

# What is Jaro-Winkler Similarity?

Authored by  
**stats writer**

November 29, 2025

## RECOMMENDED CITATION

stats writer (2025). *What is Jaro-Winkler Similarity?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=101573>

Jaro-Winkler Similarity is a string comparison algorithm used to measure the similarity between two strings. It is based on the Jaro Distance algorithm, but adds a bonus for strings that have a prefix in common. It is often used in record linkage and entity resolution tasks. It is a normalized measure of similarity between two strings, ranging from 0 (no similarity) to 1 (identical strings).

In statistics, the **Jaro-Winkler similarity** is a way to measure the similarity between two strings.

The **Jaro similarity** ( $sim_j$ ) between two strings is defined as:

$$sim_j = 1/3 * ( m /|s_1| + m /|s_2| + (m-t)/m )$$

where:

**m**: Number of matching characters

Two characters from  $s_1$  and  $s_2$  are considered matching if they are the same and not farther than - 1 characters apart.

**|s<sub>1</sub>|, |s<sub>2</sub>|**: The length of the first and second strings, respectively

**t**: Number of transpositions

Calculated as the number of matching (but different sequence order) characters divided by 2.

The **Jaro-Winkler similarity** ( $sim_w$ ) is defined as:

$$sim_w = sim_j + l_p(1 - sim_j)$$

where:

**sim<sub>j</sub>**: The Jaro similarity between two strings,  $s_1$  and  $s_2$

**l**: Length of the common prefix at the start of the string (max of 4 characters)

**p**: Scaling factor for how much the score is adjusted upwards for having common prefixes. Typically this is defined as  $p = 0.1$  and should not exceed  $p = 0.25$ .

The Jaro-Winkler similarity between two strings is always between 0 and 1 where:

**0** indicates no similarity between the strings

**1** indicates that the strings are an exact match

**Note**: The Jaro-Winkler *distance* would be defined as  $1 - sim_w$ .

The following example shows how to calculate the Jaro-Winkler similarity between two strings in practice.

## Example: Calculating Jaro-Winkler Similarity Between Two Strings

Suppose we have the following two strings:

String 1 (s1): **mouse**

String 2 (s2): **mute**

First, let's calculate the Jaro Similarity between these two strings:

where:

**m**: Number of matching characters

Two characters from s1 and s2 are considered matching if they are the same and not farther than - 1 characters apart.

In this case, - 1 is calculated as  $5/2 - 1 = 1.5$ . We would define three letters as matching: m, u, e. Thus, **m = 3**.

**|s1|, |s2|**: The length of the first and second strings, respectively

In this case, **|s1| = 5** and **|s2| = 4**.

**t**: Number of transpositions

Calculated as the number of matching (but different sequence order) characters divided by 2.

In this case, there are three matching characters but they're already in the same sequence order, so **t = 0**.

Thus, we would calculate the Jaro Similarity as:

$$\text{simj} = 1/3 * ( 3/5 + 3/4 + (3-0)/3 ) = 0.78333.$$

Next, let's calculate the Jaro-Winkler similarity (simw) as:

$$\text{simw} = \text{simj} + \text{lp}(1 - \text{simj})$$

In this case, we would calculate:

$$\text{simw} = 0.78333 + (1)*(0.1)(1 - 0.78333) = 0.805.$$

The Jaro-Winkler similarity between the two strings is **0.805**.

Since this value is close to 1, it tells us that the two strings are very similar.

We can confirm this is correct by calculating the Jaro-Winkler similarity between the two strings in R:

**library(stringdist)**

```
#calculate Jaro-Winkler similarity between 'mouse' and 'mute'  
1 - stringdist("mouse", "mute", method = "jw", p=0.1)
```

0.805

This matches the value that we calculated by hand.

The following tutorials explain how to calculate other similarity metrics: