話題性原指標計算ライブラリ DREP

使用マニュアル

目次

1. インストール

1.1 配布

動作環境

UNIX 系 OS 上（汎用連想検索エンジン GETA の動作する環境)。

汎用連想検索エンジン GETA インストール済みコンピュータ

DREP ライブラリパッケージ付属物

	ファイル名	備考
プログラムソース	Makefile	
	trep.c	サンプルプログラム
	trep.h	
	tr.c	文字列集合から文書集合へのマップ関数群
	dr.c	文書集合の話題性原尺度計算関数
	tr.h	
	dr.h	
	getopt.c	サンプルプログラム、trep 用のオプション処理
	opt.h	
	commonDef.h	マクロ定義
テスト用データ	t/etc/ ci.conf	WAM 設定ファイル。中身については、「汎用連想検索エンジン（第３版）導入・操作マニュアル」を参照
	t/test1.sh	テスト用スクリプト
	t/test2.sh	テスト用スクリプト
	t/data/o1	テスト用正解データ
	t/data/o2	テスト用正解データ
	t/data/getadoc/ cw.c cw.r xr.c xr.r	テストコーパスから作成した WAM データ

1.2 インストール

1) GETA をインストールする

汎用連想検索エンジン GETA をインストールする。詳細については、「汎用連想検索エンジン（第３版）導入・操作マニュアル」を参照のこと。以降、GETA をインストールしたディレクトリを、$geta_root で表す。

2) DREP ライブラリパッケージを適当な場所で展開する。以降、展開先のディレクトリを、$drep_dir で表す。

2) コンパイル

DREP パッケージを展開したディレクトリで、GETA をインストールした場所を指定してコンパイルする。

```
% make GETAROOT=<GETA  installed pathname>
```

※ csh 系のシェルを使用の場合、環境線数を設定してコンパイルしてもよい。

```
% setenv GETAROOT   <GETA   installed pathname>
% make
```

3) 動作確認

illegalなオプションを与えた場合の出力確認､及び話題性原尺度計算結果の確認を行う。

```
% make test1
test for single term.
…
Test done.
% make test2
test for boolean combination of terms
…
Test done
%
```

動作中、検索文字列と話題性原尺度の値等が出力される。最後に、”Test done.”と表示されれば、正しくが行われていることを意味する。最後に”The output of test program is not correct.”というようなメッセージが表示された場合､計算結果が正解と異なるものがあることを意味する。

正解出力は、$drep_dir/t/data/o1,o2 に、実際の計算結果は $drep_dri/t/data/output.txt に納められているので、詳細は中身を見比べる必要がある。

2. DREP ライブラリ

2.1 ライブラリ全体構成

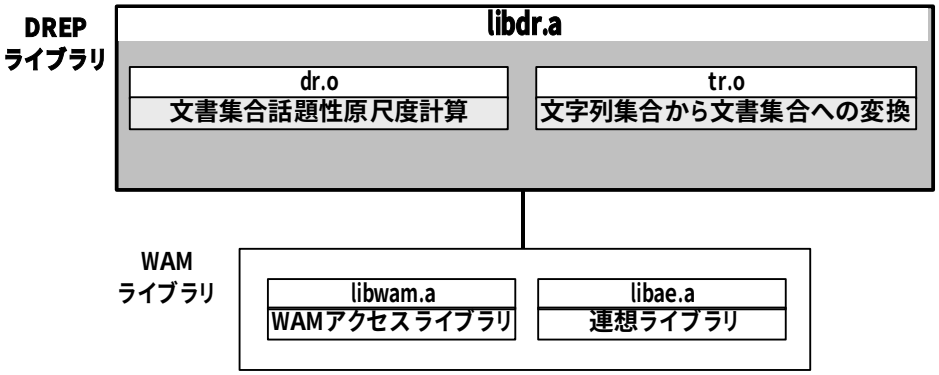

図 1：ライブラリの全体構成

表 1：提供関数構成

	関数名	定義箇所
TR 関数群	tr_mk_terminfo	tr.c
	tr_naive_bool	tr.c
DR 関数群	dr_init_handle	dr.c
	dr_clear_handle	dr.c
	dr_doc_repres_raw	dr.c

DREP は、話題性原尺度を計算する関数や、文字列集合による文書検索を行う関数を、ライブラリ libdr.a の形で提供する。libdr.a は、dr.o と tr.o の 2 つのオブジェクトからなる。libdr.a の関数群は、WAM ライブラリを使用して、単語ベクトルや記事ベクトルの連想計算を行う。

2.2 機能仕様

2.2.1. 単語ブール結合から文書集合への変換関数群(TR 関数)

単語あるいは単語の集合を、WAM の特定の Handle を用いて、記事集合にマップする。tr.c において定義。

tr_mk_terminfo

入力の文字列リストと文字列数から、syminfo 型の配列で表現した単語集合を作成する。syminfo 型については、$geta_root/include/ae.h を参照されたい。作成された単語集合を用いて文書の検索を行う際は、各単語の and 検索を行う。具体的には、syminfo 構造体の attr 属性に、WSH_AND をセットする。

文字列リストの先頭が'予約文字とマッチした場合、別の解釈を行う。具体的には表 2 を参照されたい。文字列の先頭から予約文字を省いた残りの文字列が、全て半角数字から構成されている場合は、WAM の列要素の ID と解釈する。

表 2：文字列パターンと syminfo の attr 属性との関係

文字列のパターン	syminfo の attr 属性	文書検索時の解釈
マイニング	WSH_AND	“マイニング” を含む
^マイニング	WSH_NOT	“マイニング” を含まない

tr_naive_bool

syminfo 型配列で表現される単語集合を元に、限定された BOOLEAN 検索を行い、syminfo 型配列の文書集合を作成する。図 2 に処理概要を示す。限定された BOOLEAN 検索とは、「単語あるいは単語の否定の集合の AND 検索のみ」ということを意味する。

例)

W1 ∩ W2 ∩ W3	W1 と W2 と W3 を含む文書集合を検索
W1 ∩ W2 ∩ !W3	W1 と W2 を含み、W3 を含まない文書集合を検索

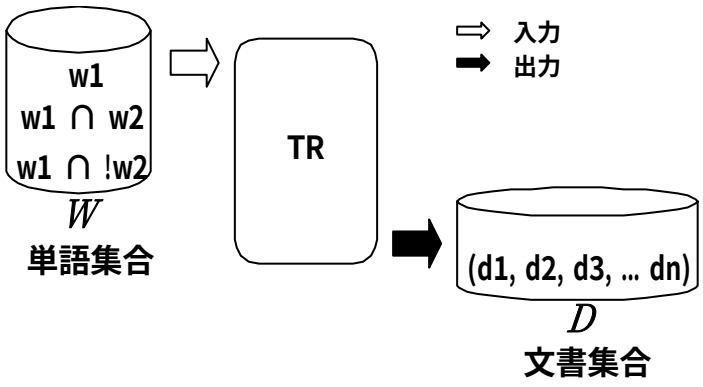

図 2：TR 関数群機能概要

2.2.2. 話題性原尺度計算関数群(DR 関数)

文書集合に対して、話題性原尺度を与える。dr.c において定義。

dr_init_handle

WAM の初期化とオープンを行い、WAM の行サイズ、列サイズ、総単語数を、DVH 型構造体に記録する。

dr_clear_handle

dr_init_handle で開いた WAM をクローズする。

dr_doc_repress_raw

文書集合に対し、DIST、DIFFNUM（報告書「単語の話題性原指標計算に関する調査研究」の 2.4 節を参照）等の話題性原尺度を計算する。

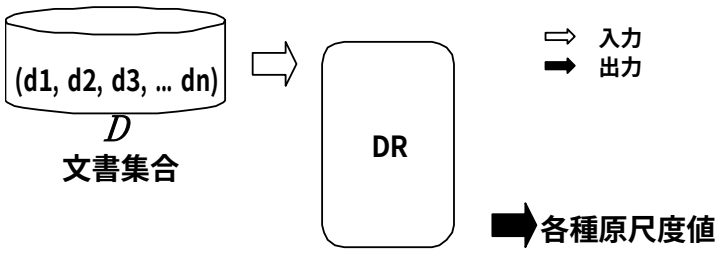

図 3：DR 関数群機能概要

2.3 I/F 仕様

2.3.1. マクロ定義

マクロ名	値	定義箇所	内 容	備考
BOOL	long	commonDef.h	提供関数の返値の型として使用	
DREP_TRUE	1	commonDef.h	関数が正しく終了	TR、DR 関数群の返値として使用
DREP_FALSE	0	commonDef.h	関数内で、エラー	

2.3.2. 構造体

syminfo 構造体

$GETAROOT/include/geta/ae.h で定義

DVH 構造体

```
typedef struct _DVH {
    WAM *w;

    long colSize;    /* WAMの列サイズ */
    long rowSize;    /* WAMの行サイズ*/
    long N;          /* WAMの総単語頻度 */
} DVH;
```

DRep 構造体

```
typedef struct _DRep {
    double tfidf;
    double llr;

    long df;    /* document number */
    long tf;    /* term frequency of documents set */
    long dtf;   /* different term frequency of documents set */

    long DF;    /* rowSize */
    long TF;    /* term frequency of whole documents */
    long DTF;   /* different term frequency of whole documents */
} DRep;
```

2.3.3. TR 関数

関数名	**tr_mk_terminfo**	
引数	**DVH *pDvh**	**dr_init_handle** で生成した **DVH** 構造体を渡す。
	char **ppStr	文字列リスト。各文字列は、**WAM** の列ベクトルに存在する単語か、**WAM** の列ベクトル **ID** である必要がある。
	long termCnt	文字列の数
	sysinfo *termVector	作成された **syminfo** 配列型単語集合。**ユーザが最後に,free する必要がある。**
返値	**BOOL**	**DREP_TRUE:** 正常 **DREP_FALSE:** エラー
機能	文字列のリストから、**syminfo** 型の配列を作成する。	

関数名	**tr_naive_bool**	
引数	**DVH *pResDVH**	**dr_init_handle** で生成した **DVH** 構造体を渡す。
	syminfo *termVec	検索単語集合
	long termCnt	単語数
	long docLimit	ドキュメントサンプリング上限。この引数で与えた数以上の文書数が検索された場合、ランダムに **docLimit** 個の文書を選択し、第 **5** 引数 **pDocVec** に渡す。それ以外の場合は、検索された文書全体を渡す。
	syminfo **pDocVec	検索結果の文書集合へのポインタ。**ユーザが最後に free する必要がある。**
	long *pDocSize	検索結果の文書集合の要素数
返値	**BOOL**	**DREP_TRUE:** 正常 **DREP_FALSE:** エラー
機能	**syminfo** 型の配列で表現された単語集合を元に,限定された **boolean** 検索を行い、**syminfo** 型の配列で表現された文書集合を作成する。	

2.3.4. DR 関数

関数名	**dr_init_handle**	
引数	**char *pHandle**	ハンドル名
	char *pProjectRoot	**GETA** インストールディレクトリ。**NULL** を与えた場合、環境変数 **GETAROOT** の値を使用する。
	DVH *pDvh	文書集合ハンドル構造体。オープンした **WAM** ハンドル,**WAM** の行サイズ、列サイズ、総単語数等を保持。
返値	**BOOL**	**DREP_TRUE:** 正常 **DREP_FALSE:** エラー
機能	文書集合ハンドル初期化。**WAM** の初期化とハンドルのオープンを行い、**WAM** マトリクスの行サイズと列サイズと総単語数を保持する。	

関数名	**dr_clear_handle**	
引数	**DVH *pDvh**	文書集合ハンドル構造体
返値	**void**	
機能	文書集合ハンドル開放	

関数名	**dr_doc_repres_raw**	
引数	**DVH *pDvh**	文書集合ハンドル構造体
	syminfo *docVec	文書集合
	int docSIze	文書集合サイズ
	DRep *pVal	話題性原尺度の計算結果を保持する構造体
返値	**BOOL**	**DREP_TRUE:** 正常 **DREP_FALSE:** エラー
機能	文書集合の原尺度尺度計算	

3. サンプルプログラム TREP

前節で説明した DREP ライブラリを用いて、入力の単語リストで検索した文書集合の話題性原尺度を計算する。

3.1 使用方法

準備

汎用検索エンジン GETA マニュアルの、「WAM の基礎」を参照しながら、検索対象となる文書群についての WAM データを作成する。通常、ハンドルに関する設定を

$geta_root/etc/ci.conf

に記述し、WAM データを

$geta_root/data/<handle_name>/

以下に作成する。

実行

trep プログラムは、以下のように、ハンドル名と、検索単語を引数として与え、コマン

ドラインで実行する。

```
% trep [オプション] [単語…]
```

※オプション

-d　handle_name

検索対象コーパスの WAM 表現のハンドル名

-m　upper_limit

ランダムサンプリングする文書数の上限値[1]。

-g　geta_root

GETAROOT を指定する。未指定の場合、環境変数 GETAROOT の値が用いられる。

-d オプションで与えるハンドル名は、$geta_root/etc/ci.conf に設定が記述されている必要がある。

処理内容

コーパス中の単語の *DIFFNUM*、*DIST* 等話題性原尺度を計算する。コーパス名は、オプションで指定する。

使用例

```
% trep –d nikkei98 –m 300 オウム
オウム
6522
LLR=103245.998988
DTF=8622
%
```

3.2　入力

オプションの後に、空白で区切った文字列を指定する。特定の単語を含まない文書を検索する場合は、'^ 'を文字列の先頭に付ける。

例）

辞書作成　自動　^画像　　辞書作成と自動を含み、画像を含まない文書を検索

3.3　出力

```
0 行目:　% trep –d nikkei98 –m 300 オウム　^サリン
1 行目:　オウム* ~サリン
2 行目:　6522* ~7471
3 行目:　LLR=55265.673426
4 行目:　DTF=6781
5 行目:　%
```

[1] 話題性原尺度の計算時間を短縮するため、閾値以上の文書集合については、その中からランダムサンプリングした部分集合により、原尺度の計算を行う。

例を用いて、出力の意味について説明する。出力の１行目は,検索に用いた単語のパターンを表す。'*'記号は単語の論理積、'~'は単語を含まないことを表す。２行目は、同じものを、文字列の WAM 列の ID で表現したものを表す。

3 行目から 5 行目は、上で指定した検索の結果得られた文書集合に対する、話題性原尺度を表す。LLR は「オウムを含みサリンを含まない」文書全体に含まれる単語の集合（重複を許す）D（”オウム”＊～”サリン”）内の単語分布とコーパス全体の単語分布の距離 *DIST*(*D*（”オウム”＊＾”サリン”）)を対数尤度比を用いて測った値、DTF は文書集合 *D*（”オウム”＊＾”サリン”）中の異なり単語数 *DIFFNUM*(*D*（”オウム”＊ ～”サリン”）)である。*DIST* および *DIFFNUM* については、成果報告書「単語の話題性尺度計算に関する調査研究」の 2．4 及び 2．5 節を参照されたい。