

ツール

項番	ツール名	配置場所 as Role	機能概要	機能説明	入力	出力	実装言語	使用するOSS/ Pythonモジュール
①	全文検索システム	Server A	着目する話題を記述したインターネットリソースの巡回と、その全文の収集	①インターネットリソースを巡回するクローラーと、②全文検索エンジンからなる。	巡回するURLの指定	⑤Elasticsearch Repository	Java	FESS elasticsearch
②	<a href="#">collect</a>	Server B	全文検索システムの新着記事の抽出	①全文検索システム から Get Request over HTTP により前回処理した以降の巡回記事を取り出し、②list.txt ファイルに登録済の記事を除外し、③結果を HTML フォーマットで書き出す。	前回処理以降の日数	④新着記事の抽出結果 yyyymmdd.html	Ruby	—
③	Webブラウザ	Server B	新着記事抽出結果の確認	①新着記事抽出ツールの抽出結果を確認し、②人間が有為と判断した記事をショートカットファイル(url)に書き出す。	新着記事抽出結果のブラウズ	bookmark(ショートカット形式)	不要	Firefox / Chrome
④	<a href="#">dir2list</a>	Server B	ショートカット群とショートカットリストの間の相互変換	ファイルシステムに階層的に展開されたショートカット群(dir)と、1ファイルからなるショートカットリスト(list.txt)の間の相互変換を行う。	①ショートカット群(dir)のルートディレクトリパス名	①ショートカットリスト(list.txt)ファイル名	Ruby	—
	<a href="#">list2dir</a>				①ショートカットリスト(list.txt)ファイル名	①ショートカット群(dir)のルートディレクトリパス名		
⑤	<a href="#">dir2keyword</a>	Server B	ショートカット群付属のキーワードファイルとキーワードリストの間の相互変換	ファイルシステムに階層的に展開されたショートカット群ディレクトリ(dir)にあるキーワードファイル(keywords.txt)と、1ファイルからなるキーワードリスト(keywords.txt)の間の相互変換を行う。	①ショートカット群(dir)のルートディレクトリパス名	①キーワードリスト(keywords.txt)ファイル名	Ruby	—
	<a href="#">keyword2dir</a>				①キーワードリスト(keywords.txt)ファイル名	①ショートカット群(dir)のルートディレクトリパス名		
⑥	<a href="#">crawl</a>	Server B	ショートカット群をクロウルし、その指し示すコンテンツをローカルファイルにダウンロード	①ファイルシステムに階層的に展開されたショートカット群(dir)をクロウルし、②その指し示すコンテンツを(ショートカットのディレクトリ名に.crawledを付加したディレクトリ上の)ローカルファイルにダウンロードする。	①ショートカット群(dir)のルートディレクトリパス名	②ショートカットのディレクトリ名に.crawledを付加したディレクトリ上のローカルファイル(HTML または PDF)	Ruby	—
⑦	<a href="#">merge keywords</a>	Server B	キーワードリストの同義語リスト化	1ファイルからなるキーワードリスト(keywords.txt)から同義語リストを生成する。	①キーワードリスト(keywords.txt)ファイル名	②同義語リスト(synonyms.txt)	Ruby	—
⑧	<a href="#">html2plaintext</a>	Server B	ダウンロードコンテンツのブレインテキスト化	ダウンロードコンテンツからHTMLのタグ情報などを取り去りブレインテキスト化する。テキストは形態素解析ツールによって形態素に分解し、その区切りは"/"などにより明示しておく。 (現時点ではまだ⑤同義語リストは参照していないが、将来形態素のエンジンに使用予定)	③ショートカットのディレクトリ名に.crawledを付加したディレクトリ上のローカルファイル(HTML または PDF)	④ショートカットのディレクトリ名に.plaintextを付加したディレクトリ上のローカルファイル(形態素分解済みテキスト)	Python	janome / mecab / juman++
⑨	<a href="#">digest</a>	Server B	ブレインテキスト化したコンテンツを適当な長さに要約する	ブレインテキスト化した形態素分解済みテキストを、オンプレミスまたはクラウドの分類処理にかけられる分量になるように要約する。	④ショートカットのディレクトリ名に.plaintextを付加したディレクトリ上のローカルファイル(形態素分解済みテキスト)	⑤ショートカットのディレクトリ名に.digestを付加したディレクトリ上のローカルファイル(要約済みテキスト)	Python	gensim (Doc2Vec)
⑩	bookmarks	Server C	ファイルシステムに階層的に展開されたショートカット群を Tree View 形式で表示する	ファイルシステムに階層的に展開されたショートカット群を treeview.js によって扱い、WebブラウザからTreeView形式で見えるようにする。	①ショートカットリスト(list.txt)ファイル名	Webブラウザでブラウズ	Ruby (Ruby on Rails)	treeview.js
⑪	classify	Server B	収集した記事を分類・タグ付けする	オンプレミスまたはクラウドの分類処理により、収集した要約済み記事を分類・タグ付けする。	⑥ショートカットのディレクトリ名に.digestを付加したディレクトリ上のローカルファイル(要約済みテキスト)	①ショートカット群(dir)のルートディレクトリパス名	Python	要検討
①	<a href="#">timestamp</a>	Server B	手動で行っていた④新着記事の抽出結果から①ショートカット群のディレクトリに振り分ける処理を半自動化する	手動で階層のない①todayディレクトリに配置したショートカットのタイムスタンプをショートカットが指すコンテンツの最終更新日時にする。最新更新日時が取得できなかったものは、手動で日時を修正しなければならない。	①todayディレクトリに配置したショートカット	同左	Ruby	—
②	<a href="#">pickup keywords</a>	Server B		①todayディレクトリに配置したショートカットのファイル名からキーワードを抽出し、そのキーワードに対応した振り分け先候補のパス名列挙したリストを生成する。候補が適切でなかったものは手動で修正しなければならない。	①todayディレクトリに配置したショートカット	各ショートカットの振り分け先候補のパス名列挙したリスト (①todayディレクトリの②list.txt)	Ruby	—
③	<a href="#">copy url</a>	Server B		各ショートカットの振り分け先候補のパス名列挙したリスト (①todayディレクトリの②list.txt)をもとに、ショートカットを①ショートカット群(dir)の振り分け先にコピーする。	各ショートカットの振り分け先候補のパス名列挙したリスト (①todayディレクトリの②list.txt)	①ショートカット群(dir)の振り分け先	Ruby	—

Role の異なるサーバを物理的に同じサーバ上に配置してもよい

実装なし  
インストールと設定のみ