Multiple Word Alignment with Profile Hidden Markov Models

Aditya Bhargava and Grzegorz Kondrak
Department of Computing Science
University of Alberta
{abhargava,kondrak}@cs.ualberta.ca
Multiple word alignment

- Given multiple words, align them all to each other
- Our approach: Profile HMMs, used in biological sequence analysis
- Use match, insert, and delete states to model changes
- Evaluate on cognate set matching
  - Beat baselines of average and minimum edit distance
What you can expect

- Introduction: word alignment
- Profile hidden Markov models
  - For bioinformatics
  - For words?
- Experiments
- Conclusions & future work
Introduction

• Multiple word alignment:
  ▫ Take a set of words
  ▫ Generate some alignment of these words
  ▫ Similar and equivalent characters should be aligned together

• Pairwise alignment gets us:
  ▫ String similarity and word distances
  ▫ Cognate identification
  ▫ Comparative reconstruction
Introduction

• Extending to multiple words gets us:
  ▫ String similarity with multiple words
  ▫ Better-informed cognate identification
  ▫ Better-informed comparative reconstruction

• We propose Profile HMMs for multiple alignment
  ▫ Test on cognate set matching
Profile hidden Markov models
Profile hidden Markov models

- Match states are “defaults”
- Insert states are used to represent insert symbols
- Delete states are used to represent the absence of symbols
Profile hidden Markov models

\[
\text{MMIIIM}
\]

AG...C
A-AG.C
AG.AA-
--AAAC
AG...C

• In this sample DNA alignment, dashes represent deletes and periods represent skipped inserts
Profile hidden Markov models

**MMIIIM**

AG...C
A−AG.C
AG.AA−
--AAAC
AG...C
Profile hidden Markov models

**MMIIIM**

AG...C

A–AG.C

AG.AA–

|--AAAC

AG...C

---

D1

D2

D3

I0

I1

I2

I3

Begin

M1

M2

M3

End
Profile hidden Markov models

\[ \text{MMIIIM} \]
\[ \text{AG\ldotsC} \]
\[ \text{A\textcolor{blue}{-AG.C}} \]
\[ \text{AG.AA\textcolor{blue}{--AAAC}} \]
\[ \text{AG\ldotsC} \]
Profile hidden Markov models

\[
\text{MMIIIM} \\
\text{AG...C} \\
\text{A-AG\_C} \\
\text{AG\_AA-} \\
\text{---AAAC} \\
\text{AG...C}
\]
Profile hidden Markov models

\[
\begin{align*}
&\text{MMIIIM} \\
&\text{AG...C} \\
&A\mathbf{-AG.C} \\
&\text{AG.AA--} \\
&\text{--AAAC} \\
&\text{AG...C}
\end{align*}
\]
Profile hidden Markov models

\[
\text{MMIIIIM}
\]

\[
\text{AG...C}
\]

\[
\text{A–AG.C}
\]

\[
\text{AG.AA–}
\]

\[
\text{--AAAC}
\]

\[
\text{AG...C}
\]
Profile hidden Markov models

\[\text{MMIIIM} \]
\[\text{AG...C} \]
\[\text{A-AG.C} \]
\[\text{AG.AA-} \]
\[\text{--AAAC} \]
\[\text{AG...C} \]
Profile hidden Markov models

• To construct a Profile HMM from aligned sequences:
  ▫ Determine which columns are match columns and which are insert columns, then estimate transition and emission probabilities directly from counts
• To construct a Profile HMM from unaligned sequences:
  ▫ Choose a model length, initialize the model, then train it to the sequences using Baum-Welch
Profile hidden Markov models

• Evaluating a sequence for membership in a family
  ▫ Use the forward algorithm to get the probability
  ▫ Use Viterbi to align the sequence

• Multiple alignment of unaligned sequences
  ▫ Construct & train a Profile HMM
  ▫ Use Viterbi to align the sequences
Profile hidden Markov models

- Profile HMMs are generalizations of Pair HMMs
  - Word similarity and cognate identification
- Unlike Pair HMMs, Profile HMMs are position-specific
  - Each model is constructed from a specific family of sequences
  - Pair HMMs are trained over many pairs of words
Profile HMMs for words

- Words are also sequences!
- Similar to their use for biological sequences, we apply Profile HMMs to multiple word alignment
- We also test Profile HMMs on matching words to cognate sets
- We made our own implementation and investigated several parameters
Profile HMMs: parameters

- Favour match states?
- Pseudocount methods
  - Constant-value, background frequency, substitution matrix
- Pseudocount weight
- Pseudocounts added during Baum-Welch
Experiments: Data

- Comparative Indoeuropean Data Corpus
  - Cognition data for words in 95 languages corresponding to 200 meanings
- Each meaning reorganized into disjoint cognate sets
Experiments: Multiple cognate alignment

- Parameters determined from cognate set matching experiments (later)
- Pseudocount weight set to 100 to bias the model using a substitution matrix
- Highly-conserved columns are aligned correctly
- Similar-sounding characters are aligned also correctly, thanks to the substitution matrix method
- Insert columns should not be considered aligned
- Problems with multi-character phonemes
  - An expected problem when using the English alphabet instead of e.g. IPA
Experiments: Cognate set matching

• How can we evaluate the alignments in a principled way? There is no gold standard!
• We emulate the biological sequence analysis task of matching a sequence to a family; we match a word to a cognate set
• The task is to correctly identify the cognate set to which a word belongs given a number of cognate sets having the same meaning as the word; we choose the model yielding the highest score
Experiments: Cognate set matching

- Development set of 10 meanings (~5% of the data)
- Substitution matrix derived from Pair HMM method
- Best parameters:
  - Favour match states
  - Use substitution matrix pseudocount
  - Use 0.5 for pseudocount weight
  - Add pseudocounts during Baum-Welch
Experiments: Cognate set matching

Average Edit Distance: 77.0%
Minimum Edit Distance: 91.0%
Profile HMM: 93.2%
Experiments: Cognate set matching

- Accuracy better than both average and minimum edit distance
- Why so close to MED?
  - Many sets had duplicate words (same orthographic representation for different languages)
Conclusions

- Profile HMMs can work for word-related tasks
- Multiple alignments are reasonable
- Cognate set matching performance exceeds minimum and average edit distance
- If multiple words need to be considered, Profile HMMs present a viable method
Future work

- Better model construction from aligned sequences
- Better initial models for unaligned sequences
- Better pseudocount methods
- N-gram output symbols