

## 時系列データの特徴をビジュアルの類似と相違の変化の比較で捉える — RankTimeChart と RankTimeTable の使い方編 —

松原 伸人

matubara@sra.co.jp

### ◆ はじめに

RankTimeChart は、複数の時系列な数値データ系列を折れ線グラフで表示します。

RankTimeTable は、複数の時系列な数値データ系列をランキングで表示します。

両方とも、日付を指定すると、その日における数値データの大きさに基づいて上位 10 系列に、順位に応じた色をつけて表示します。

RankTimeChart は、データ全体の変化と、上位 10 系列のその日に至るまでの変化や、その日から先の変化を比べてみる事ができるというものです。

RankTimeTable は、上位 10 系列がデータの全期間の中のどこあたりに現れるのかが見て取れます。

**[GSLetterNeo Vol.109 時系列データの特徴をビジュアルの類似と相違の変化の比較で捉える]**では、GSLetterNeo シリーズで用いられている月ごとの単語数のデータを例にして 2 つの Web アプリケーションプログラム を紹介しました(図 1、図 2)<sup>1</sup>。

今回は、RankTimeChart と RankTimeTable の使い方を紹介します。

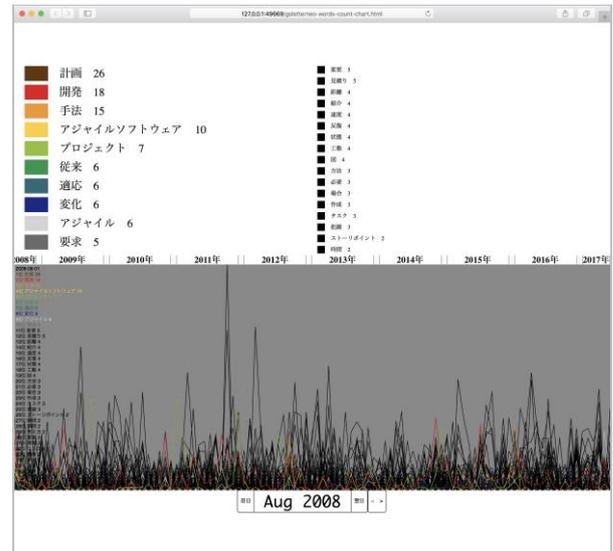


図 1 RankTimeChart

ビデオ 1 [月別の GSGLetterNeo 記事内で用いられている単語数と推移](#)

リンク 1 [単語の出現数 top10 の推移](#)

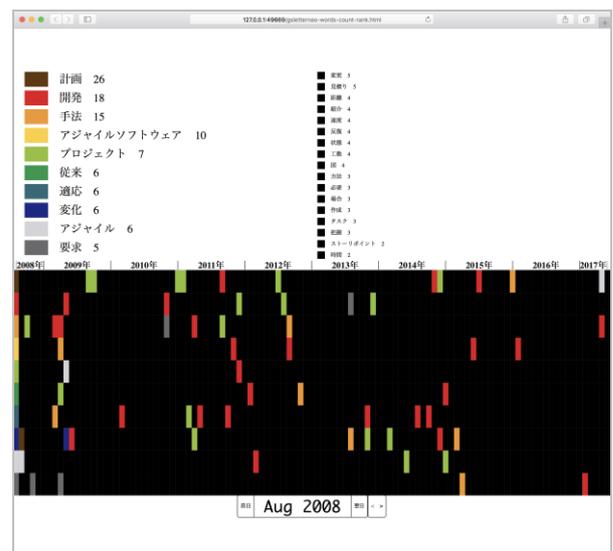


図 2 RankTimeTable

ビデオ 2 [月別の GSGLetterNeo 記事内で用いられている単語数のランキングの推移](#)

リンク 2 [単語の出現数の順位の推移](#)

<sup>1</sup> ビデオとリンクへの URL もご参照ください。

## ◆ プログラムのダウンロード

RankTimeChart と RankTimeTable は HTML と JavaScript と CSS で書かれている Web ブラウザ上で動くアプリケーションプログラムです。

特別な開発ツールはインストールしなくても、プログラムを実行するための Web ブラウザ と、プログラムを書くテキストエディタがあれば使用できます。

実際のところ [リンク 1] と [リンク 2] は、HTML と JavaScript と CSS で書かれています。

例えば [リンク 1] の HTML ファイル `gsletterne-o-words-count-chart.html` と、この中で参照しているファイル群をダウンロードすれば、手元のマシン上で実行できるようになっています。

これらのファイル群をまとめてダウンロードして実行できるようにした zip ファイルを置いておきました。

**リンク 3** [ranktimechartandtable-2017-0904.zip](#)

`ranktimechartandtable-2017-0904.zip` を解凍したフォルダの中には、`RankTimeChart.html` と `RankTimeTable.html` と `lib` フォルダが入っています。(図 3)

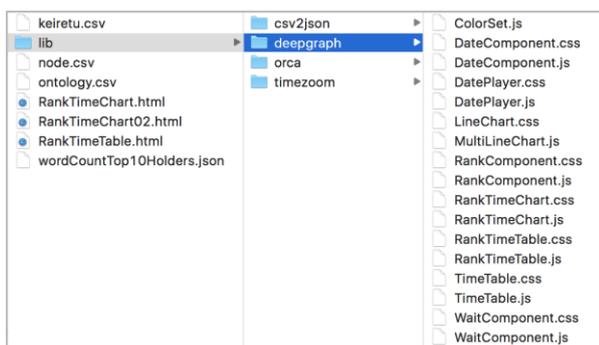


図 3 ファイル構成

`lib` フォルダ には、`RankTimeChart` と `RankTimeTable` の JavaScript で書いたプログラムと関連プログラムおよびスタイルシートが入っています。

`RankTimeChart.html` と `RankTimeTable.html` を Web ブラウザで開いて動かします。

## ◆ プログラムの実行手順

実行手順は、`RankTimeChart.html` または `RankTimeTable.html` を自分のデータを読み込むように書き換え、Web ブラウザで開く、となります。

`RankTimeChart.html` と `RankTimeTable.html` にはそれぞれ、データファイルを読み込んで表示するプログラムが書いてあります。(図 4、図 5)

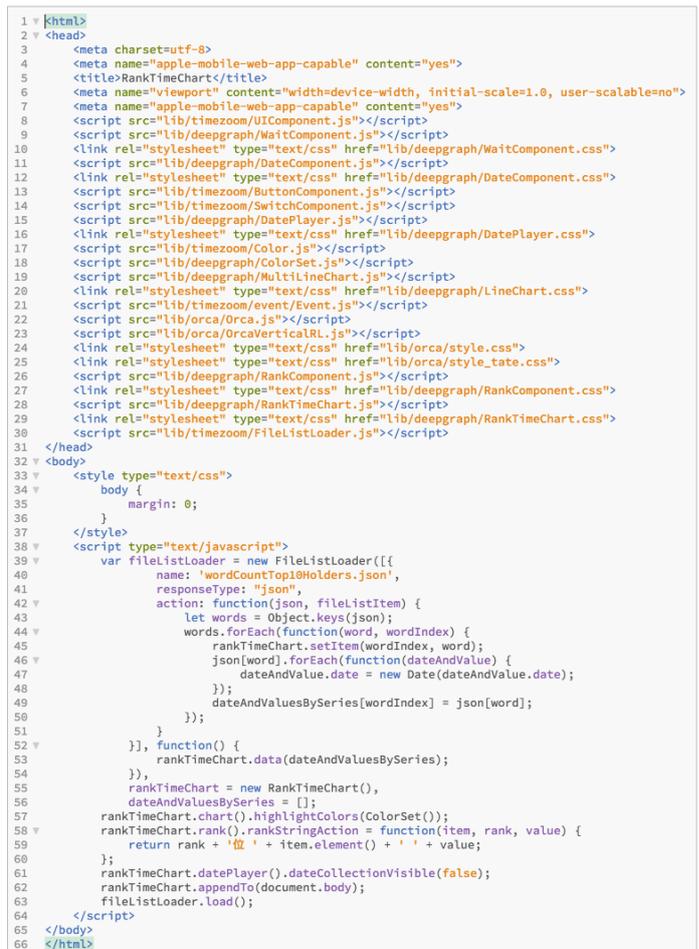


図 4 RankTimeChart.html

```

1 <html>
2 <head>
3   <meta charset=utf-8>
4   <meta name="apple-mobile-web-app-capable" content="yes">
5   <title>RankTimeTable</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
7   <meta name="apple-mobile-web-app-capable" content="yes">
8   <script src="/lib/timezoom/UIComponent.js"></script>
9   <script src="/lib/deepgraph/WaitComponent.js"></script>
10  <link rel="stylesheet" type="text/css" href="/lib/deepgraph/WaitComponent.css">
11  <script src="/lib/deepgraph/DateComponent.js"></script>
12  <link rel="stylesheet" type="text/css" href="/lib/deepgraph/DateComponent.css">
13  <script src="/lib/timezoom/ButtonComponent.js"></script>
14  <script src="/lib/timezoom/SwitchComponent.js"></script>
15  <script src="/lib/deepgraph/DatePlayer.js"></script>
16  <link rel="stylesheet" type="text/css" href="/lib/deepgraph/DatePlayer.css">
17  <script src="/lib/timezoom/Color.js"></script>
18  <script src="/lib/deepgraph/ColorSet.js"></script>
19  <script src="/lib/deepgraph/TimeTable.js"></script>
20  <link rel="stylesheet" type="text/css" href="/lib/deepgraph/TimeTable.css">
21  <script src="/lib/timezoom/event/Event.js"></script>
22  <script src="/lib/orca/Orca.js"></script>
23  <script src="/lib/orca/OrcaVerticalRL.js"></script>
24  <link rel="stylesheet" type="text/css" href="/lib/orca/style.css">
25  <link rel="stylesheet" type="text/css" href="/lib/orca/style_tate.css">
26  <script src="/lib/deepgraph/RankComponent.js"></script>
27  <link rel="stylesheet" type="text/css" href="/lib/deepgraph/RankComponent.css">
28  <script src="/lib/deepgraph/RankTimeTable.js"></script>
29  <link rel="stylesheet" type="text/css" href="/lib/deepgraph/RankTimeTable.css">
30  <script src="/lib/timezoom/FileListLoader.js"></script>
31 </head>
32 <body>
33   <style type="text/css">
34     body {
35       margin: 0;
36     }
37   </style>
38   <script type="text/javascript">
39     var fileListLoader = new FileListLoader([
40       name: 'wordCountTop10Holders.json',
41       responseType: 'json',
42       action: function(json, fileListItem) {
43         let words = Object.keys(json);
44         words.forEach(function(word, wordIndex) {
45           rankTimeTable.setItem(wordIndex, word);
46           json[word].forEach(function(dateAndValue) {
47             dateAndValue.date = new Date(dateAndValue.date);
48           });
49           dateAndValuesBySeries[wordIndex] = json[word];
50         });
51       }
52     ]), function() {
53       rankTimeTable.data(dateAndValuesBySeries);
54     },
55     rankTimeTable = new RankTimeTable(),
56     dateAndValuesBySeries = [];
57     rankTimeTable.timeTable().highlightColors(ColorSet());
58     rankTimeTable.timeTable().showingCursor(false);
59     rankTimeTable.datePlayer().dateCollectionVisible(false);
60     rankTimeTable.appendTo(document.body);
61     fileListLoader.load();
62   </script>
63 </body>
64 </html>

```

図 5 RankTimeTable.html

8 行目から 30 行目で、JavaScript のプログラム群とスタイルシート群を読み込んでいます。

つづく 33 行目から 37 行目のスタイルは、ページ全体の上下左右のマージンを 0 にセットしています。

37 行目以降の `<script>` タグの中にある JavaScript のプログラムが、データを読み込んで、RankTimeChart または RankTimeTable にデータをセットして表示するコードです。

RankTimeChart と RankTimeTable は同じデータモデルになっているので、1 個データを用意すれば、両方に同じデータを使用できます。

そのため HTML ファイルの `<script>` タグの中に書いてあるコードもほぼ同じ処理で、擬似コード的に書くと次のような流れになっています。

1. データファイルを読み込む FileListLoader を作る [39 行目から 54 行目]
2. RankTimeChart または RankTimeTable を作る [55 行目]
3. データの入れ物として配列を用意する [56 行目]
4. 作成した RankTimeChart または RankTimeTable を設定する [RankTimeChart.html 57 行目から 61 行目、RankTimeTable.html 57 行目から 59 行目]
5. ページに RankTimeChart または RankTimeTable を追加する [RankTimeChart.html 62 行目、RankTimeTable.html 60 行目]
6. データを読み込む [RankTimeChart.html 63 行目、RankTimeTable.html 61 行目]

## ◆ 1. データファイルを読み込む

FileListLoader は、XML ファイル や JSON ファイル や TXT ファイル を大量に読み込むような時に使うように作成しているプログラムです。内部的には XMLHttpRequest を使っています。

ファイルのリストをセットすると、全ファイルを読み込みます。ファイル個別にファイル読み込み終了時に実行するコードをセットしたり、全ファイルを読み終わった時に実行するコードをセットしたりできるようになっています。

この例では 1 個の wordCountTop10Holders.json という JSON ファイル を読み込んでいだけなので、FileListLoader を使う代わりに XMLHttpRequest を使うとか、FileReader を使うとか、input タグを組み合わせてファイルダイアログからファイルを選択できるようにするとかしてもいいかもしれません。

今回の例では、json ファイル を json 形式 のデータ構造にしたデータを action で指定した関数に渡すようになっています。

この関数では、読み込んだ json データ から、単語表記のリストを作って、リストのインデックスを setIt

em 関数 でふり、また、日付を表す文字列を JavaScript の Date オブジェクト に変換し、データの入れ物である dateAndValuesBySeries にセットしています。

データを読み終わったら、52 行目にあるように、RankTimeChart または RankTimeTable に読み込んだデータをセットします。

## ◆ 別のデータの作り方例

RankTimeChart と RankTimeTable は時系列データ群を扱います。

時系列データと系列名をインデックスで管理しています。

時系列データは、日付 (date) と数値 (value) をプロパティに持つオブジェクトのリストです。

サンプルの wordCountTop10Holders.json の末尾の "オントロジー" を例にすると、"オントロジー" が系列名で、時系列データは 2 個あります。

"オントロジー" が 2012 年 5 月に 1 回、2017 年 7 月に 5 回使われたことを示しています。(図 6)

```
"オントロジー": [
  {
    "date": "2012-05-01T00:00:00.000Z",
    "value": 1
  },
  {
    "date": "2017-07-01T00:00:00.000Z",
    "value": 5
  }
]
```

図 6 wordCountTp10Holders.json の抜粋

もっと大量のデータを扱うような場合は、系列ごとの時系列データファイル群と、系列名のリストファイルを作成するような方法も考えられます。

例えば時系列ファイルと系列名のファイルを CSV で作成します。

"オントロジー" の時系列データ ontology.csv と "ノード" の時系列データ node.csv と系列名のリストファイル keiretu.csv の場合、次のようにかけます。(図 7、8、9)

```
date,value
2012-05-01T00:00:00.000Z,1
2017-07-01T00:00:00.000Z,5
```

図 7 ontology.csv

```
date,value
2017-03-01T00:00:00.000Z,3
2017-06-01T00:00:00.000Z,7
```

図 8 node.csv

```
オントロジー
ノード
```

図 9 keiretu.csv

FileListLoader で読み込む場合、次のようにかけます。(図 10)

```
39 ▾ function dateAndValues (text) {
40   let dateAndValues = csv2json.csv.parse(text);
41   dateAndValues.forEach(function (dateAndValue) {
42     dateAndValue.date = new Date(dateAndValue.date);
43     dateAndValue.value = parseInt(dateAndValue.value);
44   });
45   return dateAndValues;
46 }
47
48 ▾ var fileListLoader = new FileListLoader([[
49   name: 'ontology.csv',
50   responseType: "text",
51   action: function(text, fileListItem) {
52     dateAndValuesBySeries.push(dateAndValues(text));
53   }
54 ], {
55   name: 'node.csv',
56   responseType: "text",
57   action: function(text, fileListItem) {
58     dateAndValuesBySeries.push(dateAndValues(text));
59   }
60 }, {
61   name: 'keiretu.csv',
62   responseType: "text",
63   action: function(text, fileListItem) {
64     let names = text.split("\n");
65     names.forEach(function (name, index) {
66       rankTimeChart.setItem(index, name);
67     });
68   }
69 }], function() {
70   rankTimeChart.data(dateAndValuesBySeries);
71 });
72 rankTimeChart = new RankTimeChart(),
73 dateAndValuesBySeries = [];
```

図 10 読み込むプログラム例

## ◆ 4. 作成した RankTimeChart または RankTimeTable を設定する

highlightColors は、上位 10 項目の色を設定します。CSS の色の書き方で、順位順で 10 色のリストを指定します。

ranksStringAction は、ランク表に表示する文字列を設定します。

datePlayer().dateCollectionVisible は、画面下部にある日付を指定するコンポーネントのうち、日のリストの表示／非表示を切り替えます。

showingCursor はグラフや表の上に順位を表示する／しないを切り替えます。

## ◆ 終わりに

vol.109 および vol.110 で紹介した RankTimeChart と RankTimeTable および関連プログラムの一部は、JST 科学技術振興機構 の CREST プログラム の CREST・さきがけ複合領域 における データ粒子化による高速高精度な次世代マイニング技術の創出 (研究代表者 宇野 毅明 情報・システム研究機構 国立情報学研究所 情報学プリンシプル研究系) の成果です。

URL <http://www.jst.go.jp/kisoken/crest/project/45/14531617.html>

GSLetterNeo Vol. 110  
2017 年 9 月 20 日発行  
発行者 ●株式会社 SRA 先端技術研究所  
編集者 ●土屋正人

バックナンバーを公開しています ●<http://www.sra.co.jp/gsletter>  
ご感想・お問い合わせはこちらへお願いします ●[gsneo@sra.co.jp](mailto:gsneo@sra.co.jp)

夢を。



**株式会社SRA**

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。Yawaraka Innovation  
やわらかいのバージョン