Levenshtein Alignment (Complementary Material)

Jakub Waszczuk

January 2019

1 Levenshtein Alignment

Let $\boldsymbol{x} = x_1 \dots x_n$ and $\boldsymbol{y} = y_1 \dots y_m$ be two sentences of length n and m, respectively.

Definition 1. *Let $a \subset \{1, \dots, n\} \times \{1, \dots, m\}$ be an alignment between $\boldsymbol{x}$ and $\boldsymbol{y}$. We call a a* Levenshtein alignment *if it satisfies the following properties:*

- *each word x_i is aligned with at most one word y_j; formally*

$$\forall_{(i,j)\in a} \forall_{(i',j')\in a} \big(i = i' \implies j = j'\big) \tag{1}$$

- *each word y_i is aligned with at most one word x_j; formally*

$$\forall_{(i,j)\in a} \forall_{(i',j')\in a} \big(j = j' \implies i = i'\big) \tag{2}$$

- *there are no crossing alignment arcs; formally*

$$\forall_{(i,j)\in a} \forall_{(i',j')\in a} \big(i < i' \implies j < j'\big) \tag{3}$$

2 Minimal Levenshtein Alignment

Definition 2. *Let $a \subset \{1, \dots, n\} \times \{1, \dots, m\}$ be an alignment between $\boldsymbol{x}$ and $\boldsymbol{y}$. We calculate its* cost $c(a)$ *as follows:*

- *for each x_i that is not aligned ($\equiv$ deleted), we increase the score by* 1
- *for each y_i that is not aligned ($\equiv$ inserted), we increase the score by* 1
- *for each $(i, j) \in a$, (matching or substitution) we increase the score by $\delta(x_i = y_i)$*[1]

Definition 3. *Let $L(i, j)$ be the set of all possible Levenshtein alignments between $x_1^i = x_1 \dots x_i$ and $y_1^j = y_1 \dots y_i$. We define $Q(i, j)$ as the score of a* minimal Levenstein alignment *between x_1^i and y_1^j, i.e.:*

$$Q(i, j) = \min_{a \in L(i,j)} c(a) \tag{4}$$

[1] δ is the Kroenecker delta which takes the value of 1 if its argument is true, and 0 otherwise.

At the end of the day, we are only interested to determine $Q(n, m)$, i.e., the minimal Levenstein alignment score between the entire sentences $\boldsymbol{x}$ and $\boldsymbol{y}$. This gives us a measure of similary between both sentences. However, we define $Q(i, j)$ in a generic way (for any two valid positions i, j) because it allows to calculate the value of $Q(n, m)$ efficiently, based on a recursive formula. Indeed, calculating $Q(n, m)$ directly (based on Eq. 4) would be infeasible, due to the number of distinct Levenstein alignments $L(n, m)$ that can be created.

The recursive formula for $Q(i, j)$ can be defined as follows:[2]

$$Q(i,j) = \begin{cases} i & j = 0 \\ j & i = 0 \\ R(i,j) & \text{otherwise} \end{cases} \tag{5}$$

with the recursive part defined as:

$$R(i,j) = \min \begin{cases} Q(i-1, j-1) + \delta(x_i = y_j) & \text{match} \\ Q(i, j-1) + 1 & \text{insert} \\ Q(i-1, j) + 1 & \text{delete} \end{cases} \tag{6}$$

In order to show that Eq. 4 and Eq. 5 are equivalent we first introduce an auxiliary proposition.

Proposition 1. *Every Levenshtein alignment $a \in L(i, j)$ between x_1^i and y_1^j can be broken down in one of the three following ways:*

1. *if $(i, j) \in a$, then $a = a' \cup \{(i, j)\}$ for some $a' \in L(i-1, j-1)$*
2. *if x_i is not aligned in a, then $a \in L(i-1, j)$*
3. *if y_j is not aligned in a, then $a \in L(i, j-1)$*

Proof. It should be relatively easy to see that all three points above are true. For intsance, if $(i, j) \in a$, then no word in x_1^{i-1} can be aligned with y_j and no word in y_1^{j-1} can be aligned with x_i because that would contradict the definition of the Levenshtein alignment (see Def. 1) which states that each word can be matched with at most one word.

However, how do we know that we don't need to consider the alignment of x_i with some $y_k \colon k < j$? None of the three points above explicitely takes this possibility into account. This is because, if $(i, k) \in a \colon k < j$, then y_j cannot be aligned with anything in a (it cannot be aligned with x_i because x_i is already aligned; it cannot be aligned with $x_l \colon l < i$ because that would create a crossing alignment arc). Therefore, this case is already taken handled by point 3.

Similarly, if y_j is aligned with some $x_k \colon k < i$, then x_i cannot be aligned and this case is covered by point 2. □

Proposition 2. *For any $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, the value of $Q(i, j)$ is the same whether we calculate it using Eq. 4 or Eq. 5.*

[2] In contrast with what was presented during the lecture, there are several base cases (when either $i = 0$ or $j = 0$), because $Q(i, j)$ can be then easily calculated directly.

Proof. The proof is inductive. The base cases (either $i = 0$ or $j = 0$) are easy to handle, so let's focus on $i > 0$ and $j > 0$. Based on Def. 2 and Prop. 1, we can break down the calculation of $Q(i, j)$ as follows:

$$\begin{aligned} Q(i,j) = \min_{a \in L(i,j)} c(a) = \min\{ & \min_{a \in L(i-1,j-1)} c(a) + \delta(x_i = y_j) \\ & , \min_{a \in L(i,j-1)} c(a) + 1 \\ & , \min_{a \in L(i-1,j)} c(a) + 1\} \end{aligned}$$

In words, we consider all possible alignments in $L(i, j)$ by considering three cases: (i) that x_i is aligned with y_j, (ii) that y_j is inserted, and (iii) that x_i is deleted. From the inductive hypothesis, $\min_{a \in L(i-1,j-1)} c(a) = Q(i-1, j-1)$, $\min_{a \in L(i,j-1)} = Q(i, j-1)$, and $\min_{a \in L(i-1,j)} = Q(i-1, j)$. Therefore:

$$\begin{aligned} Q(i,j) = \min\{ & Q(i-1,j-1) + \delta(x_i = y_j) \\ & , Q(i,j-1) + 1 \\ & , Q(i-1,j) + 1\} \end{aligned}$$

□