**Fuzzy Algorithms**

**Hamming Distance:**

Strings must be the same length. This algorithm is best suited to finding corruption or errors in a transmission.

Score is number of differences. 0=The same, low number=similar, high number=different.

To put this into perspective, it should be compared to the length of the strings.

Example: "karolin" and "kerstin" is 3.

**Jaccard Index & Distance**

Index aka *‘Intersection over Union’* – measures similarity between two sets.

Distance measures dissimilarity. Complement to Index.

Jaccard returns a perfect match if both strings contain the same characters, even though one may have more of them and/or they are rearranged.

Example: Wizard Pendleton = Wizzard Pendelton

**Tanimoto:**

Similar to Jaccard, slightly different algorithm.

**Jaro Distance:**

Seems similar to Hamming, but works with different lengths. Uses Levenshtein.

**Jaro-Winkler:**

Gives more favorable ratings to strings that match from the beginning. Uses Levenshtein.

**Levenshtein Distance:**

Another distance algorithm

**Longest Common Substring:**

The number of chars that can be matched all in a row.

This will give a high score even if there are a bunch of differences, but they are near the start and end (as long as the middle matches). It will give a low score if there are only a few differences but they are near the middle.

**Longest Common Subsequence:**

Subsequences are not required to occupy consecutive positions. This performs a Longest-Common-Substring test, and then moves outward from the middle looking for more common substrings. Returns the total length of all found substrings.

**Ratcliff-Obershelp:**

Recognizes multiple common substrings.

**Sorensen-Dice:**

Similar to Jaccard, but solves ‘triangle inequality’ issue of Jaccard.

Distance is the number of differences.

Distance and Similarity are complementary.

Ratcliff-Obershelp and Sorensen-Dice Index use the exact same algorithm.