Streng-similaritet

Alternativer til Damerau-Levenshtein

Tidligere strengsimilaritetsmetrikker

- Damerau-Levenshteinavstand
  - Viderutvikling av Levenshtein-avstand
  - Antall tegnendringer for å gå fra streng A til streng B
- K-gram m/Jaccard-koeffisient
  - Sett-overlapp av k-gram i streng A og streng B
- Soundex
  - Similaritet basert på fonetisk matching

WER (Word Error Rate)

- Levenshtein-avstand på ord-nivå
  - (Ekstraoppgave i Assignment B-2)
- Metrikk for ytelsen til talegjenkjenningssystemer
- Score fra 0 til 1; 0 er identisk, 1 er ingen likhet
- Eksempel:
  Forventet: "*Jeg spiser et eple*"
  Faktisk: "*Jeg spiser en appelsin*"

  WER= $\frac{2+0+0}{4} = 0.5$

$$\mathbf{WER} = \frac{\mathbf{S + D + I}}{\mathbf{N}}$$

where...
S = number of substitutions
D = number of deletions
I = number of insertions
N = number of words in the reference

Jaro-Winkleravstand

- Streng-metrikk for å måle edit-distanse mellom to sekvenser
- Score mellom 0 og 1; 0 er eksakt match og 1 betyr ingen likhet

Jaro Similarity

The Jaro Similarity sim_j of two given strings s_1 and s_2 is

$$sim_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$$

Where:

- $|s_i|$ is the length of the string s_i;
- m is the number of *matching characters*
- t is half the number of *transpositions*

Jaro–Winkler Similarity

$$sim_w = sim_j + \ell p(1 - sim_j),$$

where:

- sim_j is the Jaro similarity for strings s_1 and s_2
- ℓ is the length of common prefix at the start of the string up to a maximum of 4 characters
- p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. p should not exceed 0.25 (i.e. 1/4, with 4 being the maximum length of the prefix being considered), otherwise the similarity could become larger than 1. The standard value for this constant in Winkler's work is $p = 0.1$

The Jaro-Winkler distance d_w is defined as $d_w = 1 - sim_w$.

Monge-Elkanavstand

- Kombinerer sekvens- og settbaserte metoder – score mellom 0 og 1
  - Krever flere *tokens*

- La A, B være to tekster, |A| og |B| antall tokens, *sim'* en similaritetsmetrikk på token-nivå
- Gjennomsnittlig avstand over mest like tokens
- Asymmetrisk

$$sim_{MongeElkan}(A,B) = \frac{1}{|A|}\sum_{i=1}^{|A|}\max\{sim'(a_i,b_j)\}_{j=1}^{|B|}$$

$A =$"$Lenovo\ \ inc.$"; $a_1 =$"$Lenovo$"; $a_2 =$"$inc.$"
$B =$"$Lenovo\ \ corp.$"; $b_1 =$"$Lenovo$"; $b_2 =$"$corp.$"
$sim'(a_1,b_1) = 1 - \frac{0}{6} = 1;\ \ sim'(a_1,b_2) = 1 - \frac{5}{6} = 0.1666$
$sim'(a_2,b_1) = 1 - \frac{5}{6} = 0.1666;\ \ sim'(a_2,b_2) = 1 - \frac{4}{4} = 0$
$sim_{MongeElkan}(A,B) = \frac{1}{2}(\max(sim'(a_1,b_1), sim'(a_1,b_2)) + \max(sim'(a_2,b_1), sim'(a_2,b_2)))$
$sim_{MongeElkan}(A,B) = \frac{1}{2}(1 + 0.1666) = 0.5833$

Kilder: Jimenez, S., Becerra, C., Gelbukh, A., & Gonzalez, F. *Generalized Mongue-Elkan Method for Approximate Text String Comparison*. National University of Colombia; National Polytechnic Institute, Mexico. (https://link.springer.com/chapter/10.1007/978-3-642-00382-0_45#:~:text=The%20Mongue%2DElkan%20method%20is,i.e.%20word%20level)%20similarity%20measure.)

Snowball – Stemming på norsk

1:

- (a, e, ede, ande, ende, ane, ene, hetene, en, heten, …) - slett
- (s) hvis etterfølger b, d, f, g, … – slett
- (erte, ert) – erstatt med er

2:

- Hvis slutter på (dt, vt) – slett t

3:

- (leg, eleg, ig, eig, lig, elig, els, lov, elov, slov, hetslov) - slett

Kilder: http://snowball.tartarus.org/algorithms/norwegian/stemmer.html

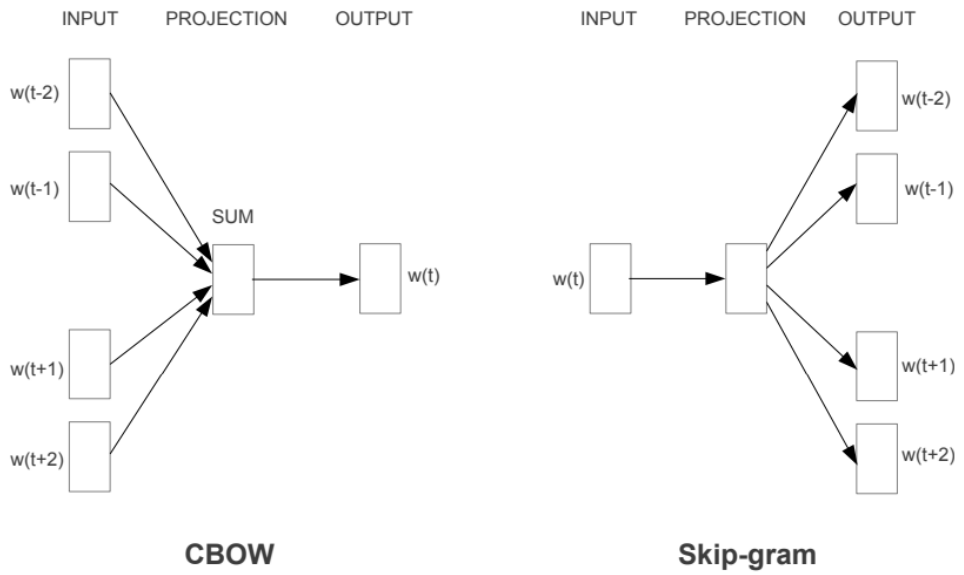
word		stem	word		stem
havnedistrikt		havnedistrikt	opning		opning
havnedistriktene		havnedistrikt	opninga		opning
havnedistrikter		havnedistrikt	opningsbalanse		opningsbalans
havnedistriktet		havnedistrikt	opningsbalansen		opningsbalans
havnedistriktets		havnedistrikt	opp		opp
havnedrift		havnedrift	oppad		oppad
havnedriften		havnedrift	opparbeide		opparbeid
havneeffektivitet		havneeffektivit	opparbeidede		opparbeid
havneeier		havneei	opparbeidelse		opparbeid
havneeiere		havneeier	opparbeider		opparbeid
havneenheter		havneen	opparbeides		opparbeid
havneforbund		havneforbund	opparbeidet		opparbeid
havneforbundets		havneforbund	opparbeiding		opparbeiding
havneformål		havneformål	oppattbygging		oppattbygging
havneforvaltningen		havneforvaltning	oppbevarer		oppbevar
havnefunksjonene		havnefunksjon	oppbevaring		oppbevaring
havnefunksjoner		havnefunksjon	oppblåst		oppblåst
havnefylkene		havnefylk	oppblåste		oppblåst
havnefylker		havnefylk	oppbrente		oppbrent
havnehagen	=>	havnehag	oppbygd	=>	oppbygd
havneinfrastrukturen		havneinfrastruktur	oppbygde		oppbygd
havneinnretningene		havneinnretning	oppbygget		oppbygg
havneinnretninger		havneinnretning	oppbygging		oppbygging
havneinteresser		havneinteress	oppbygginga		oppbygging
havnekapasitet		havnekapasit	oppbyggingen		oppbygging
havnekassa		havnekass	oppdage		oppdag
havnekasse		havnekass	oppdager		oppdag
havnekassemidler		havnekassemidl	oppdaterte		oppdater
havnekassen		havnekass	oppdeling		oppdeling
havnekassene		havnekass	oppdelingen		oppdeling
havnekassens		havnekass	oppdelt		oppdelt
havnelokalisering		havnelokalisering	oppdrag		oppdrag
havneloven		havn	oppdraget		oppdrag
havnelovens		havn	oppdragsavtale		oppdragsavtal
havneløsning		havneløsning	oppdragsgivere		oppdragsgiver
havneløsningene		havneløsning	oppdragstakaren		oppdragstakar
havneløsninger		havneløsning	oppe		opp
havnemessig		havnemess	oppebærer		oppebær
havnemyndighetene		havnemynd	oppfarende		oppfar
havnemyndigheter		havnemynd	oppfatning		oppfatning

```
/* special characters (in ISO Latin I) */

stringdef ae   hex 'E6'
stringdef ao   hex 'E5'
stringdef o/   hex 'F8'

define v 'aeiouy{ae}{ao}{o/}'

define s_ending  'bcdfghjlmnoprtvyz'

define mark_regions as (

    $p1 = limit

    test ( hop 3 setmark x )
    goto v  gopast non-v  setmark p1
    try ( $p1 < x  $p1 = x )
)

backwardmode (

    define main_suffix as (
        setlimit tomark p1 for ([substring])
        among(

            'a' 'e' 'ede' 'ande' 'ende' 'ane' 'ene' 'hetene' 'en' 'heten' 'ar'
            'er' 'heter' 'as' 'es' 'edes' 'endes' 'enes' 'hetenes' 'ens'
            'hetens' 'ers' 'ets' 'et' 'het' 'ast'
                (delete)
            's'
                (s_ending or ('k' non-v) delete)
            'erte' 'ert'
                (<-'er')
        )
    )

    define consonant_pair as (
        test (
            setlimit tomark p1 for ([substring])
            among(
                'dt' 'vt'
            )
        )
        next] delete
    )

    define other_suffix as (
        setlimit tomark p1 for ([substring])
        among(
            'leg' 'eleg' 'ig' 'eig' 'lig' 'elig' 'els' 'lov' 'elov' 'slov'
            'hetslov'
                (delete)
        )
    )
)

define stem as (

    do mark_regions
    backwards (
        do main_suffix
        do consonant_pair
        do other_suffix
    )
)
```

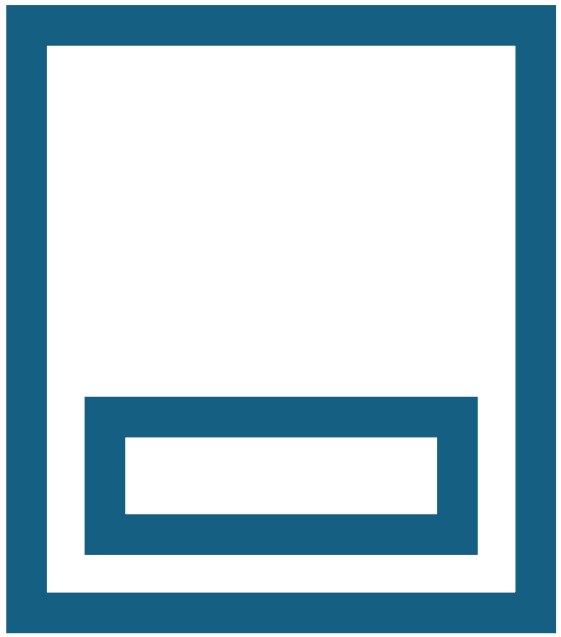
Kilder:
http://snowball.tartarus.org/algorithms/norwegian/stemmer.html

Semantisk similaritet

- Klasse similaritetsmetrikker; basert på NLP
- Ignorerer tegnsimilaritet
  - Fokuserer på bruk – synonymer og relaterte ord
- Word2vec (2013):
  - Trening – CBOW og Skip-gram
  - Resultat: Vektorrom av vokabular
  - Bruk cosinus-similaritet over vokabular
  - Behov for stort treningssett



- Kilder: Mikolov, T., Chen, K., Corrado, G., & Dean, J. *Efficient Estimation of Word Representations in Vector Space*. Google Inc., Mountain View, CA. (https://arxiv.org/pdf/1301.3781); https://www.geeksforgeeks.org/different-techniques-for-sentence-semantic-similarity-in-nlp/



Andre similaritetsmetrikker

- LCS (Longest common subsequence)
- Metaphone
  - Forbedret versjon av Soundex, kun for engelsk
- Gestalt pattern matching

- Universal sentence encoder
- BERT