

複数カテゴリを持つ時系列データ集合の 効果的な情報の可視化

曾原寿允[†] 堀幸雄[†] 今井慈郎[†]

RSS リーダや Twitter に見られるような、カテゴリ分けされた大量の時系列データを閲覧する機会が増えている。これらの情報を俯瞰的に閲覧し、必要な情報を素早く入手することを目的とするインタフェースを提案する。

Effective information visualization on time series data set with multi categories

Toshimitsu Sohara[†] Yukio Hori[†] Imai Yoshiro[†]

The chance to inspect a large amount of categorized time series data shown in the RSS reader and Twitter has increased. We propose an interface to aim to inspect these information in the down shot, and to acquire necessary information quickly.

1. はじめに

近年のインターネット技術の進歩により、ユーザはインターネットを通して大量の情報を得ることができるようになった。特に、Google リーダー[1]のような RSS リーダや Twitter[2]に見られるような、カテゴリ分けされた大量の時系列データを閲覧する機会が増えている。ここでいうカテゴリとは、例えば、RSS リーダの場合は RSS フィードを提供している各サイト、Twitter の場合では各個人を表す。つまり、カテゴリ分けされ時系列データとは、複数の情報元から時系列にそってデータを集合した情報を表している。

これらの情報は一般的な利用場面では、各カテゴリの情報をひとまとめにし、時系列でソートされ、一つの画面内においてリスト形式で閲覧されることが多い。ユーザはこれらの情報を新しい情報、もしくは未読で古い情報から順に一つ一つ目を通して自分に必要な情報を見つけていく。しかし、未読の情報が増加した場合、大量の情報の中から必要な情報を取捨選択することは、ユーザの負担となり貴重な時間を浪費することになりかねない。

本研究では、これらの情報を俯瞰的に閲覧し、必要な情報を素早く手に入れることを目的とするインタフェースを提案する。具体的な特徴としては以下ようになる。

- (1) 1画面内でのカテゴリごとのグリッド表示
- (2) ユーザによる情報元の重要度付け

また、より汎用的に利用できるようにインタフェースの実装をフレームワークとして提供できるように設計を行う。

本稿では、第2節にて研究背景として、既存のウェブサービスのインタフェースや新しく世に出てきているウェブサービスを紹介する。次に第3節において本研究にて提案するインタフェースの概要を説明し、同時に実装例を示す。そして、第4節にて今後の課題を述べる。

2. 研究背景

2.1 既存ウェブサービスにおけるインタフェース

Google リーダー (図1) や Twitter (図2) に見られるような、複数の情報元からの時系列データの閲覧には、時系列でソートされたリスト形式の項目を順に読んでくインタフェースが採用されている。このインタフェースでは以下の利点と欠点が挙げられる。

[†] 香川大学
Kagawa University



図 1 Google リーダー



図 2 Twitter

利点

- 時系列でならんでいるので、閲覧方法が直感的である
- 全ての情報がリスト内にあるのでカテゴリを選択し直す動作が不要である

欠点

- 不必要な情報にも順に目を通さないといけない
- 全ての情報に目を通そうとすると相当な時間がかかり負担が大きくなる
- 情報が大量になった場合現実的な情報閲覧方法として破綻する

2.2 新しいインタフェースを用いたウェブサービス

Webテクノロジーの発展により、ブラウザ上でよりリッチなコンテンツを作成することが可能になってきている。上記のサービスを改善するために Good Noows[3], feedly[4]などの RSS 情報を効果的に提示するウェブサービスも登場している。どちらのサービスも RSS フィードを閲覧するための、RSS リーダーである。

Good Noows では、提供される情報のレイアウトをカスタマイズすることができる。用意されているレイアウトは、格子状にならべたり左上の記事だけ大きく表示したりするものがある。ユーザはリスト形式以外のレイアウトで閲覧することが可能で、使っていてより楽しく便利なサービスとなっている。

feedly も同様にフィードの情報を雑誌のようにレイアウトし効果的に情報の提示を行っている。また、重要度の高い情報のみを抽出して提示する機能もある。しかし必ずしもユーザの欲しい情報が抽出されるわけではなく、本当に必要としている情報が抜け落ちてしまう可能性があると考えられる。

3. インタフェースの概要

3.1 提案インタフェースの特徴

提案インタフェースのイメージを図 3 に示す。

本研究で提案するインタフェースの特徴を以下に示す。

(1) 1画面内でのカテゴリごとのグリッド表示

本研究で提案するインタフェースは、1画面内で大量の情報を俯瞰的に閲覧することを目的としているため、カテゴリごとに時系列で並べたまとまりをひとまとまりとして、それらのまとまりを1画面内にグリッド形式で配置をする。ユーザは画面の遷移をすることなく大量の情報に目を通すことができると考えられる。

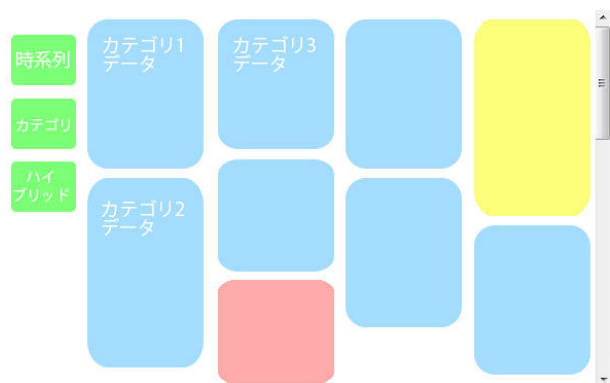


図 3 提案インターフェースのイメージ図

(2) ユーザによる情報元の重要度付け

ユーザによる情報元の重要度付けとして以下の2点を行うことができる。

● ユーザの操作によるカテゴリのソート

ユーザは自分のお気に入りのフィード提供サイトや個人を、閲覧インターフェース内でドラッグ&ドロップにより表示の順番の入れ替えを行うことができる。一度上位に移動された提供元の情報はサーバーに保存され、次に新しい情報を得た時には優先的に上位に表示されるよう設定することが可能である。

● カテゴリのカラーリング

ユーザの手によってカテゴリごとに色の設定を行う。カテゴリと同様に、新しい情報を得た時にはユーザの指定したカラーにて表示され、ユーザは一目で、自分のお気に入りの情報元からの情報を見つけることが可能になる。

これらのユーザによる情報元の重要度付けを行う利点は、一度設定しておけば、新しい情報が提供されたときに、ユーザは意識して画面内を走査せずに、自分にとって重要度の高い情報を見つけることが可能になる点が挙げられる。

また、このインターフェースの実装において、より汎用的に利用できるよう、情報の閲覧インターフェースのフレームワークとして作成を行う。つまり、カテゴリ分けされた時系列情報であれば、元情報とインターフェースの間を取り持つ専用のミドルウェアを用意すれば統一のインターフェースで閲覧できるというものである。具体例として Twitter クライアントの実装について 3.3 節で説明を行う。

3.2 インタフェースを用いたシステム実装

インタフェースを構成するシステムの概要を図 4 に示す。

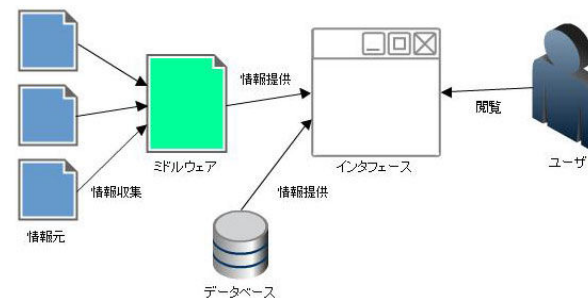


図 4 システムの概要図

(1) 情報元

ユーザに情報を提供する情報元。各ウェブサイトが提供する RSS フィードや、Twitter における個人のつぶやきがこれに該当する。これらの情報は各サービスによって様々な形式で提供されている。例えば、様々な WebAPI などで提供されている、XML, Atom, json などがある。これらの形式の違いによる差異は(2)で説明するミドルウェアで単一の形式に変換することにより吸収される。

(2) ミドルウェア

情報元から得た情報をひとまとめにし、データ構造の変換を行い、インターフェースに提供を行う。このミドルウェアにおけるデータ構造の変換は、情報元のデータ形式にあった結びつけが必要になる。そのため、各情報元に対応したプログラムの作成が必要になる。

インターフェースに提供する情報の形式として本研究では Atom を採用した。Atom はウェブ上の各種コンテンツを配信するための XML 文書フォーマット及びコンテンツの編集を行なうための通信プロトコルなど幾つかの仕様群の総称である。本稿では特に Atom とは、ウェブサイトの更新情報等のメタデータやコンテンツの配信 (Syndication)、保存 (Archive) を受け持つ XML 文書の仕様である Atom Syndication Format を指すものとする。

Atom を採用した理由は以下の点があげられる。

- 標準化が進められている仕様である
- RSS フィードなどで広く採用されており、データ形式の変換が最小限ですむ
- 本研究で扱うカテゴリ分けされた時系列データを扱うのに優れている

Atom では以下に示す 3 つの要素を含んでいる必要がある。

- 一意識別子
- タイトル
- タイムスタンプ

さらに、提案するインタフェースで必要となる情報として以下を採用した。

- 著者
- 元情報へのリンク
- サマリー
- カテゴリを識別するアイコン
- 画像

(3) インタフェース

ミドルウェアから受け取った Atom 形式のデータをパース、整形しユーザに情報を提供する。ウェブブラウザ上でのインタフェースを想定しているため、プログラムにより HTML を生成しページを構成する。また、カテゴリのグリッド表示、ソートや色づけなどには JavaScript を用いた。

(4) データベース

各情報元から取得した情報の他に、ユーザごとの基本情報を格納する。格納する情報としては、ユーザの閲覧履歴、ソート順や色付け情報を保存する。

3.3 Twitter での実装例

本研究で提案するインタフェースを用いた Twitter クライアントの実装を行った。

Twitter よりユーザのタイムラインを取得し、取得したデータを変換するプログラムを作成することにより、提案するインタフェースを用いて、Twitter の特定ユーザのタイムラインを閲覧することができる。Twitter では、特定ユーザのタイムラインを取得するには認証が必要になるため、ミドルウェアのプログラムに認証を行う処理を記述する。

以下に Twitter クライアントの実装過程で作成したプログラムの機能を示す。

(1) ユーザ認証の実行

タイムライン取得のためにユーザ認証を行う。今回は OAuth 認証により認証を行うプログラムを作成した。

(2) データの取得

ユーザの認証情報をもとに、TwitterAPI を使用し特定ユーザのタイムラインを取得する。TwitterAPI では取得できるデータ形式として xml 形式と json 形式を提供している。ここでは、json 形式にてデータを取得した。

(3) データ形式の変換

取得した json 形式のデータを元に Atom 形式へデータ形式の変換を行う。



図 5. Twitter での実装例

以上の過程により作られたデータをインタフェース部に取り込む。実際に取得して提案インタフェースにて表示した画面を図 5 に示す。

4. おわりに

本稿では、複数カテゴリを持つ時系列データ集合の効果的な情報の可視化を行うインタフェースについて、概要の説明と Twitter クライアントとしての実装を行った。今後の課題として、実際にユーザに利用してもらい評価実験を実施したい。

謝辞 本研究を進めるにあたり、ご指導を頂いた今井慈郎先生、堀幸雄先生に感謝致します。

参考文献

- 1) Google リーダー, <http://www.google.co.jp/reader/>
- 2) Twitter, <http://twitter.com/>
- 3) Good Noows, <http://goodnoows.com/>
- 4) feedly, <http://www.feedly.com/>