NICT Web クローラ
マニュアル

Ver. 1.2

平成 25 年 7 月 23 日

独立行政法人　情報通信研究機構

■更新履歴

Version	更新日	更新内容
1.0	2013/5/14	初版発行
1.1	2013/6/19	「4. インストール方法」を修正 「問い合わせ先」を追記
1.2	2013/07/23	「4. インストール方法」を修正

■目次

1. NICT Web クローラについて

1.1. クローラとは？

クローラとは、インターネット上の Web ページを自動収集するプログラムです。クローラは、与えられた URL で示される Web ページをダウンロードすると、その Web ページに張られたリンクから URL を抽出し、リンク先の Web ページをダウンロードします。これを繰り返すことで、リンクを辿りながら自動で Web ページを収集する仕組みです。

クローラは、検索エンジン用のデータベース作成に活用される他に、Web ページの保存を目的としたアーカイブや統計調査などに利用されています。その目的により、新しく作成された Web ページのみ収集するクローラもあれば、Web ページの更新を検知して内容に変更があった場合は再収集するもの、画像なども含めて Web ページ全体を取得するものや文字データのみを収集するものなどがあります。

1.2. NICT Web クローラの特徴

NICT Webクローラは、独立行政法人 情報通信研究機構（以下、NICTとする）が開発したクローラです。非同期並列のDNS名前解決、非同期HTTPリクエストの発行、robots.txtへの対応、同一ホストに対するアクセス頻度の制御などのクローラとしての基本的な機能を備えています[1] [2]。

特徴としては、Web ページの更新間隔推定機能があります。収集済みの Web ページを最新の状態に保つためには、定期的に再アクセスし、更新されていた場合は再ダウンロードする必要があります。しかしながら、再アクセスの間隔が長すぎると最新状態を保つことができません。一方で、間隔が短すぎると、まだ更新されていないタイミングで Web ページへアクセスし、無用なリソースを消費することになります。NICT Web クローラでは、Web ページの更新間隔を推定することで効率的に更新チェックを行い、最新状態の Web ページを収集することが可能となります。

これらの機能を備えたNICT Webクローラは、情報分析システムWISDOM[3]用のクローラとして動作しており、累計約 69 億ページの日本語Webページ（ユニークURL数は約 10 億）の収集実績[4]があります。

1.3. NICT Web クローラの動作概要

図 1 に NICT Web クローラの動作概要を示し説明します。

---

[1] クローラとしての基本的な機能には、東京大学の田浦 健次朗先生により開発された exCrawler をベースとしたモジュールを用いています。

[2] NICT Web クローラは対象 Web ページの HTML を収集します。画像、動画などは収集しません。

[3] 情報分析システム WISDOM (Web Information Sensibly and Discreetly Ordered and Marshaled)は Web 上にある情報を様々な観点から分析することによって、ユーザが情報を多角的に捉えながら情報の信頼性を判断できるように支援するシステムです。

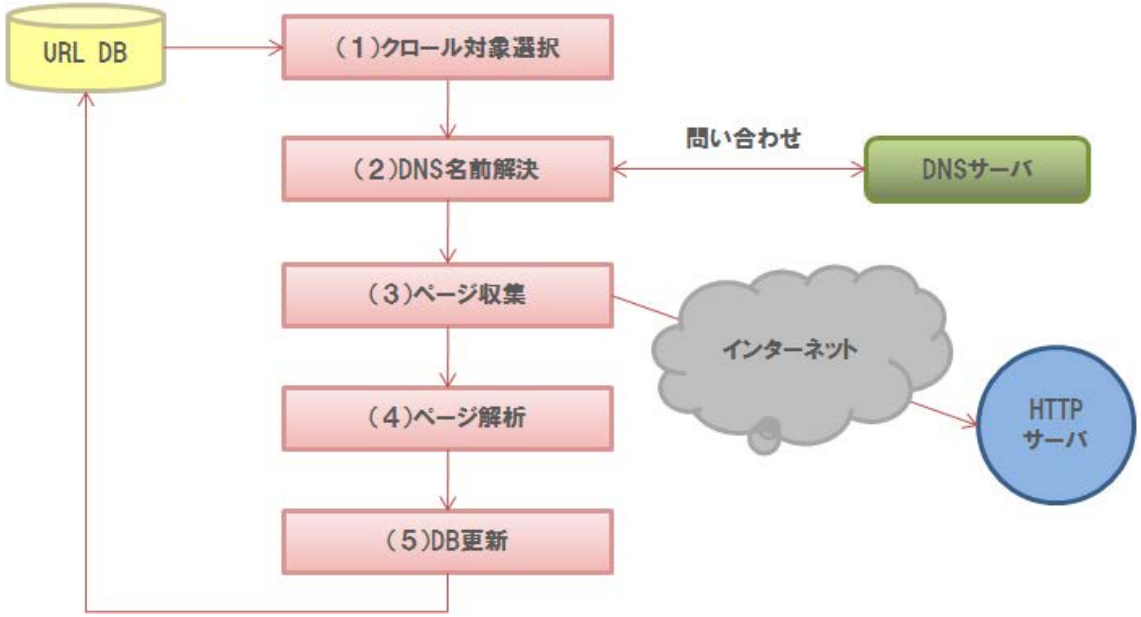

図 1 NICT Web クローラ動作概要

（1） クロール対象選択

URL DB から収集対象の URL を選択します。

URL に関する情報は、URL DB と呼ばれるデータベースにより管理しています。URL ごとに最終更新時刻、累計更新回数、平均更新間隔などの情報を管理しています。収集済みの Web ページに再アクセスする場合は、最終更新時刻や平均更新間隔の値などから現時点でその Web ページが更新されている確率を計算し、その確率に応じて収集対象を選択しています。

（2） DNS 名前解決

ホスト名の名前解決を行います。

URL に含まれるホスト名から名前解決のために、非同期で DNS サーバに問い合わせを行い、対象 Web ページを提供する Web サーバの IP アドレスを取得します。

（3） ページ収集

Web ページを取得します。

IP アドレスにより指定される Web サーバにアクセスを行い、Web ページを取得します。 Web ページの収集に際して、NICT Web クローラはアクセス先のサイトが定めるポリシーを遵守するため、robots.txt によりアクセスが禁止されている

[4] 2012 年 9 月 30 日時点の実績です。

Web ページの取得は行わない仕組みとなっています。また、Web ページの収集によりサイトに対して負荷をかけすぎないよう、同一ホストあるいは同一 IP アドレスで指定されるサーバに対しては、一定以上の間隔を空けてアクセスする制御が可能となっています。

（４） ページ解析

収集した Web ページの HTML を解析します。

収集した Web ページに対して、リンク抽出、フィルタリング、更新判定などの処理を実施します。リンク抽出では、HTML をパースし、リンクとして張られている URL 及びアンカーテキストを抽出します。フィルタリングでは、複数の条件により URL をチェックし、不要な URL を収集対象から除外します。更新判定では、HTML ファイルのチェックサムを計算し、前回の収集時に保存していたチェックサムと比較することで更新の有無を判定します。

（５） DB 更新

収集結果と抽出したリンク情報を URL DB へ反映します。

収集結果は、HTTP ステータス情報、クロール時刻、更新の有無、フィルタ適用の有無などの情報を含みます。リンク情報は URL DB に既に登録済みかどうかを確認した上で、未登録の場合は新規に登録します。

以上の（1）から（5）を 1 サイクルとして、繰り返しクロールを実施します。

2. 注意事項

クローラのご利用に際して、ご注意して頂きたい点を以下に記載します。内容をよく理解した上でご利用頂けるようお願い致します。

【Web サーバへのアクセス間隔について】

- 本ソフトウェアの使用に際しては、Web サーバに過度の負荷を掛けることのないよう適切なアクセス間隔を設定して下さい。短期間に連続してアクセスを繰り返した場合、相手サーバに対して損害を及ぼす可能性があります。
- また上記理由により、過度のアクセスは Web サーバの機能不全を目的とした敵対的攻撃と見做される恐れがありますのでご注意下さい。
- アクセス間隔として、どの程度の間隔が適切であるかの明確な規定はありません。相手サーバの能力やサーバ管理者の判断によりますので、必要以上にアクセス間隔を狭めずに運用されるようご注意下さい。なお、NICT Web クローラの初期設定では 60 秒間隔に設定されています。
- アクセス間隔の設定は、表 6 に記載の「default_host_crawl_delay」及び「default_addr_crawl_delay」にて指定できます。

【身元情報の適切な通知について】

- クローラは Web サーバへアクセスする際に、「クローラ識別子」「メールアドレス」「ホームページ URL」を含む身元情報（User-Agent）を通知します。
- この情報は Web サーバのアクセスログに記録され、Web サーバの管理者が連絡を取ったり、クロールの目的を確認したりする際に参照されます。従いまして、利用者自身あるいは所属する組織などの情報を、お間違えのなきよう適切に記載して下さい。
- 身元情報の登録はインストール時に行われます。詳しくは、「4.2 NICT Web クローラのインストール」をご参照下さい。

3. 動作環境

NICT Web クローラの動作環境を以下に示します。

- OS
  - ✧ Linux 系システム
    ※動作保証 OS は CentOS 5.8 となります。
- メモリ
  - ✧ 256MB 以上の RAM
- ハードディスク
  - ✧ 1G 以上のハードディスク空き容量
    ※必要な空き容量は収集するページ数に依存します。目安としては、1 万ページで約 200MB の空き容量が必要となります。

4. インストール方法

4.1. 必須モジュール類のインストール

1. 動作環境 をインストールします。
   動作保証している OS は、CentOS5.8 となります。
   また以下の実行環境がインストールされていない場合は、個別にインストールして下さい。括弧内は動作確認済みのバージョンとなります。
   - ✧ python (2.4.3)
   - ✧ perl（5.8.8）

2. 以下のモジュールをインストールします。
   括弧内は動作確認済みのバージョンとなります。
   - ✧ mysql（5.0.77）
   - ✧ mysql-server（5.0.77）
   - ✧ zlib-devel（1.2.3）
   - ✧ GNU adns（1.2）
   - ✧ python-adns（1.1.1）
   - ✧ python2-chardet（2.0.1）

3. Perl で利用する以下のライブラリをインストールします。
   - ✧ URI（1.35）
   - ✧ HTTP::Date（1.47）
   - ✧ Log::Handler（0.76）
   - ✧ PerlIO::gzip（0.18）
   - ✧ DBI（1.601）
   - ✧ DBD::mysql（4.006）

4.2. NICT Web クローラのインストール

1. ダウンロードした NICT Web クローラのアーカイブを展開します。
   展開すると、NICT-Crawler フォルダが作成されます。

```
$ tar□xvfz□./NICT-Crawler_x.x.tar.gz
NICT-Crawler
```

※「□」マークは半角スペースを表します。
※「x.x」はバージョン番号を表します。

2. インストール時に登録する URL リストを編集します。
   NICT Web クローラは、リストに記載された URL を起点としてクロールを開始し、Web ページ内のリンクを辿って収集を行います。
   URL リストファイルは、以下のファイルとなります。

```
NICT-Crawler/conf.sample/initialize.urls
```

3. install.sh を実行します。

```
$ cd□NICT-Crawler/
$ sh□install.sh□<OPTION>

※「□」マークは半角スペースを表します。
```

なお、以下のオプションを指定することができます。

表 1 install.sh オプション

オプション	設定内容
--target-directory <インストールパス>	スクリプト類をインストールするディレクトリを指定します。指定が無い場合は、ホームディレクトリに crawler フォルダを作成してインストールします。
--initialize-urls <ファイルパス>	URL DB 初期化に使用する URL リストファイルを指定します。URL リストファイルは 1 行 1 URL で記述します。指定が無い場合は、同梱されている URL リストファイル(initialize.urls)を使用します。
--database-host <ホスト名>	URL DB に使用する MySQL のホスト名を指定します。localhost 以外の MySQL を使用する場合は、当該 MySQL に外部から root でアクセスできるよう事前に設定して下さい。なお、指定が無い場合は、localhost を使用します。
--need-database-rootpass <パスワード>	URL DB に使用する MySQL に root で

	アクセスする際、パスワードが必要な場合に指定します。
--overwrite	<インストールパス>先に何らかのファイルが既に存在する場合でも、エラー終了せずに強制的にインストールを行います。

4. インストール処理が実行されます。
必須モジュール及びライブラリの確認、各種ディレクトリの作成などを行います。
モジュール及びライブラリの確認で NG となった場合は、該当するモジュール及びライブラリをインストールした上で、再度 install.sh を実行して下さい。

なお、「クローラ識別子」「メールアドレス」「ホームページ URL」は、収集先の Web サーバに通知されるクローラの身元情報です。これらは収集先 Web サーバの管理者が参照することを想定した情報となっておりますので、利用者自身あるいは所属する組織などの情報を適切に記載して下さい。

インストール処理中に入力を求められる項目を以下に示します。

表 2 インストール処理入力項目

入力項目	設定内容
データ保存先ディレクトリ	収集したページやログの保存先ディレクトリを指定します。デフォルト設定で良い場合は、何も入力せずに Enter キーを押して下さい。
作業用ディレクトリ	収集処理中のデータの保存先ディレクトリを指定します。デフォルト設定で良い場合は、何も入力せずに Enter キーを押して下さい。
一時ディレクトリ	一時的に出力されるファイルの保存先ディレクトリを指定します。デフォルト設定で良い場合は、何も入力せずに Enter キーを押して下さい。
クローラ識別子	クローラを識別するための記号を指定します。例えば、株式会社 ABC が収集を行う場合は、「ABC_bot」などの他の

	クローラと区別可能な識別子を設定して下さい。
メールアドレス	収集責任者のメールアドレスを設定します。収集先 Web サーバの管理者が問い合わせなどの連絡を行う際に利用するため、適切なアドレスを記載して下さい。
ホームページ URL	収集実施主体の情報や収集の目的・概要などを記載した Web ページの URL を設定します。

```
$ [13/05/10 10:23:55] Checking mysql client ... OK
$ [13/05/10 10:23:55] Checking mysql connection ... OK
$ [13/05/10 10:23:55] Checking python command ... OK
$ [13/05/10 10:23:55] Checking python bsddb module ... OK
$ [13/05/10 10:23:55] Checking python profile module ... OK
$ [13/05/10 10:23:55] Checking python adns module ... OK
$ [13/05/10 10:23:55] Checking python chardet module ... OK
$ [13/05/10 10:23:55] Checking perl command ... OK
$ [13/05/10 10:23:55] Checking perl DBI module ... OK
$ [13/05/10 10:23:55] Checking perl URI module ... OK
$ [13/05/10 10:23:55] Checking perl HTTP::Date module ... OK
$ [13/05/10 10:23:55] Checking perl PerlIO::gzip module ... OK
$ [13/05/10 10:23:55] Checking perl Log::Handler module ... OK
$ [13/05/10 10:23:56] Checking perl Digest::MD5 module ... OK
$ [13/05/10 10:23:56] Check install directory ... done
Input keep data directory (/home/username/crawler/data): [データ保存先ディレクトリ]
Input crawl work directory (/home/username/crawler/work): [作業用ディレクトリ]
Input temporary directory (/home/username/crawler/tmp): [一時ディレクトリ]

Input crawler name (required) : [クローラ識別子]
Input contact mail : [メールアドレス]
Input contact homepage : [ホームページ URL]
$ [13/05/10 10:24:12] Install crawler script ... done
$ [13/05/10 10:24:12] Initialize URL Database ... done
$ [13/05/10 10:24:13] Insert initial URLs from /home/username/NICT-Crawler/conf.s
```

```
ample/initialize.urls  ...  done
```

5. インストール時に、下記フォルダ内に設定ファイルが自動で作成されますので、必要に応じて修正します。
   なお、${INSTALL_DIR}にはクローラのインストールディレクトリのパスが入ります。

```
${INSTALL_DIR}/conf
```

4.3. インストールフォルダ構成

インストール後のフォルダ構成は図 2 のようになります。

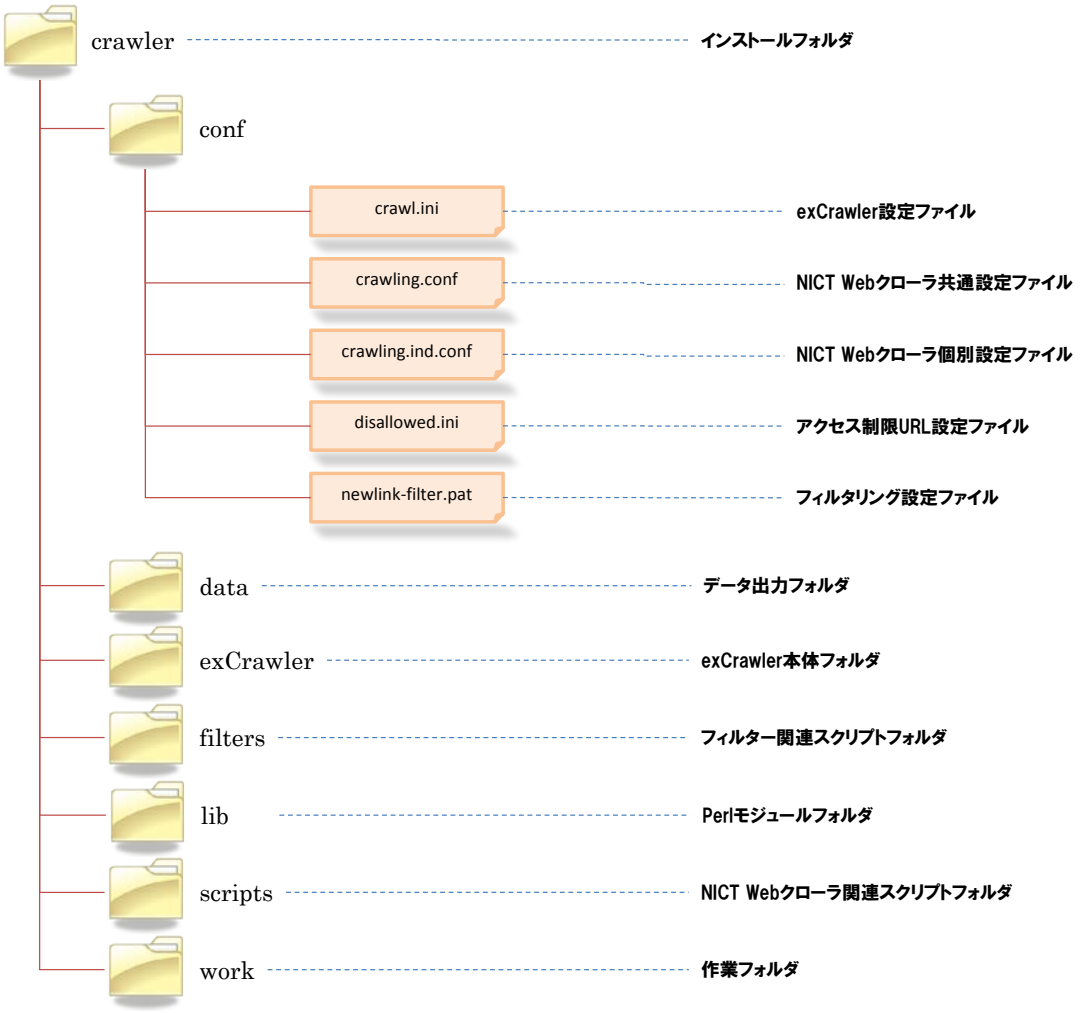

図 2 インストール後のフォルダ構成

5. 使用方法

5.1. クローラの実行方法

端末より、以下のコマンドで実行して下さい。

なお、${INSTALL_DIR}にはクローラのインストールディレクトリのパスが入ります。

```
$ sh□${INSTALL_DIR}/scripts/crawl-control-process.sh□¥
--com□${INSTALL_DIR}/conf/crawling.conf□¥
--ind□${INSTALL_DIR}/conf/crawling.ind.conf
```

※「¥」マークの個所は実際には改行せずに一行で記載します。

※「□」マークは半角スペースを表します。

表 3 実行時オプション

オプション	設定内容
--com <ファイルパス>	NICT Web クローラ共通設定ファイルを指定します。指定が無い場合は、スクリプトと同じディレクトリの crawling.conf を使用します。
--ind <ファイルパス>	NICT Web クローラ個別設定ファイルを指定します。本設定は必須です。

クローラは収集したページから抽出した URL を辿ることで Web ページを収集し続けます。また、収集済みの Web ページについても、推定した更新間隔に基づき再収集します。従いまして、クローラを終了する場合は、フォアグラウンド実行の場合は Ctrl-C で、バックグラウンド実行の場合は crawl-control-process.sh の実行プロセスに TERM シグナルを送信して下さい。

5.2. クローラの各種設定について

クローラの設定のうち、主要なものを以下に説明します。

- NICT Web クローラ共通設定ファイル（crawling.conf）

NICT Web クローラの基本設定を行う設定ファイルです。設定項目とその内容を表 4 に記載します。

表 4 crawling.conf 設定

設定項目	設定内容
DATABASE_HOST_URL	URL DB のホスト名又は IP アドレスを指定します。
DATABASE_NAME	データベース名を指定します。 （デフォルトは crawler）
DATABASE_USER	データベースのユーザを指定します。
DATABASE_PASS	データベースのパスワードを指定します。
SCRIPT_BASE	クローラのインストールディレクトリを指定します。
PATH_LOG_KEEP	データ出力フォルダのパスを指定します。 （デフォルトは${SCRIPT_BASE}/data）
FILE_DISALLOWEDLIST	exCrawler 用 Disallowed 設定ファイルのパスを指定します。
FILE_EXCRAWLCONFIG	exCrawler 用 共通設定ファイルのパスを指定します。
HTTP_PROXY	実行時に経由する HTTP プロキシのアドレスとポートを指定します。

- NICT Web クローラ個別設定ファイル（crawling.ind.conf）

NICT Web クローラの個別設定を行う設定ファイルです。設定項目とその内容を表 5 に記載します。

表 5 crawling.ind.conf 設定

設定項目	設定内容
NODE_CRAWLEXEC	クローラ実行ノードのホスト名を指定します。 （デフォルトは localhost）
REQUEST_HOSTLIMIT	1 回のクロールでアクセスを許可する、同一ホストへのリクエスト最大数を指定します。 なお同項目は、crawl.ini の default_host_request_limit に優先します。
REQUEST_ADDRLIMIT	1 回のクロールでアクセスを許可する、同一 IP アドレスへのリクエスト最大数を指定します。 なお同項目は、crawl.ini の default_addr_request_limit に優先します。
REQUEST_COUNT	1 回のクロールでアクセスする URL の件数を指定します。
PATH_CRAWL_OUTPUT	クローラ実行時の作業用ディレクトリを指定します。

	（デフォルトは${SCRIPT_BASE}/work）
EXECUTE_INTERVAL	クローラの実行間隔を秒単位で指定します。 指定が無い場合は、終了後すぐに次のクロールを実行します。

- exCrawler 設定ファイル（crawl.ini）

NICT Web クローラが利用している exCrawler モジュールのための設定ファイルです。設定項目とその内容を表 6 に記載します。

表 6 crawl.ini 設定

セクション	設定項目	設定内容
global	url_ext_filter	フィルタリング対象の拡張子を空白区切りで記載します。
resolv	max_concurrent_requests	DNS への同時リクエスト数を指定します。
collect	additional_headers	リクエストに付加されるヘッダを指定します。
	default_host_max_requests_per_connection	1 コネクション中のリクエスト数を指定します。2 以上を指定することで、キープアライブとなります。
	default_addr_max_requests_per_connection	1 コネクション中のリクエスト数を指定します。2 以上を指定することで、キープアライブとなります。
	default_max_response_size	レスポンスサイズの最大値を指定します。指定されたサイズ以上のレスポンスは破棄されます。
	default_host_request_limit	同一のホストに対して、1 回の起動中に発信できるリクエスト数の上限を指定します。
	default_addr_request_limit	同一のアドレスに対して、1 回の起動中に発信できるリクエスト数の上限を指定します。
	giveup_max_collect_time	1 回のクロールの最大実行時間を

		秒単位で指定します。
	default_host_crawl_delay	同一ホストに対するアクセス間隔を秒単位で指定します。
	default_addr_crawl_delay	同一アドレスに対するアクセス間隔を秒単位で指定します。
readr	output_encoding	ページファイルの出力エンコードを指定します。対応する文字コードは、「utf-8」「shift-jis」「euc-jp」「iso-2022-jp」など、Python の標準エンコーディングが指定できます。
	record_filter	フィルタリングの指定を行います。 ※この項目は編集しないで下さい。
checkdb	language	言語フィルタで採用する言語を指定します。現在は japanese 以外設定できません。
	drop_url	URL が指定された正規表現とマッチした場合、当該ページをフィルタリングします。 ※ここでのパターンは、Python の正規表現で指定して下さい。
	content_type	指定された正規表現にマッチするコンテンツタイプのみ収集します。 ※ここでのパターンは、Python の正規表現で指定して下さい。

5.3. フィルタリング機能の使い方

NICT Web クローラのフィルタリング機能には、URL を対象とするものや、Web ページのコンテンツを対象とするものなどがあります。以下に各フィルタリング機能を説明します。

- クロール禁止リスト

${INSTALL_DIR}/conf/disallowed.ini

クロール禁止対象のホストあるいは IP アドレスを記載します。記載例を図 3 に示します。

```
;; disallowed.ini
[collect]

disallowed_hosts:
    sample.com
    disallowed.com
disallowed_addrs:
    123.123.123.123
```

図 3 クロール禁止リスト記載例

- リンクフィルタ

${INSTALL_DIR}/conf/newlink-filter.pat

ダウンロードした Web ページのリンクから抽出した URL を対象とします。リンクフィルタには、以下の 2 つの機能が含まれています。

1. パターンフィルタ機能
   指定された正規表現にマッチする URL を収集対象から排除し、URL DB に登録しません。
2. リプレース機能
   指定された正規表現を用いて URL の置換処理を行います。セッション ID などの不要文字列を削除して URL DB に登録します。

各機能の設定書式を表 7 に、記載例を図 4 に示します。

表 7 newlink-filter.pat 設定書式

機能	書式
パターンフィルタ	f□<正規表現>
リプレース	c□<正規表現>□<出力フォーマット>

```
f□¥.(?:m4a|m4e|mid|midi|mp1|mp3)$
f□^http://sample¥.jp/home¥?

c□(.*)PHPSESSID=[^&]*(.*)□$1$2
c□(.*)jsessionid==[^&]*(.*)□$1$2
```

図 4 newlink-filter.pat 記載例

なお、ここで指定する正規表現は、Perl の正規表現に準じます。

- 辞書フィルタ

```
${INSTALL_DIR}/filters/sexual.dic
```

辞書に含まれる単語が一定種類以上出現した場合にフィルタリングを行います。現在は、出現数や辞書ファイルなどは固定設定となっており、基本的にアダルト系 Web ページをフィルタリングするために使用しています。

- その他のフィルタリング処理

上記以外のフィルタリング処理を以下に示します。

1. URL に含まれる文字列に対して、正規表現にてフィルタリングを行います。
   →crawl.ini [drop_url] 参照
2. content-type に対して、正規表現によるフィルタリングを行います。
   →crawl.ini [content_type] 参照
3. 取得した Web ページの言語によりフィルタリングを行います。
   →crawl.ini [language] 参照

5.4. 収集データについて

収集されたデータは、crawling.conf の「PATH_LOG_KEEP」にて指定されたフォルダ内の下記ファイル内に保存されます。ここで「YYYYMMDDhhmmss」は、クロールを開始した時刻（年月日時分秒）を表します。

```
${ PATH_LOG_KEEP }/YYYYMMDDhhmmss_0_0/page.gz
```

上記ファイルには、1 回のクロールで収集された全ページ情報が 1 ファイルとして出力されます。フォーマットは下記の形式で 1 つのレコードを構成し、1 ページ分の情報は BEG レコードから始まり、END レコードにより終了します。

:::<TAG>:::□<ContentsSize>□<Contents>¥r¥n

TAG の種類と内容を表 8 に記載します。

表 8 TAG 一覧

タグ	内容
BEG	ページの開始を表すタグ。 <Contents> は常に空であるため、<ContentsSize>は 2 バイト固定。
RAR	リクエストリスト文字列
URL	ページ取得を行った URL
RES	レスポンスステータス
RBT	対応する robots.txt の URL
TIM	リクエストの読出、ページ取得の開始、終了時刻
REQ	実際に送信した HTTP リクエスト
STA	サーバから返信された HTTP ステータス
HDR	サーバから返信された HTTP ヘッダ
RAW	サーバから返信されたページ本体
CON	UTF-8 にてエンコードされたページ本体
LNK	ページから抽出されたリンク URL LNK レコードは抽出されたリンクの数だけ存在し、またリンクが存在しない場合は、本レコードも存在しない。
ANC	ページから抽出されたリンク URL 及びアンカーテキスト ANC レコードは抽出されたリンクの数だけ存在し、またリンクが存在しない場合は、本レコードも存在しない。
END	ページの終了を表すタグ <Contents> は常に空であるため、<ContentsSize> は 2 バイト固定。

6. 参考文献

- 独立行政法人 情報通信研究機構 知識処理グループ 情報信頼性プロジェクト：情報分析システム WISDOM ―Web の健全な利活用を目指して―, 2011 年 3 月 31 日.

■問い合わせ先

〒619-0289
「けいはんな学研都市」京都府相楽郡精華町光台 3-5
独立行政法人情報通信研究機構 ユニバーサルコミュニケーション研究所
企画室内 高度言語情報融合フォーラム事務局
E-mail: info@alagin.jp
TEL:0774-98-6304
FAX:0774-98-6955

以上