CMPT 413
Computational Linguistics

Anoop Sarkar

http://www.cs.sfu.ca/~anoop
Minimum Cost Edit Distance

• String edit distance: what is the minimum number of changes (char insertions or deletions) to transform the string *intention* into *execution*?

• Assume cost of insertion is 1 and cost of deletion is 1

• Note that we assume that we can only change one character at a time
Levenshtein Distance

• Cost is fixed across characters
  – Insertion cost is 1
  – Deletion cost is 1

• Two different costs for substitutions
  – Substitution cost is 1 (transformation)
  – Substitution cost is 2 (one deletion + one insertion)
Minimum Cost Edit Distance

• Think of it as an alignment between target and source

\[ D(i, j) = \min \begin{cases} 
D(i-1, j) + \text{cost}(t_i, \emptyset) \text{ insertion into target} \\
D(i-1, j-1) + \text{cost}(t_i, s_j) \text{ substitution/identity} \\
D(i, j-1) + \text{cost}(\emptyset, s_j) \text{ deletion from source} 
\end{cases} \]

\[ D(0,0) = 0 \]

Find \( D(n,m) \) recursively

\[ D(i,0) = D(i-1,0) + \text{cost}(t_i, \emptyset) \]

\[ D(0,j) = D(0, j-1) + \text{cost}(\emptyset, s_j) \]
Function MinEditDistance (target, source)

n = length(target)
m = length(source)
Create matrix D of size (n+1,m+1)
D[0,0] = 0

for i = 1 to n
    D[i,0] = D[i-1,0] + insert-cost

for j = 1 to m
    D[0,j] = D[0,j-1] + delete-cost

for i = 1 to n
    for j = 1 to m
        D[i,j] = MIN(D[i-1,j] + insert-cost,
                    D[i-1,j-1] + subst/eq-cost,
                    D[i,j-1] + delete-cost)

return D[n,m]
<table>
<thead>
<tr>
<th>source</th>
<th>g</th>
<th>a</th>
<th>m</th>
<th>b</th>
<th>l</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>u</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>m</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>o</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

The target sequence is read from the left to the right, starting from the 0 in the source row and following the arrows.
Edit Distance and FSTs

• Algorithm using a Finite-state transducer:
  – construct a finite-state transducer with all possible ways to transduce intention (source = input) into execution (target = output)
  – We do this transduction one char at a time
  – A transition x:x gets zero cost and a transition on ε:x (insertion) or x:ε (deletion) for any char x gets cost 1
  – Finding minimum cost edit distance == Finding the shortest path from start state to final state
Edit distance and FSTs

0 stands for empty string, $x:0$ means delete $x$, $0:x$ means insert $x$.
Edit distance and FSTs
Edit distance

• Useful in many NLP applications
• In some cases, we need edits with multiple characters, e.g. 2 chars deleted for one cost
• Comparing system output with human output, e.g. input: ibm output: IBM vs. Ibm (TrueCasing of speech recognition output)
• Error correction
• Defined over character edits or word edits, e.g. MT evaluation:
  – Foreign investment in Jiangsu ‘s agriculture on the increase
  – Foreign investment in Jiangsu agricultural investment increased
Pronunciation dialect map of the Netherlands based on phonetic edit-distance (W. Heeringa Phd thesis, 2004)
Variable Cost Edit Distance

• So far, we have seen edit distance with uniform insert/delete cost
• In different applications, we might want different insert/delete costs for different items
• For example, consider the simple application of spelling correction
• Users typing on a qwerty keyboard will make certain errors more frequently than others
• So we can consider insert/delete costs in terms of a probability that a certain alignment occurs between the correct word and the typo word
Spelling Correction

• Types of spelling correction
  – non-word error detection
    e.g. hte for the
  – isolated word error detection
    e.g. acres vs. access (cannot decide if it is the right word for the context)
  – context-dependent error detection (real world errors)
    e.g. she is a talented acres vs. she is a talented actress

• For simplicity, we will consider the case with exactly 1 error
Noisy Channel Model

Source

original input

Noisy Channel

noisy observation

Decoder

\[ P(\text{original input} \mid \text{noisy obs}) \]
Bayes Rule: \textit{computing } P(\text{orig} \mid \text{noisy}) \\

\begin{itemize}
\item let \( x = \text{original input} \), \( y = \text{noisy observation} \)
\end{itemize}

\[ p(x \mid y) = \frac{p(x, y)}{p(y)} \quad p(y \mid x) = \frac{p(y, x)}{p(x)} \]

\[ p(x, y) = p(y, x) \]

\[ p(x \mid y) \times p(y) = p(y \mid x) \times p(x) \]

\[ p(x \mid y) = \frac{p(y \mid x) \times p(x)}{p(y)} \quad \text{Bayes Rule} \]
Chain Rule

\[ p(a,b,c \mid d) = p(a \mid b,c,d) \times p(b \mid c,d) \times p(c \mid d) \]

Approximations: Bias vs. Variance

\[ p(a \mid b,c,d) \approx p(a \mid b,c) \text{ less bias} \]
\[ p(a \mid b) \]
\[ p(a) \text{ less variance} \]
Single Error Spelling Correction

• Insertion (addition)
  – across vs. cress

• Deletion
  – across vs. actress

• Substitution
  – across vs. access

• Transposition (reversal)
  – across vs. caress
Noisy Channel Model for Spelling Correction  
(Kernighan, Church and Gale, 1990)

- \( t \) is the word with a single typo and \( c \) is the correct word
  \[
P(c \mid t) = p(t \mid c) \times p(c)
  \]

- Find the best candidate for the correct word
  \[
  \hat{c} = \arg \max_{c \in C} P(t \mid c) \times P(c)
  \]
  \[
  P(t \mid c) = ??
  \]
  \[
  P(c) = \frac{f(c)}{N}
  \]

\( C \) is all the words in the vocabulary; \(|C| = N\)
Noisy Channel Model for Spelling Correction
(Kernighan, Church and Gale, 1990)
single error, condition on previous letter

\[ P(t \mid c) = \begin{cases} 
\frac{\text{del}[c_{p-1}, c_p]}{\text{chars}[c_{p-1}, c_p]} (xy)_c \text{ typed as } (x)_t \\
\frac{\text{ins}[c_{p-1}, t_p]}{\text{chars}[c_{p-1}]} (x)_c \text{ typed as } (xy)_t \\
\frac{\text{sub}[t_p, c_p]}{\text{chars}[c_p]} (y)_c \text{ typed as } (x)_t \\
\frac{\text{rev}[c_p, c_{p+1}]}{\text{chars}[c_p, c_{p+1}]} (xy)_c \text{ typed as } (yx)_t 
\end{cases} \]

\[ t = \text{poton} \]
\[ c = \text{potion} \]
\[ \text{del}[t,i]=427 \]
\[ \text{chars}[t,i]=575 \]
\[ P = .7426 \]

\[ t = \text{poton} \]
\[ c = \text{piton} \]
\[ \text{sub}[o,i]=568 \]
\[ \text{chars}[i]=1406 \]
\[ P = .4039 \]
Noisy Channel model for Spelling Correction

- The $del$, $ins$, $sub$, $rev$ matrix values need data in which contain known errors (training data)
- Accuracy on single errors on unseen data (test data)
Noisy Channel model for Spelling Correction

• Easily extended to multiple spelling errors in a word using edit distance algorithm (however, using learned costs for ins, del, replace)

• Experiments: 87% accuracy for machine vs. 98% average human accuracy

• What are the limitations of this model?

… was called a “stellar and versatile across whose combination of sass and glamour has defined her …

KCG model best guess is acres