Practical Python

7: Basic machine Learning with NLTK

Dafydd Gibbon

First South African Workshop in Digital Humanities
North-Western University, Potchefstroom, SA
2015-04-04 to 2015-04-05

ML with NLTK

- ML types:
  - Naive Bayes
  - Maximum Entropy / Logistic Regression
  - Decision Tree
  - SVM (?)

- NLTK-Trainer

  `https://github.com/japerk/nltk-trainer`
  - command line scripts
  - train custom models
  - analyse corpora
  - analyse models against corpora

Machine Learning – supervised learning

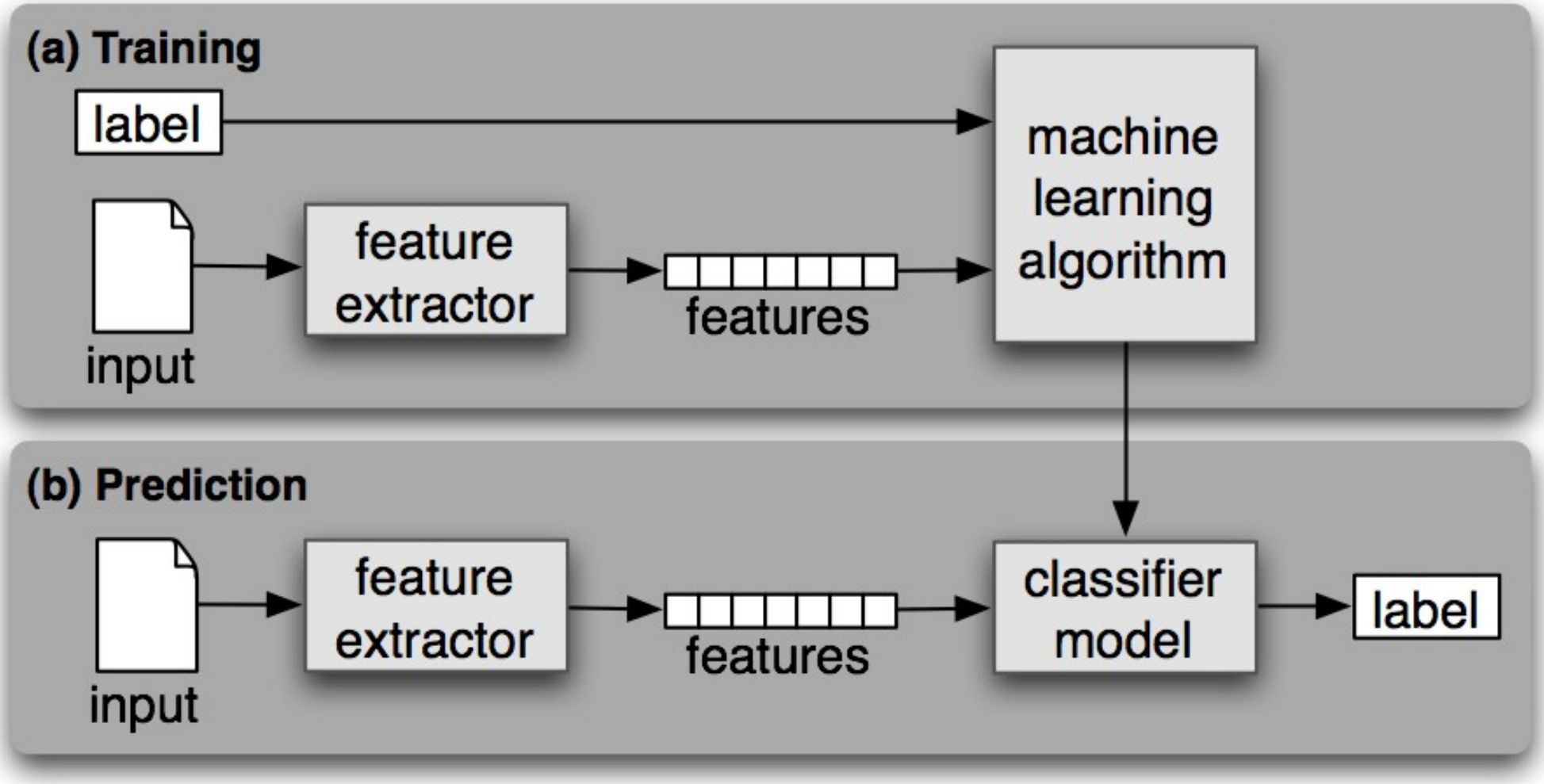


Figure from NLTK book.

A textbot with NLTK (unsupervised learning) (and its limitations):

```
import nltk


def generate_model(cfdist, word, num=15):
    for i in range(num):
        print word,
        word = cfdist[word].max()
# Get data
text = nltk.corpus.genesis.words('english-kjv.txt')
# Define probabilities between neighbouring words:
bigrams = nltk.bigrams(text)
cfd = nltk.ConditionalFreqDist(bi
# Predict
generate_model(cfd,'living')
```

Suggestion:
Solve the circularity problem of the textbot.

Simple gender-feature name classifier

```
#!/home/gibbon/miniconda2/bin/python
# Simple gender classifier from NLTK book, Ch. 6 (edited)

import nltk
from nltk.corpus import names
import random

# Define a (kind of) feature extractor
def gender_features(word):
    return {'last_letter': word[-1]}

# Get data
male = [(name, 'male') for name in names.words('male.txt')]
female = [(name, 'female') for name in names.words('female.txt')]
labeled_names = male + female
random.shuffle(labeled_names)

# Get feature sets, partition into training and test sets
featuresets = [(gender_features(n), gender) for (n, gender) in
labeled_names]
train_set = featuresets[500:]
test_set = featuresets[:500]
```

Simple gender-feature name classifier

```
# Training
classifier = nltk.NaiveBayesClassifier.train(train_set)

# Testing
accuracy = nltk.classify.accuracy(classifier, test_set)

# Results
print "train:test set sizes:",len(train_set),':',len(test_set)
print "Accuracy with", len(test_set), "items:", accuracy
print "Most Informative Features: (likelihood ratios per
feature):"
classifier.show_most_informative_features(5)

# Predicting for single items
testlist = ['Robert', 'Anna', 'Neo', 'Trinity']
maxlen = max([len(x) for x in testlist])
for x in testlist:
   gender = classifier.classify(gender_features(x))
   print ('"'+x+'":').ljust(maxlen+4), gender
```

Learning distances between languages

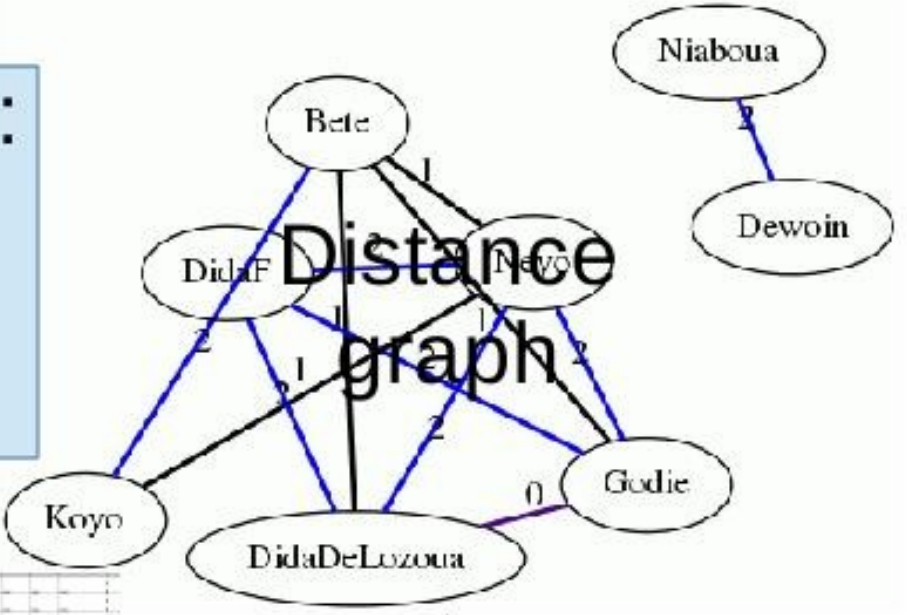
- The DistGraph web site describes and demonstrates the discovery of similarities and differences (interpreted as 'distances') between languages of the Ivory Coast:
  - http://wwwhomes.uni-bielefeld.de/gibbon/DistGraph/
- The data are consonant systems of Ivory Coast languages, harvested from atlases of Ivory Coast languages dating back to fieldwork done in the late 1970s and early 1980s.
- Consonant systems are chosen
  - because of their relative ease of idenfication in comparison to other phonological properties such as vowels or tones,
  - because analysis of lexical similarities (characteristic of many similar studies) presupposes phonological analysis as the main basis of lexical similarities.

A complex Python web application:

Learning distances between languages

Learning distances between languages: workflow

Preprocessing:
1. scan/type
2. correct
3. create CSV



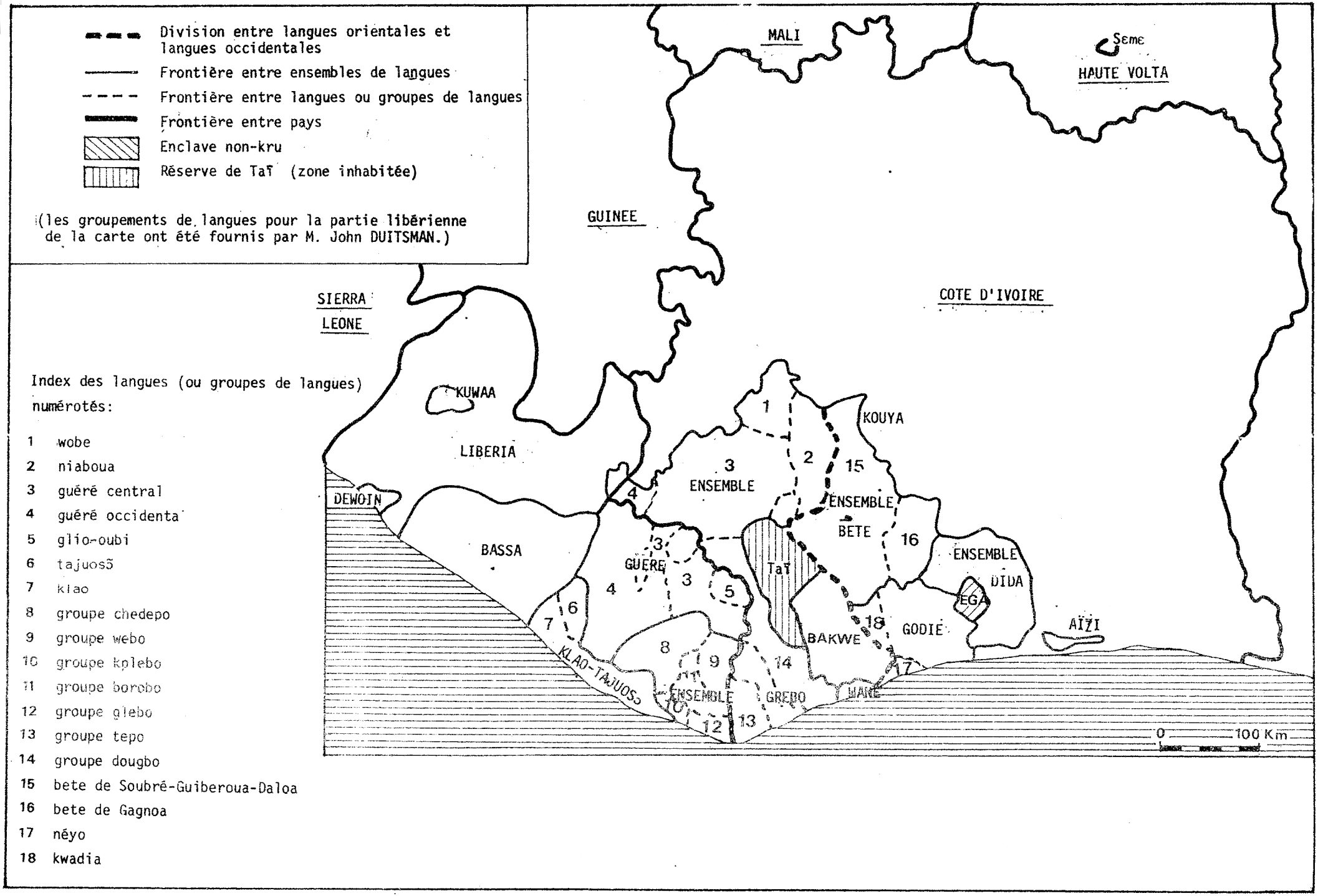
Data input: CSV

GraphViz

Levenshtein Edit Distance comparison

Distance matrix

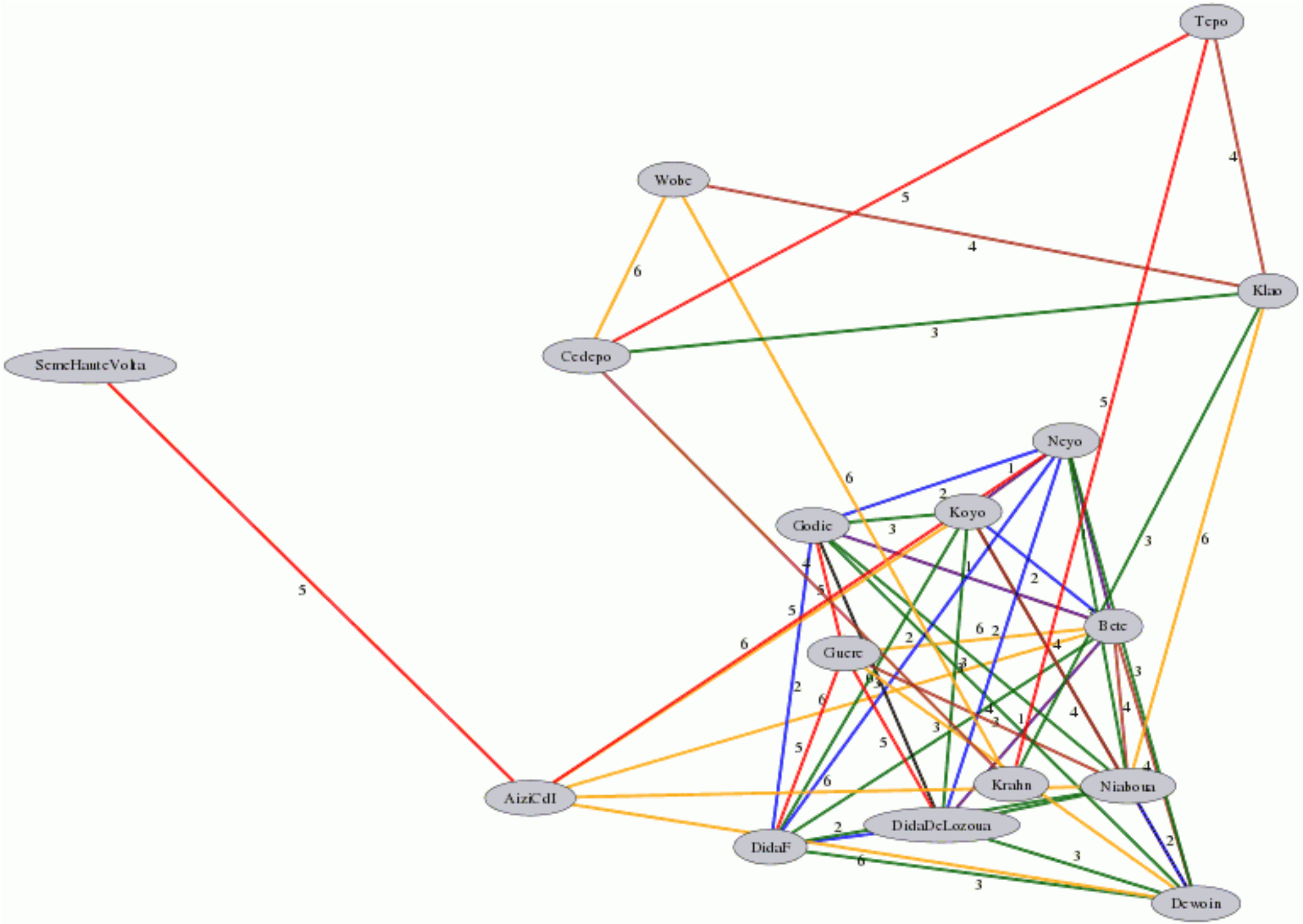
Learning distances between languages: hand-drawn language map



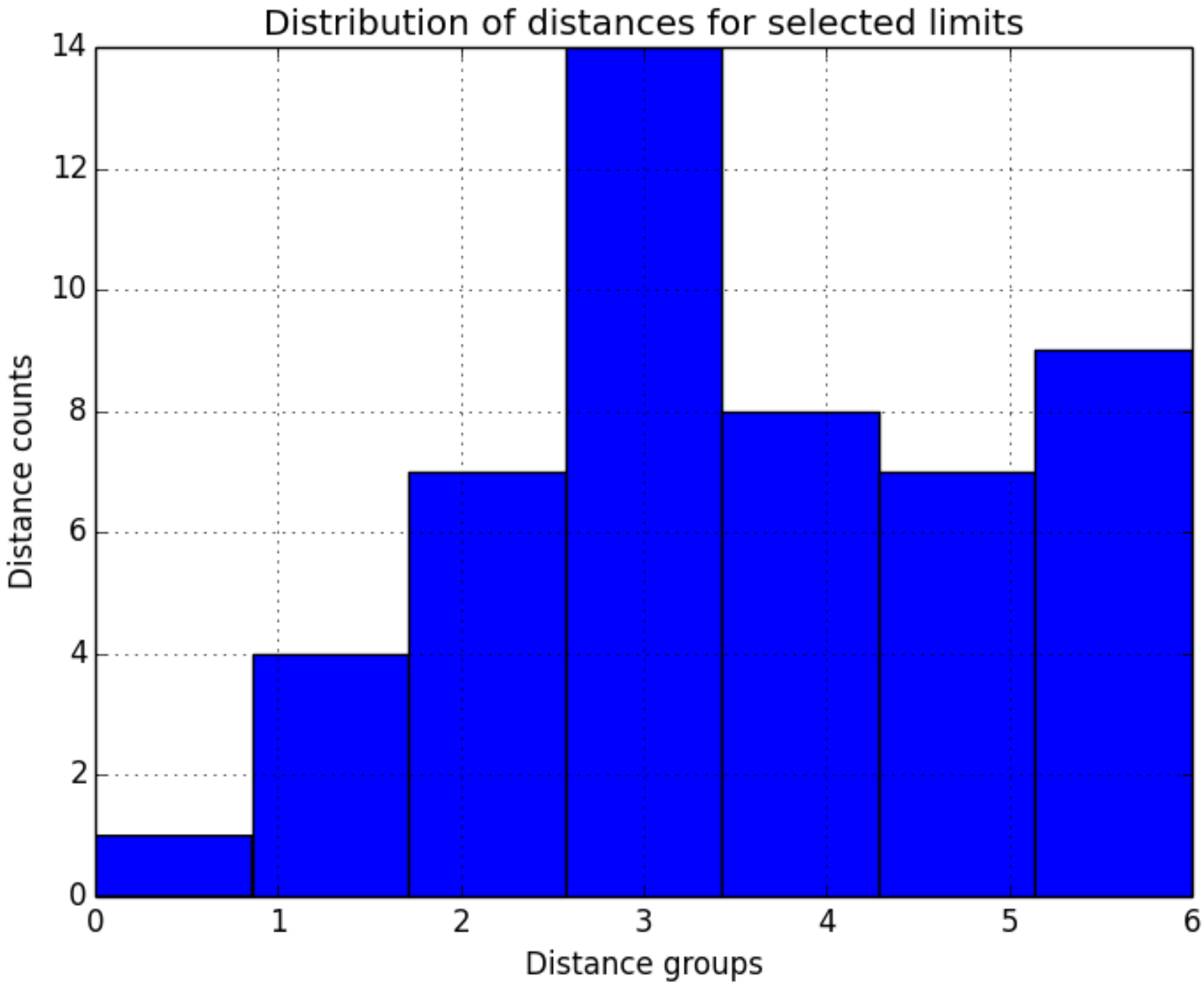
Learning distances between languages: Kru consonant data

```
Bete;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;_;f;s;_;v;z;_;_;_;B;_;l
Godie;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;gw;f;s;_;v;z;_;_;_;B;_
Koyo;p;t;c;k;kp;kw;kj;b;d;C;_;g;gb;_;f;s;_;v;z;_;_;_;B;_;
Neyo;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;_;f;s;_;v;z;_;_;_;B;_;l
DidaDeLozoua;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;gw;f;s;_;v;z;_;
DidaF;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;gw;f;s;_;v;z;_;_;_;B;_
Wobe;p;t;c;k;kp;kw;_;b;d;C;_;_;gb;_;f;s;_;_;_;_;_;_;_;_;_
Guere;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;gw;f;s;_;v;z;_;_;_;B;D
Krahn;p;t;c;k;_;kw;_;b;d;C;_;_;gb;_;f;s;_;_;_;_;_;_;_;_;l
Cedepo;p;t;c;k;kp;kw;_;b;d;C;_;_;gb;_;f;s;_;_;_;_;h;_;_;_
Klao;p;t;c;k;kp;kw;_;b;d;C;_;_;gb;_;f;s;_;_;_;_;_;_;_;_;l
Niaboua;p;t;c;k;kp;kw;_;b;d;C;_;g;gb;gw;f;s;_;v;z;_;_;_;B
Dewoin;p;t;_;k;kp;kw;_;b;d;C;_;g;gb;gw;f;s;_;v;z;_;_;_;B;
Bassa;p;t;c;k;kp;_;_;b;d;C;dj;g;gb;_;f;s;_;v;z;_;h;hw;B;_
Grebo;p;t;c;k;kp;_;_;b;d;C;_;g;gb;_;f;s;_;_;_;_;h;hw;_;_;
Tepo;p;t;c;k;_;kw;_;b;d;C;_;g;gb;_;f;s;_;_;_;_;h;_;_;_;l;
KuwaaLiberia;p;t;_;k;kp;kw;_;b;d;C;_;_;_;_;f;s;_;_;_;_;_;
SemeHauteVolta;p;t;c;k;kp;_;_;b;d;C;_;g;gb;_;f;s;S;v;_;_;
AiziCdI;p;t;c;k;kp;_;_;b;d;C;_;g;gb;_;f;s;S;v;z;Z;_;_;_;_
```

Learning distances between languages: Hamming Distances



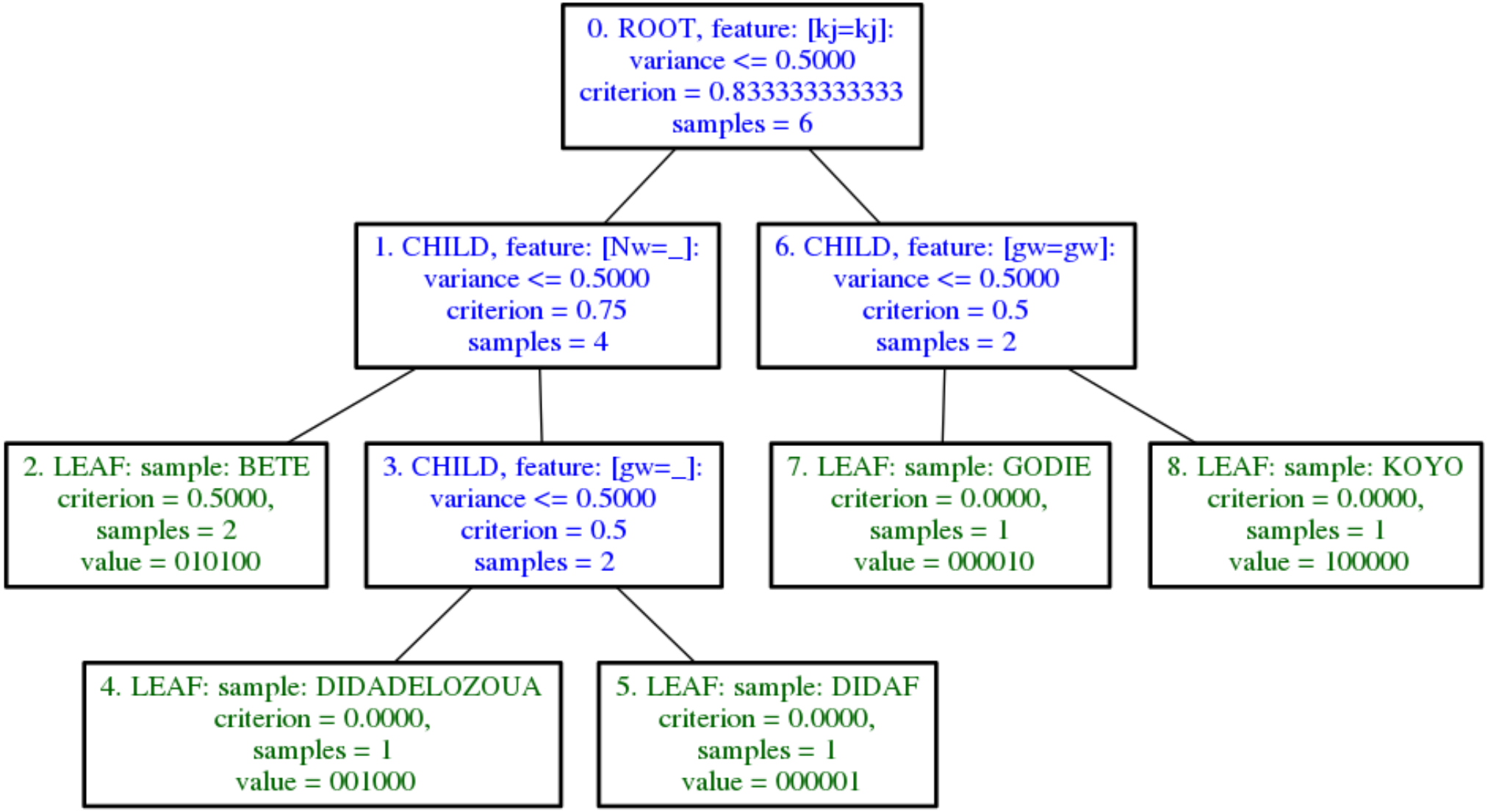
Learning distances between languages: distance histogram



Learning distances between languages: language classification

- Machine Learning:
  - First determine the relative importance of features
  - Then apply a classifier to group the languages in terms of selected features
- Work in progress … :)

Induction of Decision Trees



Learning distances between languages: Python implementation

- Resource: Legacy Ivory Coast Language Atlas
- User interaction via:
  - HTML input form
    - parameter settings
    - CSV table pasting
  - Common Gateway Interface (CGI)
  - Server-side Python application
  - HTML output:
    - Text, Statistics
    - Images linked to graph
- User can download graphs and copy text and statistics

- Processing:
  - CSV transformation to Python list of lists
  - Pairwise application of Levenshtein Edit Distance or Hamming Distance
  - Construction of triangular distance matrix
  - Transformation of matrix into distance triples
  - Conversion of distance triples into GraphViz network code
  - System call to GraphViz dot application for graph generation
  - Graph storage

End of Unit 7