

MayaLEX: A Digital Etymological Dictionary for Early Mayan Languages

Kevin Barcenas-Martinez, Dr. Danny Law², Dr. Mallory Matsumoto³, Dr. Iyaxel Cojti Ren⁴, Dr. Todd B Krause⁵
¹Department of Linguistics, ²Linguistics Research Center, ³Department of Religious Studies, ⁴Department of Anthropology, The University of Texas

Introduction

- The Early Mesoamerican Languages Initiative (EMLI) at the Linguistics Research Center aims to increase access to and awareness of early indigenous languages of Mesoamerica.
- Early texts like the K'iche' Maya *Popol Vuh* and the *Florentine Codex* in Older Nahuatl provide invaluable cultural and historical information that would otherwise have been lost to colonial suppression.
- MayaLEX builds on existing lexicographic work, including Kaufman & Justeson's (2003) Preliminary Mayan Etymological Dictionary and specialized resources for individual languages such as Colonial Ch'olti' (Robertson, Law & Haertel, 2010).
- The MayaLEX project creates a comprehensive etymological dictionary resource to make these important languages and their historical relationships accessible to both scholars and the public.

Research Goal

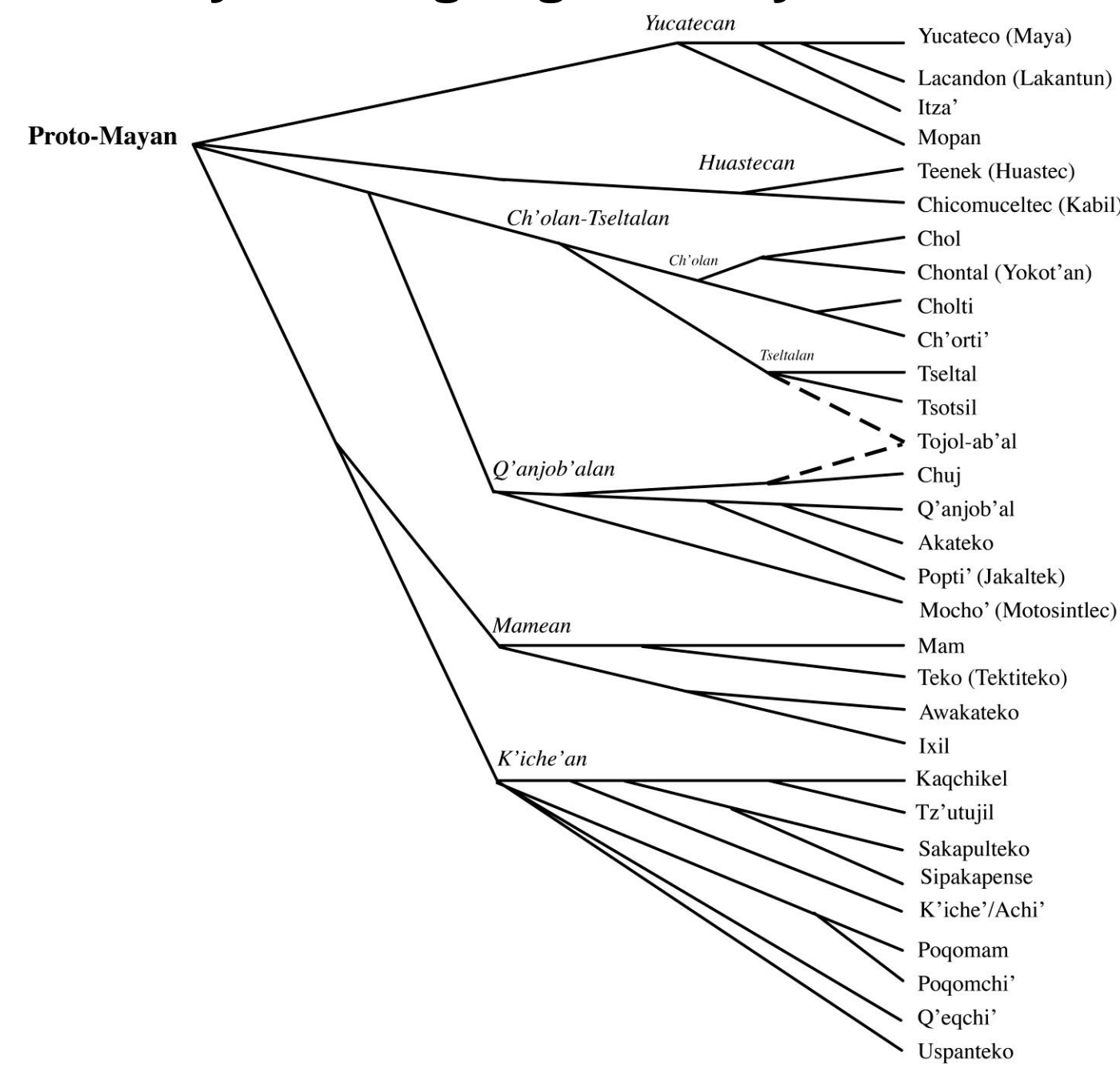
MayaLEX aims to create a comprehensive digital etymological resource that links words to their etymological ancestors in Proto-Mayan and enables users to trace words across languages through their historical connections.

Methods

- The project implements several key methodological approaches:
- Adapting the computational infrastructure of the LRC's Indo-European Lexicon (IELEX) to handle Mayan language data
 - Extracting lexical data from Colonial manuscripts and published dictionaries
 - Incorporating established Proto-Mayan reconstructions from Kaufman & Justeson (2003)
 - Creating a relational database that establishes links between related words across languages
 - Developing a user-friendly interface for exploring etymological connections

Figures and Results

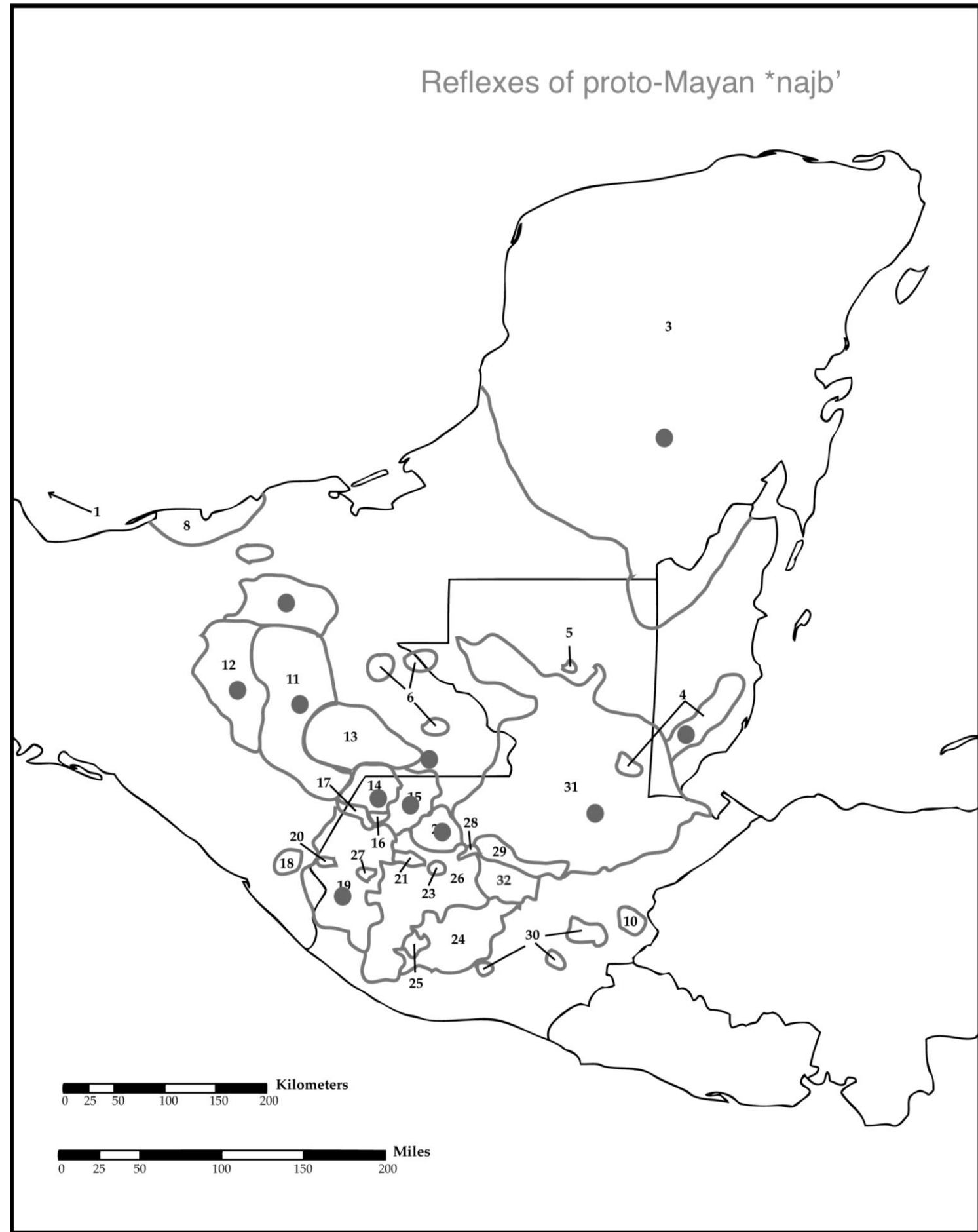
The Mayan Language Family



Case Study: Proto-Mayan *najib' 'pond'

Etymological Relationships:

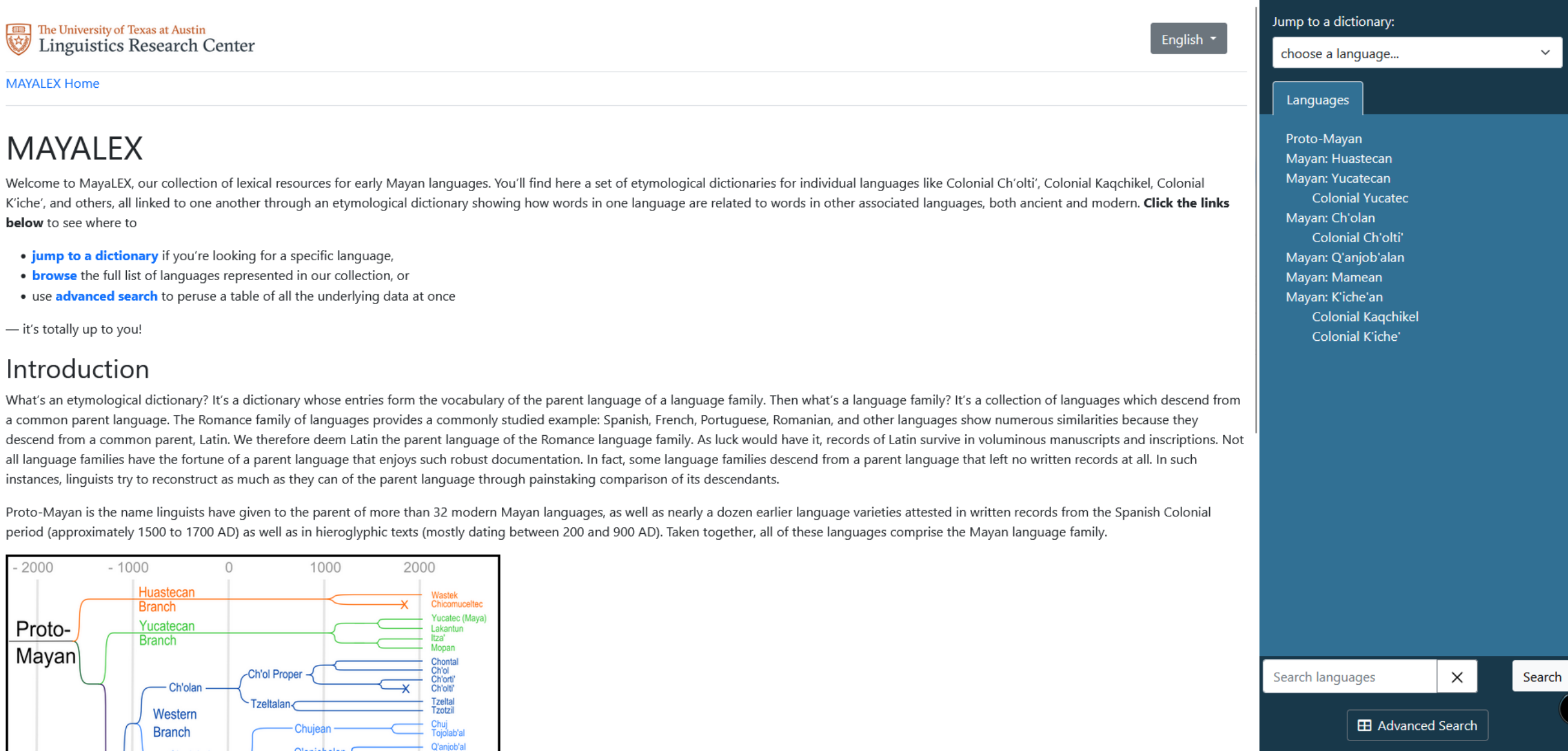
- Proto-Mayan *najib' 'pond'
→ Epigraphic Mayan <SEA-bi, na-bi> /nahb'/ 'pool, sea'
→ Proto-Ch'olan *nabh' 'sea'
- Ch'ol najb' 'lagoon'
 - Tsotsil nab' 'lagoon'
 - Tzeltal nabh'il 'lagoon's
- Q'anjob'al languages:
- Chuj nhajab' 'lagoon, well'
 - Q'anjob'al najab' 'lagoon, sea'
- Eastern Mayan:
- Mam najab' 'lagoon'
 - Ixil nab'a' 'lagoon'
- Yucatecan:
- Yucatec k'a'nab' 'lagoon'
 - Mopan k'ak'nab' 'lagoon, sea'
 - Q'eqchi' kaq nab' 'Lagoon, Sea'



MayaLEX Digital Interface

The MayaLEX system allows users to:

- Start with a word in any of the included languages (e.g., Ch'ol najb' 'lagoon')
- See its Proto-Mayan etymon (*najib' 'pond')
- Explore all related forms in other Mayan languages



Computational Linguistics Approach

1. Data Extraction & Transformation

Converting Colonial texts and dictionaries into structured data and excel files

```
def parse_input(filename):  
    with open(filename, 'r') as file:  
        lines = file.readlines()  
    parsed_data = []  
    for line in lines:  
        parsed_data.append()  
    return parsed_data
```

2. Semantic Analysis with NLP

Using spaCy word vectors to automatically classify dictionary entries

```
import spacy  
import pandas as pd  
# Load Kaufman's etymological data  
df = pd.read_excel("Kaufman_Database.xlsx")  
# For entries missing semantic tags:  
for index, row in df.iterrows():  
    eng_def = row['Meaning (English)']  
    eng_vector = nlp(eng_def)  
    # Find most similar semantic domain  
    # Assign if above threshold  
    if max_similarity >= SIMILARITY_THRESHOLD:  
        df.at[index, "Semantic Tag"] = best_tag
```

3. Advanced Language Tree Visualization

Generating comprehensive Mayan language family trees with Graphviz

```
import pandas as pd  
import graphviz  
from collections import defaultdict  
# Load language data  
df = pd.read_excel("Kaufman_Lects.xlsx",  
                  sheet_name="Relational Database")  
# Initialize visualization  
# Add terminal nodes  
for index, row in df.iterrows():  
    if row['Language/Terminal Node'] == 'yes':  
        dot.node(row['ID'], label=row['Language'])  
# Add branch relationships  
for index, row in df.iterrows():  
    if pd.notna(row['Parent ID']):  
        parent = row['Parent ID'].strip()  
        child = row['ID']  
# Render the visualization  
dot.render("mayan_tree", view=True)
```

MayaLEX Computational Pipeline

- Automated semantic tagging using vector similarity (spaCy)
- Extraction of structured data from unstructured Colonial texts/primary sources
- Programmatic visualization of language family relationships
- Integration with IELEX computational infrastructure