

LEXDIS query types: Measuring lexical proximity

FULL BASIC QUERY : [Word1,pos:Pos1,Word2,pos:Pos2,w:Weight,m:Mode,Flag_adjust]

Word1 and Word2 can be quoted 'attache case', 'go through'

Pos : adj, adv, n, v

Weight : [2-100] (minimum threshold) , t (=top), t3 (=best 3), none, m(merging entries)

Mode : g (=global), l (=local)

Flag_adjust : adjust, noadjust

Query expansion

- A. [Word1,pos:Pos,Word2] ---> [Word1,pos:Pos,Word2,pos:Pos,w:none,m:g,adjust]
no minimum weight specified : we set it to 'none',
no mode specified : we set it to 'global'
• shared POS
- B. [Word1,pos:Pos1,Word2,pos:Pos2] ---> [Word1,pos:Pos1,Word2,pos:Pos2,w:none,m:g,adjust]
we supply global mode and weight is set to 'none'
- C. [Word1,pos:Pos,Word2,m:Mode] ---> [Word1,pos:Pos,Word2,pos:Pos,w:none,m:Mode,adjust]
no minimum weight specified : we set it to 'none'
• shared POS
- D. [Word1,pos:Pos1,Word2,pos:Pos2,m:Mode] --->
[Word1,pos:Pos1,Word2,pos:Pos2,w:none,m:Mode,adjust]
no minimum weight specified : we set it to 'none'
- E. [Word1,pos:Pos1,Word2,pos:Pos2,w:MW,m:Mode] --->
[Word1,pos:Pos1,Word2,pos:Pos2,w:MW,m:Mode,adjust]
we supply, as everywhere else above, the adjust flag

NOTE: THE 'noadjust' FLAG MUST BE MENTIONED AS LAST ARG AND IN FULLY EXPANDED QUERIES TO BE OPERATIVE

e.g. [cat,pos:n,dog,pos:n,w:none,m:g,noadjust]

The adjust/noadjust flag has no import in local mode

Proposed Query Types

- A. [Word1,pos:Pos1,Word2,pos:Pos2,m:Mode].
- B. [Word1,pos:Pos,Word2,m:Mode].
- C. [Word1,pos:Pos1,Word2,pos:Pos2,w:MW,m:Mode].

where

Mode is either g(global:words)
l(local:wordsenses),

and

Pos1, Pos2 and Pos are in [adj,adv,n,v]

and

MW is - an integer expressing the required minimum connectivity threshold
- t to get the top pairings only (restricted to m:l, i.e local mode)

Displaying lexical items

A. `show(word,pos,spec:data).`

where `spec` is either

`gw,`

`lb,`

`df`

`ex`

and `data` is a label, guideword or word

B. `show(Idnum).`

where `Idnum` is an identifier returned by the system

Exiting

nadamas. to exit

Additional query types

% triplets: t(Target,PosTarget,Arrow1,PosArrow,Arrow2,PosArrow)

 or

 t(Target,Arrow1,Arrow2) expanded to t(Target,v,Arrow1,n,Arrow2,n)

% best three matches : [Word1,pos:Pos1, Word2,pos:Pos2,w:t3,m:l,noadjust]

% merge mode : [Word,pos:Pos, Word,pos:Pos,w:m,m:l,noadjust]

% idnum instead of pos in local mode: [Word,Idnum, Word2,pos:Pos,w:Threshold,m:l]

% computing the LEXDIS weights for all semdic items : w(Word,Pos) (needs semdic, not lightdic !)

% finding the 'friends' of a Word,Pos pair : friends(Word,pos:Pos,w:MinimumWeight)

Operators in show mode

AND e.g. show(Word,Pos,df:and(Defellist)

OR e.g. show(Word,Pos,df:or(Defellist)

Note that the args to the AND and OR operators need to be LISTS
+ show(Word,Pos)