NAME

wsh – 連想計算を行う

SYNOPSIS

```
#include <geta/ae.h>

syminfo * wsh( syminfo *q, int qlen, WAM *wam, int dir_q, int weight_type,
        int *ndp, int *totalp, void *cookie,
        int (*ucompar)(syminfo *, syminfo *) )
```

DESCRIPTION

wsh は与えられた行(列)のIDリスト *q* 中の各行(列)IDに対応するベクタに現れるID(すなわち列(行)ID)を最大で *ndp 個集め, リストにして返す.

q の要素で, メンバ attr に WSH_AND をセットしているものは必須であり, それらの ID のうち一つでも欠くものは集められない. attr に WSH_NOT をセットしたものは除外を示し, それらの ID を一つでも含むものは集められない.

なお, *ndp 個以上の ID が発見される場合には, その weight がより大きいものを, weight が等しいもの同士については ID が小さいものを優先して集める. なお, この weight の比較は, *ucompar(a, b)* に a が b より真に小さい場合に 1 (true) を, a と b が等しいか a が b より大きい場合に 0 (false) を返す関数を渡すことにより override することができる.

wsh は集めた列(行)ID と *q* との類似度を計算する. 類似度の計算式は *weight_type* で指定された名前を持つ計算式ファイル $GETASRC/lib/ae/wt/<weight_type>.f で与えられる(厳密には計算式の名前は各ファイルの name セクションで付けられる). 計算式のカスタマイズの方法については $GETASRC/lib/ae/wt/custom.txt を参照されたい. *q* の長さは *qlen* で与えられる. *wam* には *q* の属する WAM を, *dir_q* には *q* が 行IDのリストならば WAM_ROW を, 列IDのリストならば WAM_COL を渡す.

入力 *q* の各メンバは以下の様にセットされていなければならない.

メンバ	説明
id	q の ID (wam(dir_q) で引ける. random order 可, 重複不可)
TF_d	TF(ID|current_document_set) を与える
TF	TF(ID|source_corpus) を与える
weight	重み. 不要であれば不定値 (NaN) で良い
attr	WSH_OR, WSH_AND, WSH_NOT のいずれか
その他	不定で良い

集められた列(行)IDは, **wsh** が確保した syminfo * 型の配列に詰め合わされて返される. ユーザは結果が不要になった場合, free(3) によりこの領域を解放することができる. **wsh** が返した配列の長さは *ndp* が指す場所に格納される. なお結果の配列は(nil などで)終端されてはいない. *totalp* が NULL でなければ, 発見された全IDの数が *totalp* が指す場所に格納される. 結果の配列はソートされていない. **wsh** が返す配列の各要素(以下 dp-> と記述)のメンバには次の値がセットされる.

メンバ	説明
id	ID (wam(!dir_q) で引ける方のID)
DF_d	DF(dp->id|q)
DF	DF(dp->id|wam)
TF_d	TF(dp->id|q)
TF	TF(dp->id|wam)
weight	similarity(dp->id | q) -- if weight_type != WT_NULL 0 -- if weight_type == WT_NONE

attr WSH_OR

wsh は *cookie* に渡された値を使わない. この引き数はユーザが wt/<weight_type>.f に記述したコードにデータを渡すために自由に使うことができる.

SEE ALSO

wam(3), ws(3), wsp(3), wsq(3), syminfo(3), dump_syminfo(3)