A Probabilistic Model for Ancient Egyptian Hieroglyph

Fahrurrozi Rahman
Supervised by: Mark-Jan Nederhof
There are, at least, three main classes for classifying the signs:

1. Phonogram, pronounces the signs without having any semantic meaning;
   ![Phonogram]
   “pr”

2. Logogram, depicts the meaning of the signs;
   ![Logogram]
   “pr”

3. Determinative, clarifies the meaning of the phonograms it follows;
   ![Determinative]
   “hnw”

The same word can be written in several ways;
   ![Variations] or ![Variations]  “snw” or “brothers”
Aligning between the signs and the transliteration
Knowing the **functions** of the sign in the word

therefore

Need to have the **probability distributions** of the functions from some available data
What we need:

Data to learn

What we have:

2 annotated corpora

- Do learn from the data
- Learn better when more data become available
Finite State Automaton: configurations as states and functions as transition labels.
Strategy

- **Supervised**: learn from the *correct sequence of functions* for the signs and the transliteration in the annotated corpora
- Suitable when there is abundant amount of annotated data

1. **NGram**
   \[
P(f_i | f_{i-N+1}^{i-1}) = \frac{C(f_{i-N+1}^{i-1})}{C(f_{i-N+1}^{i-1})}
   \]
   to estimate the probability of having a function following previous N-1 functions

2. **Hidden Markov Model (HMM)**
   \[
P(f_1 | f_{i-N+1}^{i-1}) \approx P(c_i | c_{i-N+1}^{i-1}) \times P(f_i | c_i)
   \]
   to estimate the probability of a function based on the NGram of the function's class (e.g. logogram, phonogram, determinative, etc.)

3. **Simple Good-Turing** smoothing and
4. **Katz's** back-off for **flexibility** and **robustness**
Strategy

- **Unsupervised**: learn from the *possible sequence of functions* in the automaton to know if learning from unannotated data is possible to build the probabilistic model
- It is suitable when there is only small amount of available annotated data

**Forward-Backward** technique:
- set uniform initial probability for all functions
- walk along the paths in the automata
- compute the forward and backward value for every state
- reestimate the probability for each function using the forward and backward value
- do until convergence

Collaborate with NGram but **ad-hoc penalty** is used instead of smoothing
Evaluation

F1 score for supervised learning

- Ship. Sail (train) vs. P. Westcar (test)
- P. Westcar (train) vs. Ship. Sail (test)

F1 score for unsupervised learning

- P. Westcar (train) vs. Ship. Sail (test)
- 1st half P. Westcar (train) vs. 2nd half P. Westcar (test)
- 2nd half P. Westcar (train) vs. 1st half P. Westcar (test)
Conclusion

● Both techniques, supervised and unsupervised, perform better with larger training material.
● The unsupervised learning suffers from the absence of smoothing technique which is quite helpful when predicting the correct function in the case it has never been seen before.
● The supervised learning can be used to build the language model if we have enough training material whereas a good smoothing technique is needed to improve the performance of the unsupervised technique.

Future Improvements

● Exploiting the smoothing technique for the unsupervised approach.
● Combining the supervised learning with the unsupervised one.
● Using neural network based language models.