Fuzzy Matching

Autor: Helmut Wollmersdorfer

CPAN, github: wollmers

Deutscher Perl-Workshop 2016

https://github.com/wollmers/talks-gpw2016

Problem

- Measure Similarity
- Find similar objects
- Locate differences
- Recognize as „nearly the same“
- Clustering
- De-duplication
- Spelling correction

Objects

- sequence (string, array)

  'anna', ['a', 'n ', 'n', 'a']

- Set

  ['a', 'n']

- bag (=multiset)

  {'a' => 2, 'n' => 2}

Elements

- Characters (bytes, codepoints, graphemes)
- Words (or string tokens)

  ['The', 'big', 'brown']
- N-grams

  bi_gram('bar') => ['ba', 'ar']
- objects

Similarity of Sequences

Similarity('bar','bar') => 1

Similarity('foo','bar') => 0

Similarity('Nürnberg','Nuremberg') => 0.70

Nü**r**n **berg** matches 6 of 8

Nu**r**em**berg** matches 6 of 9

Similarity = matches * 2 / (length1 + length2)

= 6 * 2 / (8 + 9) = 12 / 17 = 0.70

Align ('diff') Sequences

- LCS (longest common subsequence)
- SES (shortest edit script, Levenshtein)
- Damerau

=> DEMO

Modules

- Algorithm::Diff(::XS)

  LCS, LCS_idx, diff, length_LCS
- String::Similarity
- LCS(::Tiny)

  LCS, LLCS, all_LCS
- LCS::BV (+ Perl 6)
- LCS::XS (alpha)

Smart Align

LCS::Similar

uses approximate comparison function for each element (or confusables table)

'double approximate'

Similarity of Sets

Dice = (A intersect B) / 0.5 (A + B)

Jaccard = (A intersect B) / (A union B)

Overlap = (A intersect B) / min(A,B)

Cosine = (A intersect B) / (sqrt(A) * sqrt(B))

Set::Similarity

Bag::Similarity

Approximate Search

- Large or big data

  > 1 Mio. per language

  > 10 Mio. names
- Smart index
- Fast access

Phonetic simplification

- Metaphone and friends
- Customize for other languages
- Numbers and punctuation?

SimString

- Set of CDBs (constant data base)
- CDB per string length
- N-Gram based
- Cosine, dice, jaccard, overlap, exact
- Finds similars in 1 ms out of 1 Mio.

forked to support UTF-8

github.com/wollmers/simstring

SimHash

- Approximate cosine by Hamming-Space
- Use SQL
- Patent by Google

not sharp enough for words (?)

Next Steps

Approximate, multi-level parsing

Support Vector Machine (SVM)

fast machine learning

Questions?

???