Formal Machine Interpretation for the Semasiographic Mixtec Codices of Precolonial and Early Colonial Mesoamerica

Christopher Driggers-Ellis
University of Florida
driggersellis.cw@ufl.edu

Gabriel Ayoubi
University of Florida
gabriel.ayoubi@ufl.edu

Girish Salunke
Independent
girish.salunke811@gmail.com

Christan Grant
University of Florida
christan@ufl.edu

Abstract

The precolonial and early colonial Mixtec codices describe the history and stories of the region in a semasiographic medium that is full of symbolic representations and meant to be narrated. Recently, the community has introduced datasets of XML representations of related media, including Aztec codices and Mayan hieroglyphic script, in a step towards symbolic machine interpretation of these historic Mesoamerican artifacts. In this work, we propose formal symbolic machine interpretation of XML encodings representing facsimile images from the Mixtec Codex Zouche-Nuttall. We demonstrate the efficacy of symbolic machine interpretation from XML step-by-step, showing how our parser and interpreter process text capturing a scene from the Mixtec Codex Zouche-Nuttall. We hope our contribution and the example we provide motivate collaboration among the archaeological, historical, linguistic, and natural language processing research communities to apply machine interpretation to Mixtec codices and similar manuscripts.

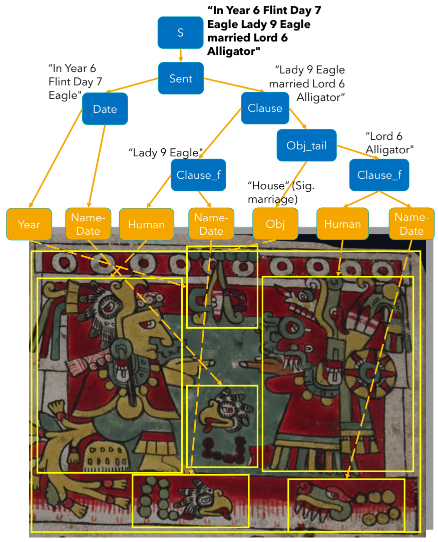

Figure 1: The abstract syntax tree (AST) for the sentence "In Year 6 Flint Day 7 Eagle Lady 9 Eagle married Lord 6 Alligator" from Page 26 of the Codex Zouche-Nuttall (obverse) (Provided courtesy of the British Museum). **Nonterminal** tree nodes are in Blue. **Tokens** corresponding to the glyphs are in Orange.

1 Introduction

The Mixtec codices are pictographic manuscripts left by the Mixtec peoples of precolonial and early colonial Mesoamerica and are an example of semasiographic media. Modern examples of semasiographic media include comics, manga, many street signs, storyboards, etc. Like comics or motion picture storyboards (Boone, 2000; Byland and Pohl, 1994), Mixtec Codices use human glyphs interacting with objects and each other, mediated by artistic conventions, to convey legends and histories about ancestors and/or to record those ancestors' genealogies (Williams, 2013; Boone, 2000). A long body of historical and linguistic literature explores human interpretation of Mixtec Codices (Williams, 2013), but there is little or no work investigating machine interpretation of the codices. Our primary contribution is therefore a recursive descent parser and interpreter for XML representations of Mixtec scenes, and we will discuss this contribution in the context of a broader machine interpretation pipeline to drive discussions about future work necessary to automatically generate this XML representation.

Figure 1 provides an example of the parser at work. The semasiographic interpretation pipeline is elaborated in Figure 2, and the grammar is provided in Figure 6. Our code is available online.[1]

2 Related Work

For more than a century, interpretations of the pictographic codices left behind by the indigenous peoples of Mexico, the Mixtec, have been offered in the academic literature (Williams, 2013). In

[1] https://github.com/ufdatastudio/mixtec-parser

Proceedings of the 4th Workshop on Advances in Language and Vision Research (ALVR), pages 230–238
July 3, 2026 ©2026 Association for Computational Linguistics

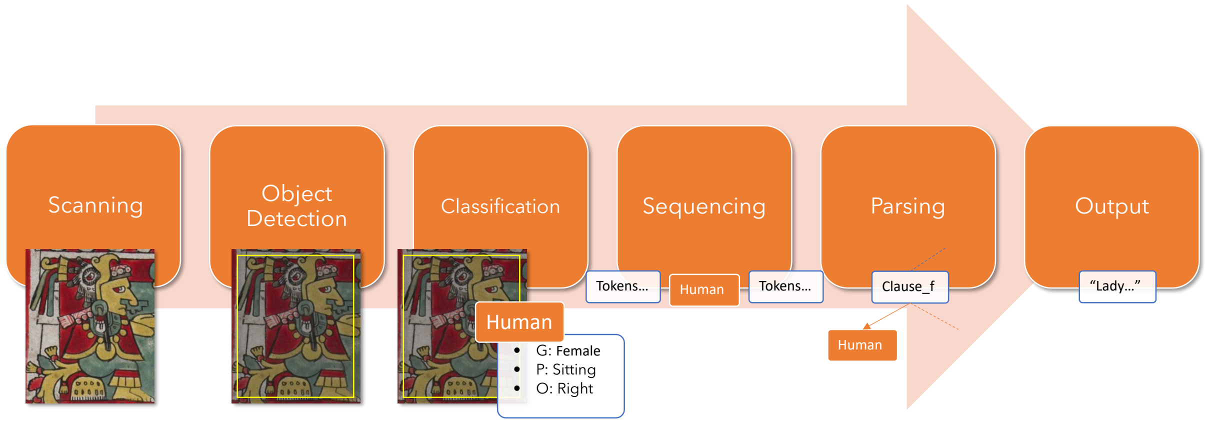

Figure 2: Semasiographic machine interpretation pipeline processing a single human glyph. (Example provided courtesy of the British Museum.) *Scanning* receives the input image. *Object Detection* detects the human glyph. *Classification* extracts its features and assigns them to a human token. *Sequencing* places the token in sequence with other tokens. *Parsing* places it into an AST. Finally, *Output* says "Lady" corresponding to the glyph's gender.

this work, we focus on an example of pictographic Mixtec writing from Page 26 of the Codex Zouche-Nuttall (CZN) (obverse) which was first published in facsimile by Zelia Nuttall and the Peabody Museum at Harvard University in 1902 (Williams, 2013; Nuttall, 1902). Williams (2013) dates the writing of this manuscript to the 14th or 15th centuries. Since its initial publication for the modern audience more than 120 years ago, the CZN has been reproduced, analyzed, and interpreted by numerous authors (Long, 1926; Caso, 1977; Troike, 1978; Boone, 2000; Williams, 2013). Despite this rich history of human interpretation and recent advancements in computer vision and machine learning technologies, machine interpretation of Mixtec codices like the CZN is not well studied.

Related works in the current century take steps toward machine interpretation for similar Mesoamerican manuscripts. Perri et al. (2024) explore encoding and interpreting the nested features corresponding to agglutinative linguistic elements present in a folio of the Aztec Codex Mendoza (Perri et al., 2024; Fedorova et al., 2024). Other research develops an XML encoding schema for the hieroglyphic script of the Maya (de la Iglesia et al., 2021). We are therefore inspired to persue sybmolic interpretation of the Codex Zouche Nuttall through XML representation. However, unlike the agglutinative Codex Mendoza or Mayan hieroglyphic scripts in these examples, Mixtec codices are pictographic manuscripts that do not correspond systematically to specific linguistic content. Instead, they are read semasiographically like a storyboard or comic book (Boone, 2000; Byland and Pohl, 1994).

We are most interested in three glyph types from Mixtec codices: human glyphs, name-dates, and toponyms. Human glyphs represent persons, normal human beings or deities, in the Mixtec Codices; name-dates correspond to calendar dates or names of characters; and toponyms convey setting information (Boone, 2000; Williams, 2013).

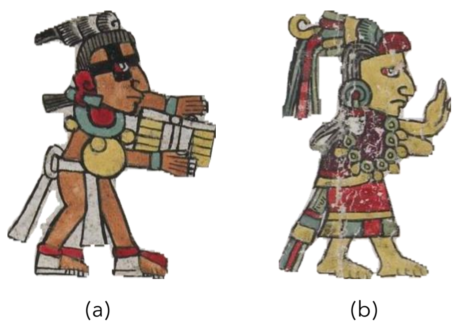

Figure 3: **Human Glyphs:** male Lord (a) and female Lady (b) human glyphs from the CZN (Provided courtesy of the British Museum).

Human Glyphs Interpretation of human glyphs relies on artistic conventions regarding the characters' genders, poses and orientations. Webber et al. (2024) propose classifiers to determine whether glyphs are male/female (Gender) or standing/sitting (Pose). Binary classifiers learn classification on isolated glyphs for the gender and pose attributes with >97% test accuracy. Examples of male and female human glyphs are shown in Figure 3.

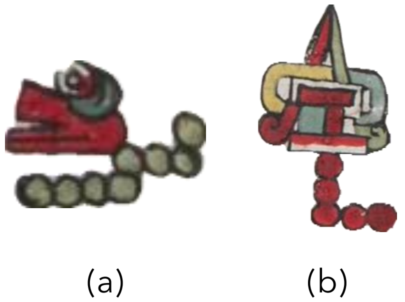

Figure 4: **Name-Date Glyphs:** Name/Date 8 Wind (a) and Year 5 House (b) from the CZN (Provided courtesy of the British Museum). Note the distinguishing presence of the AO symbol in (b).

Figure 5: **Toponyms:** Typonym representing “Yucu Dzaa” or “Hill of the Bird” (Boone, 2000) from page 45 of the CZN (reverse) (Provided courtesy of the British Museum).

Names-Date Glyphs Transcriptions of name-date glyphs take forms like “4 Wind” or “10 Serpent,” when they represent names, or forms like “Day 1 Alligator” or “Year 13 Rabbit,” when they represent days and years, respectively. Examples of a name-date and a year are provided by Figure 4. Name-date interpretation is more mechanical than interpreting the other glyphs. Glyphs are comprised of one of 20 symbols (only 4 are valid for years) and a number 1 through 13 according to the number of beads affixed to the glyph. First, one classifies glyphs as names/dates or years based on the presence of an ‘AO’ symbol (Boone, 2000), visible in Figure 4 (b), corresponding to a binary classification task. Second are two independent multiclass classification tasks. One is to recognize the name-date’s symbol, and the other to *count* (classify according to quantity) bead subglyphs that convey the name-date’s number (Boone, 2000). Results for these classification tasks already exist in the Mixtec machine interpretation literature (Salunke et al., 2025).

Toponyms Toponyms (place signs), complex glyphs in Mixtec codices that typically correspond to a place, often indicate the setting of events recorded in the codices (Boone, 2000; Williams, 2013). Interpretation of toponyms is nuanced and often relies on archaeological evidence (Smith, 1973; Pohl, 1994; Boone, 2000) and sometimes word-play in the Mixtec language (Boone, 2000). As an example, we include a toponym that reads “Yucu Dzaa” or “Hill of the Bird.” (Boone, 2000) from page 45 of the CZN (reverse) as Figure 5.

3 Methodology

In this work, we introduce a formal symbolic machine interpretation of an XML representation of Mixtec codices with aspirations of integrating it into a broader pipeline. Figure 2 lays out this pipeline in detail. As our contribution, we provide a context free grammar (CFG), parser, and interpreter for the Mixtec codices. We situate the XML parser and interpreter at the final *Parsing* and *Output* steps of the semasiographic machine interpretation pipeline we introduce. Assuming correct tokenization (in the XML form defined by the grammar) at the *Sequencing* step and correctly populated attributes at the *Classification* step, we implement *Parsing* by devising a CFG and implementing an ANTLR4-generated parser (Parr, 2013). We accomplish *Ouput* via the interpreter.

3.1 Rationale

We employ *ANTLR recursive descent parsing* since the relationship between the Mixtec glyphs is hierarchical in nature. By defining the last stages of the pipeline first, we establish a precise design contract for the earlier feature extraction steps. In particular, the parser’s need to resolve nested structures, such as a name-date modifying a human, requires that a finished pipeline must capture not just glyph identity and position, but relational attributes like orientation and containment in *Sequencing*

3.2 Token Structure

In the CFG elaborated by Figure 6, the following token types appear: **Humans** (`H`), **Years** (`Y`), **Name-Dates** (`ND`), **Objects** (`OBJ`), and **Near Objects** (`NEAR_OBJ`). These correspond to classes of glyphs that appear in Mixtec

```
document    → XML_DECL DOCUMENT_OPEN (
               SCENE_OPEN sent SCENE_CLOSE
              )+ DOCUMENT_CLOSE
sent        → clause
              | OBJ ( date clause | clause )
              | date ( OBJ clause | clause )
clause      → clause_f+ ( date_tail | obj_tail | ε )
date_tail   → date ( OBJ clause_f+ | clause_f+ | ε )
obj_tail    → OBJ ( date clause_f+ | clause_f+ | ε )
date        → Y ( ND | ε )
clause_f    → H ( ND | near_date | ε )
near_date   → NEAR_OBJ ( ND | ε )
```

Figure 6: The Context Free Grammar (CFG) implemented by the ANTLR4 parser to check whether Mixtec tokens sequences are valid for interpretation. Uppercase items denote lexer tokens; document is the start rule enclosing one or more scenes, and sentences (sent) take object-initial, date-initial, or clause-only forms.

codices. In addition, XML_DECL, DOCUMENT_OPEN, DOCUMENT_CLOSE, SCENE_OPEN, and SCENE_CLOSE are structural tokens used to define the document format. The start symbol document requires an XML declaration and a document wrapper containing one or more scene elements. An enumeration of the token types appears in Table 1.

Tokens of type H represent human glyphs from the Mixtec codices and have three attributes: **Gender (Male/Female)**, **Pose (Sitting/Standing)**, and **Orientation (Left/Right)**. Year and Name-Date tokens encapsulate **Symbol (One of 20 Symbol Names)** and **Number (Integer, 1-13)** attributes.

Tokens of the obj and near_obj types simply have an **Identity** attribute that labels the object for the parser and interpreter. Assignment of the **Identity** attribute corresponds to a straightforward object recognition task performed on object glyphs. The distinction between these two types of token is that near_obj represent objects that are associated with a specific person, giving context about them.

3.3 Parser

The Mixtec parser is designed for the *Parsing* step of the semasiographic interpretation pipeline in Figure 2 and implements the grammar in Figure 6. It takes XML input, and an example of this XML follows this paragraph. It is built using an ANTLR4 lexer and parser, which process the tokens assuming the token structure we elaborate in the previous section. The lexer recognizes each glyph-bearing XML element (e.g., <human .../>, <year .../>, <obj .../>) as a single token, and the parser applies the context-free rules for documents, scenes, and sentences.

As a concrete example, consider the following XML-form input:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <scene>
    <year x="0" y="0" symbol="House" number="5"
        />
    <human x="0" y="0" gender="Lord" pose="
         standing" orientation="right" />
    <name_date x="0" y="0" symbol="Wind" number
         ="4" />
    <human x="0" y="0" gender="Lady" pose="
         sitting" orientation="left" />
    <name_date x="0" y="0" symbol="Serpent"
         number="10" />
  </scene>
</document>
```

The parser and interpreter process this input into the following output:

> *"In Year 5 House Lord 4 Wind consulted Lady 10 Serpent."*

The ANTLR4 parser produces a concrete parse tree for valid inputs. A custom visitor then traverses this tree, converts the ANTLR tokens into the internal token classes described in the previous section, and constructs the corresponding abstract syntax tree (AST).

3.4 Interpreter

Our interpreter implements the *Output* step of the machine interpretation pipeline by applying a visitor pattern over the ANTLR4 parse tree. The visitor methods construct an abstract syntax tree (AST) from the parser rule contexts (PRCs) and attach interpretation rules to each node.

The algorithm is as follows. For each internal node r (corresponding to a parser rule context):

1. Visit all of r's children and collect their return values.
2. Construct a TreeNode object of the appropriate type (e.g., Clause, Date, Sent) using the results from step (1).
3. Return the constructed object if r is not the start symbol. If r is the start symbol (document), return the result of calling interpret() on the root node, which produces the final textual interpretation.

For each leaf node l (terminal token):

1. Read the XML string of the token.
2. Construct the appropriate token object (e.g., Human, Year, Name-Date, Object, Near-Object), populating its attributes from the XML.
3. Wrap the token in a LeafNode and return it.

Figure 7: Example from Page 26 of the CZN (obverse) (Provided courtesy of the British Museum) shown in Figure 1. The AST and annotations are removed to exhibit image features. **Interpretation: “In Year 6 Flint Day 7 Eagle Lady 9 Eagle married Lord 6 Alligator”**

Each `TreeNode` implements an `interpret()` method that specifies how its children combine into semasiographic meaning. For example, a `Clause` node may contain a rule recognizing the Mixtec motif of two seated persons of opposite gender in a house, which corresponds to a marriage (Boone, 2000). A `Date` node joins a year (`Y`) and an optional name-date (`ND`), while a `Sent` node concatenates its parts into a sentence. The start rule (`document`) aggregates across multiple sentences to yield the final interpretation.

This visitor-based design cleanly separates parsing from interpretation: ANTLR4 enforces the grammar, the visitor constructs the AST, and the `interpret()` methods provide domain-specific semantics grounded in the Mixtec codices.

4 Illustrative Example

In this section, we will demonstrate the present contribution using the example in Figure 7. In particular, we will show how the parser and interpreter process an XML representing that scene into the following output:

> *“In Year 6 Flint Day 7 Eagle Lady 9 Eagle married Lord 6 Alligator”*

We begin with the XML document that the parser assumes from previous stages in the pipeline, show how it is lexed into a token stream, how it is parsed, and how it is finally interpreted into the above output.

4.1 Lexing Mixtec Scenes in XML Encoding

From the stages prior to *Parsing*, we begin with the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <scene>
    <year x="0" y="0" symbol="Flint" number="6"
          />
    <name_date x="0" y="0" symbol="Eagle"
        number="7" />
    <human x="0" y="0" gender="Lady" pose="
        sitting" orientation="right" />
    <name_date x="0" y="0" symbol="Eagle"
        number="9" />
    <object x="0" y="0" identity="house">
    <human x="0" y="0" gender="Lord" pose="
        sitting" orientation="left" />
    <name_date x="0" y="0" symbol="Alligator"
        number="6" />
  </scene>
</document>
```

The ANTLR parser combines the lexing and parsing required to convert this raw text in to an AST. In this section, we cover the former. Following the tokens shown in Figure 6, we define a document as an XML declaration (`<?xml ...>`) followed by a document opening tag (`<document>`). Within the document, a number of scenes (at least one) are nested. Each begins with scene opening tag (`<scene>`) and ends with the corresponding closing tag (`</scene>`). The document ends with the closing (`</document>`). Each of these lexical features corresponds in the obvious way to the tokens in the grammar, but they are enumerated explicitly in Table 1.

Token	**Feature**	**Regular Expression**
Y	Year	‘<year’ .*? ‘/>’
H	Human	‘<human’ .*? ‘/>’
ND	Name-Date	‘<name_date’ .*? ‘/>’
OBJ	Object	‘<obj’ .*? ‘/>’
NEAR_OBJ	Near Object	‘<near_obj’ .*? ‘/>’
DOCUMENT_OPEN	Meta	‘<document>’
DOCUMENT_CLOSE	Meta	‘</document>’
SCENE_OPEN	Meta	‘<scene>’
SCENE_CLOSE	Meta	‘</scene>’
XML_DECL	Meta	‘<?xml’ .*? ‘?>’

Table 1: Token classes, their corresponding features in the Mixtec input, and their regular expression as it appears in our ANTLR grammar file. “Meta” in the **Feature** column indicates that the token is metadata and represents an XML tag rather than input features.

Within the scenes, things are more interesting. The straightforward correspondence between tag names and token types is maintained, but information is encoded within tags that must be retained by the tokens after lexing. However, lexing is only concerned with the type of tokens and their proper order in its output: the token stream, so it merely records the tags within scenes as the corresponding token type (`H`, `ND`, `NEAR_OBJ`, `OBJ`, `Y`) and encoded

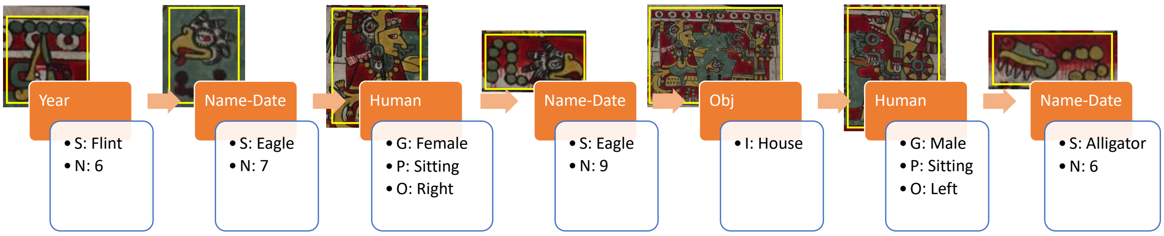

Figure 8: The token stream produced by lexing the XML received from the *Scanning* stage with insets of corresponding features in the example from Page 26 of the CZN (obverse) (Provided courtesy of the British Museum). Full example provided in Figure 7.

the information inside of the tags in these tokens' attributes. The result is shown in Figure 8. The tokens are ordered according to their appearance within the scene, and the scene is nested within `SCENE_OPEN` and `SCENE_CLOSE` tokens, which are nested inside the document tokens.

Astute readers may have noted two things about the regular expressions in Table 1 and the XML we assume from the pipeline's early steps in this section. We discuss these observations here.

First, the attributes populated in each of the XML tags corresponding to feature tokens are not governed by the regular expressions. In each case, there is a wild card inside the that says any input can be placed between the open and close of the XML tag. This policy may seem too loose, but it is actually an implementation decision taken with a view toward the parser's continued development as other steps of the interpretation pipeline emerge. Likewise, the second observation stems from taking advantage of the loose regular expressions. In each of the XML tags provided in our example, we have 'x' and 'y' coordinates of 0. These are left as 0, even though the XML parser and interpreter we contribute is not responsible for filling in these attributes, as an example of attributes that future parsers and interpreters will require of the XML. Future contributions at previous steps in the overall pipeline will decorate XML tags with these positional attributes. This exemplifies how permissive regular expressions in this contribution more readily catalyze such advancements in the project.

4.2 Parsing From Tokens

The stream of tokens produced in the previous subsection is the input for the parser. The grammar dictated in Figure 6 has been encoded into a grammar file for the ANTLR compiler compiler package complete with token definitions from Table 1. ANTLR generates a parser and lexer for the XML input. While lexing has been covered in depth, we now review how the parser produces an AST for the input through the recursive descent parsing algorithm.

From tokenization of the XML, according to the regular expressions given by Table 1, the parser implements recursive descent and creates an AST for valid inputs by expanding rules from the CFG (see Figure 6). For example, the rule for a `document` is expanded to get at the one or more `sent` sentence constructions within it. In turn, the parser decomposes each sentence into the appropriate form according to the corresponding `<scene>` tag's internal lexical features. In Figure 1, we give the AST our XML parser generates for the scene in Figure 7 as a first impression of our contribution.

4.3 Unhandled Features

Before our concluding remarks, we will review shortly some examples that our current parser and interpreter are fit to handle. Here, we will give several negative examples and explain briefly why handling them has been left to future work.

In Section 2, we first visited toponyms to describe their place in the human Mixtec interpretation literature. Since that point, we have left them out of the dicsussion, and analysis of the grammar and regular expressions we provide will show no token type(s) dedicated to them or place for these features in the grammar. The reasons for this follow from existing Mixtec literature, but we have left it for this section to explain.

Recall that the toponym in Figure 5 is read "Yucu Dzaa" which translates to the English "Hill of the

Bird" (Boone, 2000). We observe the head of a bird drawn behind an ornate pattern on the right side of a hill-shaped figure. Less distinct in this example is the human chin drawn through the bird's lower jaw. This feature is a phonetic component that specifies we should read the bird in this toponym as "Dzaa" (Boone, 2000).

Mixtec toponyms have been actively studied since the 1970s (Smith, 1973; Pohl, 1994; Boone, 2000). Despite this rich tradition, interpretation is very nuanced, as Figure 5's example shows. Interpretation can rely on phonetic features in the original language (Smith, 1973; Pohl, 1994; Boone, 2000) or even fall back on archaeology (Boone, 2000). Interpretations of most toponyms remained elusive even at the turn of this century (Boone, 2000). For this reason, toponyms are excluded from the present work, and a parser we detail in previous sections is not designed for them.

The number of objects in the CZN alone is large enough that enumerating, detecting, and classifying them could be its own project. We do not try to attempt it here; and the range of objects supported by the current implementation of *Parsing* and *Output* is quite limited in comparison to sum total of objects and near-objects that one could find in the source material.

Likewise, distinguishing between what is an object on its own, such as a house, and what is a near-object, like parts Lord 6 Alligator's headdress, has not yet been accomplished, either here or in the related work. While it may in that case seem paradoxical to include them in the present implementation, we have reason to include the `NEAR_OBJ` type anyway.

Despite the important setting details they impart, we cover above how toponyms are presently excluded because they cannot be well understood by the current method. It stands to reason we would exclude near-objects or incorporate them with the generic object tokens for the same reason. Instead, however, we understand that such features are not just pivotal to understanding the codices but also know they have a place within the sentences we form from scenes. Further, while no method currently exists, the problem does not seem as doubtful as toponym interpretation from a machine learning perspective. We therefore include near-objects in the grammar to anticipate future works that will enable us to distinguish them at earlier pipeline steps.

5 Future Work

Future work must expand the limited grammar, parser, and interpreter we propose before these tools are suitable for broad application. In particular, the coverage of sentence structures in the current grammar is narrow and should be extended to include additional motifs and constructions described in the Mixtec literature. While the present contribution and example serve as proof of concept, significant work remains before it can be applied to a variety of examples.

Another key direction is the development of systems that complete steps in the machine interpretation pipeline prior to *Parsing*. In particular, we anticipate systems that automatically segment glyphs in codex facsimiles and output the XML required by our regular expressions. The community must devise computer vision tools that are competent to segment scenes in the CZN into the various token classes we identify.

Equally important is extending classifier coverage beyond humans and name-dates. In particular, automated methods for detecting `OBJ` and `NEAR_OBJ` tokens are necessary for coherent scene-level interpretation.

Toponyms are, as we have already discussed, important sources of setting information for Mixtec interpretation. Without sound methods for deciphering toponyms, setting details in Mixtec Codices, important context for overall narratives, will remain out of reach even as the parser and interpreter support for other features expands. However, as we review previously, the correct interpretations of these complex, enigmatic glyphs are often in debate among Mixtec experts, which significantly complicates evaluation of automatic toponym recognition. Thus, it is difficult to say whether a machine interpretation of toponyms is possible given current understanding of those glyphs.

Limitations

The present grammar, corresponding parser, and the interpreter are quite limited in their scope, only covering a handful of Mixtec scene structures.

Related works have yet to produce any system that completes the steps of the machine interpretation pipeline up to *Scanning*, and these steps are necessary for the generation of XML encoded scenes that the present contribution is fit to interpret. The current work cannot serve any practical function without intensive human supervision. Without an automated method for producing XML, the significant hurdle of manually preprocessing any input into the XML encoding our system expects stands in the way of future progress and improvements to interpreting Mixtec Codices. Further, while we devise the machine interpretation pipeline as a framing device in which to place our contribution, its use presupposes that the early steps are feasible with present computer vision and natural language processing techniques. While we are confident in the ability of language-vision researchers, we make very few claims about how automated construction of the required XML may be affected.

In particular, there is currently no classifier to label object glyphs or to populate the **Identity** attribute of `OBJ` and `NEAR_OBJ` tokens, but these tokens are included in the grammar and handled by the machine parser and interpreter because object handling is necessary for any coherent interpretation of Mixtec codices. There is also not yet recognition systems to distinguish objects in the Mixtec Codices as `OBJ`s versus `NEAR_OBJ`s.

Curiously, our example in Figure 7 illustrates a final limitation. Citing nearby context, experts conclude that the scribe who authored the CZN (obverse) mistakenly renders the Lord's name as "6 Alligator" instead of "5 Alligator" (Williams, 2013). Detecting such errors is an important task but well beyond our present capabilities. The present contribution merely interprets XML representations of the Mixtec Codices as they are written. Indeed, it will be an interesting challenge for language-vision research to detect typographical errors such as the one in Page 26 of the CZN, and we look forward to future contributions in this direction.

Acknowledgments

The authors gratefully acknowledge Arnold and Lisa Goldberg, whose financial support helped to make this work possible.

References

Elizabeth Hill Boone. 2000. *Stories in red and black : pictorial histories of the Aztecs and Mixtecs / Elizabeth Hill Boone.*, 1st ed. edition. University of Texas Press, Austin.

B.E. Byland and J.M.D. Pohl. 1994. *In the Realm of 8 Deer: The Archaeology of the Mixtec Codices*. University of Oklahoma Press.

Alfonso Caso. 1977. *Reyes y reinos de la mixteca / Alfonso Caso.*, 1. ed. edition. Sección de obras de antropología. Fondo de Cultura Económica, México.

Martin de la Iglesia, Franziska Diehr, Uwe Sikora, Sven Gronemeyer, Maximilian Behnert-Brodhun, Christian Prager, and Nikolai Grube. 2021. The code of maya kings and queens: Encoding and markup of maya hieroglyphic writing. *Journal of the Text Encoding Initiative*.

Liudmila Fedorova, Antonio Perri, et al. 2024. Emblematic techniques as textual strategies in non-linear and linear scripts. *GRAPHOLINGUISTICS AND ITS APPLICATIONS*, 9:75–99.

Richard C. E. Long. 1926. The zouche codex. *The Journal of the Royal Anthropological Institute of Great Britain and Ireland*, 56:239–258.

Zelia Nuttall. 1902. *Codex Nuttall; facsimile of an ancient Mexican codex belonging to Lord Zouche of Harynworth, England*. Cambridge, Mass., Peabody Museum of American Archaeology and Ethnology, Harvard University.

Terence Parr. 2013. *The Definitive ANTLR 4 Reference*. Pragmatic Programmers.

Antonio Perri, Luciano Perondi, and Daniele Capo. 2024. Rethinking unicode: how to digitally encode non-linear written artefacts? a tentative encoding of codex mendoza, folio 2r. In *Proceedings of Grapholinguistics in the 21st Century, 2024*.

J.M.D. Pohl. 1994. *The Politics of Symbolism in the Mixtec Codices*. Vanderbilt University publications in anthropology. Vanderbilt University.

Girish Salunke, Christopher Driggers-Ellis, and Christan Grant. 2025. Classifying name-date and year figures in mixtec codices. In Taylor Arnold, Margherita Fantoli, and Ruben Ros, editors, *Computational Humanities Research 2025*, pages 1210–1218. Anthology of Computers and the Humanities.

M.E. Smith. 1973. *Picture Writing from Ancient Southern Mexico; Mixtec Place Signs and Maps*. Civilization of the American Indian series. University of Oklahoma Press.

Nancy P. Troike. 1978. Fundamental changes in the interpretations of the mixtec codices. *American Antiquity*, 43(4):553–568.

Alexander Webber, Zachary Sayers, Amy Wu, Elizabeth Thorner, Justin Witter, Gabriel Ayoubi, and Christan Grant. 2024. Analyzing finetuned vision models for Mixtec codex interpretation. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 42–49, Mexico City, Mexico. Association for Computational Linguistics.

Robert Lloyd Williams. 2013. *The complete codex Zouche-Nuttall : Mixtec lineage histories and political biographies / by Robert Lloyd Williams ; foreword* *by Rex Koontz.*, 1st ed. edition. Linda Schele series in Maya and pre-Columbian studies. University of Texas Press, Austin.