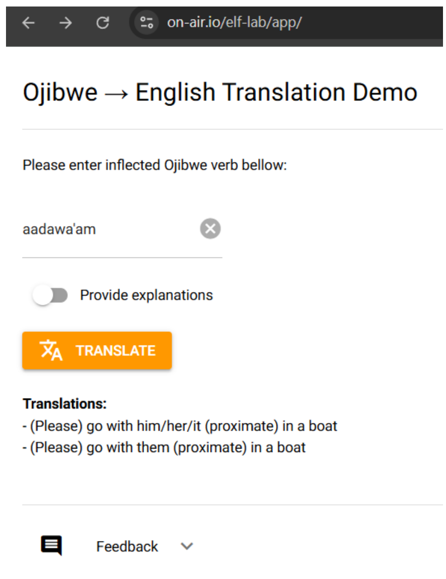

Figure 4: Web Application Interface

B Translation examples

Examples of Ojibwe verb inputs and their corresponding translations are provided in Table 3

Ojibwe verb	English translation
odaadawa'amaan	he/she (proximate) goes with him/her/it (obviative) in a boat
	he/she (proximate) goes with them (obviative) in a boat
aadawa'am	(Please) go with him/her/it (proximate) in a boat
	(Please) go with them (proximate) in a boat
abweninjii	he/she (proximate) has a sweaty hand
gaawiin gii-abweninjiisiin	he/she (proximate) did not have a sweaty hand

Table 3: Translation examples

English verb	Context	Inflected verb
be	3rd Singular Subject ("He/she"), present tense, positive polarity	(he/she) is
be	1st Plural subject ("We"), past tense, negative polarity	(we) were not
be	1st Singular subject ("I"), future/wish tense, positive polarity	(I) want to be
dance	3rd Plural subject ("They"), Dubiative mode, past tense, positive polarity	(They) might have danced
dance	2nd Singular subject ("You"), neutral mode, future tense, negative polarity	(You) will not dance
dance	1st Singular subject ("I"), preterit mode, past tense, positive polarity	(I) used to dance

Table 4: Verb inflection examples

C English verb inflection examples

Examples of how some English verbs are transformed and inflected according to the input Ojibwe FST context (subject, tense, mode, negation, etc) are provided in Table 4

D System versus Gold translation examples

Examples of system (hypothesis) translations compared with gold (reference) translations of inflected Ojibwe verbs are included in Table 5. Note that extra information inside parentheses was removed in both gold and system translations before calculating ChrF and semantic similarity scores.

E Prompt used for VTA verbs

A screenshot of the prompt used to create templates for VTA verbs, with LLM model "llama3-70b-8192" can be found in Figure 5.

Ojibwe verb	Gold translation	System translation	ChrF score	Semantic Similarity score
nimbakade	I am hungry	I am hungry	1.0	1.0
gibakade	you are hungry	you are hungry	1.0	0.99
apatoo	he/she runs	he/she (proximate) runs in a certain way	0.87	0.69
nimindid	I am big	I am big	1.0	0.99
wii-wiisini	he/she want/will eat	he/she (proximate) wants to eat	0.65	0.95
izhaa	he/she is going to a certain place	he/she (proximate) goes to a certain place	0.75	0.99
niwaabamaag	I see them	I see them (proximate)	1.0	1.0
nindizhaa	I am going to a certain place	I go to a certain place	0.71	0.99

Table 5: Gold versus System translations

```
prompt_template = """A given definition example: d =  "smudge, cense h/; smoke h/ (for preservation)".
Analyze the definition d. What is subject and object? Rewrite definition by replacing subject and object by literal `{{subject}}` and `{{object}}`.
Replace verbs to infinitive form (e.g. wants -> want, is -> be, gets -> get).
Answer in form {"verbs":[], "templates":[]}. Split the definition for each main verb.
Note the words like "something" or "(it)", don't parse them as "{{object}}", keep them as literal.
Translate "h/ or it" to "{{object}}".
Extract the main verbs only, if the sentence is in passive voice, the main verb is "be". The answer for definition d should be in JSON format
output = {verbs:["smudge", "cense", "smoke"],"templates":["{{subject}} smudge {{object}}", "{{subject}} cense {{object}}", ""{{subject}} smoke {{object}} (for preservation)"]}.
Do not invent new verbs. Keep the new definitions literally close as the original definition. Keep things in brackets as literal, e.g. (it), (something) or (by someone).

Bellow are more examples:

Definition = pull h/ aboard
Output = {'verbs': ['pull'], 'templates': ['{{subject}} pull {{object}} aboard']}
--------------------
Definition = fix, repair (it) for h/
Output = {'verbs': ['fix', 'repair'], 'templates': ['{{subject}} fix (it) for {{object}}', '{{subject}} repair (it) for {{object}}']}
--------------------
Definition = throw h/ aboard
Output = {'verbs': ['throw'], 'templates': ['{{subject}} throw {{object}} aboard']}
--------------------
Output = {'verbs': ['cool'], 'templates': ['{{subject}} cool {{object}} with water']}
--------------------
Definition = cook it (animate)
Output = {'verbs': ['cook'], 'templates': ['{{subject}} cook {{object}} (animate)']}
--------------------
Definition = throw (it) here to h/
Output = {'verbs': ['throw'], 'templates': ['{{subject}} throw (it) here to {{object}}']}
--------------------
Definition = cut it (animate; sheet-like) short
Output = {'verbs': ['cut'], 'templates': ['{{subject}} cut {{object}} ((animate; sheet-like) short ']}
--------------------
Definition = cut it (animate) so wide
Output = {'verbs': ['cut'], 'templates': ['{{subject}} cut {{object}} (animate) so wide']}
--------------------
Definition = staunch h/ bleeding
Output = {'verbs': ['staunch'], 'templates': ['{{subject}} staunch {{object}} bleeding']}
--------------------
Definition = ride mounted on top of h/; sit astride h/
Output = {'verbs': ['ride', 'sit'], 'templates': ['{{subject}} ride mounted on top of {{object}}', '{{subject}} sit astride {{object}}']}
--------------------
Definition = warm something (liquid) up for h/
Output = {'verbs': ['warm'], 'templates': ['{{subject}} warm something (liquid) up for {{object}}']}
--------------------
Definition = warm something for h/ at the fire
Output = {'verbs': ['warm'], 'templates': ['{{subject}} warm something for {{object}} at the fire']}
--------------------
Definition = warm h/ foot or feet
Output = {'verbs': ['warm'], 'templates': ['{{subject}} warm {{object-possessive}} foot or feet']}
--------------------
Definition = catch up to h/ following h/ tracks or trail
Output = {'verbs': ['catch'], 'templates': ['{{subject}} catch up to {{object}} following {{object-possessive}} tracks or trail']}
--------------------
Definition = dye, color h/ or it (animate)
Output = {'verbs': ['dye', 'color'], 'templates': ['{{subject}} dye {{object}} (animate)', '{{subject}} color {{object}} (animate)']}
--------------------
Definition = dye, color (it) for h/
Output = {'verbs': ['dye', 'color'], 'templates': ['{{subject}} dye (it) for {{object}}', '{{subject}} color (it) for {{object}}']}
--------------------

Now process a new definition
"""
```

Figure 5: Prompt used for VTA templates