NLP - Lecture 01 - Handout

Minimum Edit Distance (MED)

Given String X and Y, three operations allowed for editing string X or Y:

- Insertion (Ins)
- Deletion (Del)
- Substitution (Sub)

Example $X = \text{INTENTION}$

$Y = \text{EXECUTION}$

Let $O = [o_1\, o_2 \cdots o_k]$ be k operations that we can take X to Y, where each $o_k \in \{\text{Ins, Del, Sub}\}$.

$$X \xrightarrow{\text{Del: I}} \text{NTENTION} \xrightarrow{\text{Sub: N}\to\text{E}} \text{ETENTION}$$

$$\xrightarrow{\text{Sub: T}\to\text{X}} \text{EXENTION} \xrightarrow{\text{Ins: C}} \text{EXECNTION}$$

$$\xrightarrow{\text{Sub: N}\to\text{U}} \text{EXECUTION}$$

o_1: Del, o_2: Sub, o_3: Sub, o_4: Ins, o_5: Sub. $k = 5$.

If Ins, Del cost 1 and Sub costs 2.

Then. this alignment costs 8.

We call $O = [O_1 O_2 \cdots O_k]$ that takes X to Y an alignment.

Q: How to find the optimal alignment?

(By optimal, we mean the cost of O is minimum). Let us define:

$X = x_1 x_2 \cdots x_n$, i indexing ith char of X

$Y = y_1 y_2 \cdots y_m$, j indexing j-th char of Y.

Naive method: See slide P77.

An alternative way, consider a smaller problem (subproblem).

$X_i = [x_1, x_2, \cdots x_i]$, $Y_j = [y_1, y_2, \cdots, y_j]$.

Let $X_i = [x_1, x_2, \cdots, x_i]$, $Y_i = [y_1, y_2, \cdots, y_j]$

Define $D[i,j]$: MED from $X_i \to Y_j$.

Then, we know :

- $D[i-1, j]$: MED from $X_{i-1} \to Y_j$
- $D[i, j-1]$: MED from $X_i \to Y_{j-1}$
- $D[i-1, j-1]$: MED from $X_{i-1} \to Y_{j-1}$.

Let $O = o_1 o_2 \cdots o_k$ be the optimal operations occured in the MED process.

Example :

$$\begin{array}{cccccccccc} I & N & T & E & * & N & T & I & O & N \\ \downarrow & \downarrow & \downarrow & \vdots & \downarrow & \downarrow & \vdots & \vdots & \vdots & \vdots \\ * & E & X & E & C & U & T & I & O & N \end{array}$$

$$X_i = X_i^0 \xrightarrow{o_1} X_i^1 \xrightarrow{o_2} X_i^2 \to \cdots \xrightarrow{o_{k-1}} X_i^{k-1} \xrightarrow{o_k} X_i^k = Y_j$$

where o_k : $a \to b$

a : could be either $*$ or $x \in X_i$.

b : could be either $*$ or $y \in Y_j$.

- <u>optimal substructure</u>:

$$X_i = X_i^0 \xrightarrow{O_1} X_i^1 \xrightarrow{O_2} X_i^2 \rightarrow \cdots \xrightarrow{O_{k-1}} X_i^{k-1} \xrightarrow{O_k} X_i^k = Y_j$$

cut the last operation

$X_i \rightarrow X_i^{k-1}$: $O_1 O_2 \cdots O_{k-1}$ is still optimal operations from X_i to X_i^{k-1}. (Note X_i^{k-1} could be either Y_j or Y_{j-1}.

Why? (Hint: Make a contradiction)

$$O_k = \begin{cases} \text{case 1. } x_i \text{ is deleted from } X_{i-1} \\ \text{case 2. } y_i \text{ is inserted into } Y_{i-1} \\ \text{case 3. } x_i \text{ is substituted by } y_i \end{cases}$$

For case 1. : $D[i-1, j] + Del(x_i)$

$$D[i,j] = D[i-1, j] + Del(x_i)$$

For case 2:

$$D[i,j] = D[i, j-1] + Ins(y_i)$$

For case 3:

$$D[i,j] = D[i-1, j-1] + Sub(x_i, y_i)$$

Clearly, $D[i,j]$ must be one of them, i.e.,

$$D[i,j] = \min \begin{cases} D[i-1,j] + Del(x_i) \\ D[i,j-1] + Ins(y_j) \\ D[i-1,j-1] + Sub(x_i, y_j) \end{cases}$$

$i = 1, 2, \cdots, n$
$j = 1, 2, \cdots, m$

Note For $i = 0$, $D[i,j] = j$.

For $j = 0$, $D[i,j] = i$.

We call the above method a dynamic programming method.

Remarks and Questions:

① Prove or disprove the uniqueness of operations.

② Find out different type of costs (edit distance metrics).

③ If n, m are very large, can you find approximate solutions?

(Hint: find "approximate string matching").