

# Simple Document-by-Document Search Tool “Fuwatto Search” Using Web API

Masao Takaku<sup>1</sup> and Yuka Egusa<sup>2</sup>

<sup>1</sup> University of Tsukuba  
1-2 Kasuga, Tsukuba, Ibaraki, Japan  
`masao@slis.tsukuba.ac.jp`

<sup>2</sup> National Institute for Educational Policy Research  
3-2-2 Kasumigaseki, Chiyoda, Tokyo, Japan  
`yuka@nier.go.jp`

**Abstract.** In this paper, we propose a new search method *Fuwatto Search* that allows users to retrieve documents in a document-by-document manner via a Web API. We present an implementation of the proposed method (i.e., Fuwatto CiNii Search), which targets the CiNii Article database, one of the largest academic article databases in Japan. The experimental evaluation of Fuwatto CiNii Search with newspaper articles demonstrates the retrieval effectiveness of 0.25 for precision at 10 and 0.17 for mean average precision.

**Keywords:** document retrieval, Web API, effectiveness, CiNii Articles.

## 1 Introduction

Digital libraries, domain-specific databases, and Web search engines are important for our daily lives. When we need certain information for health, fashion, travel, and many other areas, we typically use Web-based search services.

Most search services support keyword-based queries. In other words, we must specify keywords that are appropriate for our information needs. Keyword-based queries can cause a gap between users and search systems because users do not always have enough knowledge on how to express appropriate keywords that correspond to their information needs and database content.

One solution for this issue is to use another querying method, i.e., the document-as-a-query method. A search system that supports document-by-document queries accepts an existing document as a query and returns a list of similar documents. Currently, some news websites, electronic commerce websites, and digital library services support this function. On such websites, a user can obtain a list of other items that are similar to the current item on that page. Such content-based similarity search functions can be beneficial in certain situations. However, these functions are not always available for all databases.

In this paper, we propose a new content-based similarity search methodology, “Fuwatto Search.” Fuwatto Search allows users to search any arbitrary search service with a document-by-document query via a simple keyword-based search Web API.

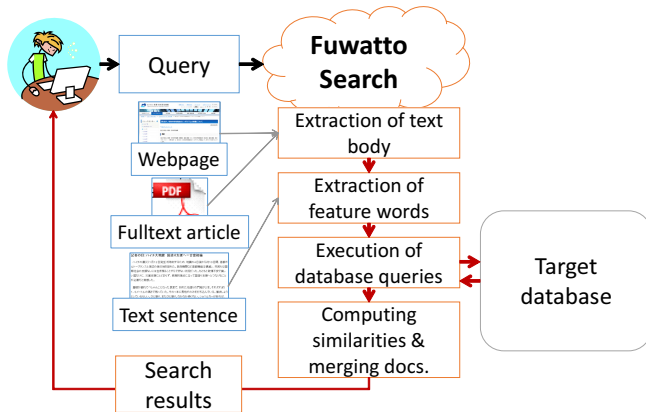


Fig. 1. Overview of Fuwatto Search

## 2 Proposed Methodology

An overview of the proposed *Fuwatto Search* methodology is shown in Figure 1.

Fuwatto Search is a document-by-document search method for remote databases. Fuwatto Search can search target databases that do not support document-by-document search functions. It iteratively accesses the keyword search functionality of a target database, and then analyzes and re-ranks the retrieved results. This process is described as follows.

### 1. Extraction of full text

Fuwatto Search accepts PDF, HTML, and plain text data as a query. When a PDF or HTML page is given to Fuwatto Search as a query, the system first extracts the text content. For web pages written in HTML, the page header, footer and navigational elements are excluded. The system extracts the main body elements (i.e., the text) of an HTML page. We use the `extractcontent.rb` text extraction tool to extract main body elements from an HTML page [1].

### 2. Extraction of feature words with weights

Extracted words are segmented using a Japanese morphological analyzer (i.e., MeCab). We simultaneously count the term frequency ( $TF$ ) in the query document.  $TF$  is multiplied by the term occurrence cost, which is pre-defined in the MeCab dictionary.

### 3. Search execution

We select the top  $n$  words from a vector of the feature words (computed by the previous process). Each of the top  $n$  words is sent as a query to the target database to determine if there is a match. If no hits are returned for a given word, we exclude that word from the set of feature words. We then tests the next feature word using the same process.