

# Web クリッカーシステムへのテキストマイニング機能実装の試み

## Implementation of Text Mining in a Web-based Clicker System

古賀 掲維\*, 野崎 剛一\*  
Aoi KOGA\*, and Koichi NOZAKI\*

長崎大学\*  
Nagasaki University

Web クリッカーシステムはスマートフォンやタブレットなど様々な端末から利用でき、さらに専用の端末を用いるクリッカーシステムでは収集が難しい記述式の回答を収集できるというメリットがある。記述式の質問は様々な意見を収集したい場合や選択肢を用意するのが難しい場合に便利であるが、得られたデータを整理する必要があり、すぐに結果を把握するのが難しいというデメリットがある。本論文では、Web クリッカーシステムにテキストマイニング機能を実装することにより、記述式の回答の把握を容易にしたクリッカーシステムの構築の試みについて報告する。キーワード：テキストマイニング、クリッカーシステム、スマートデバイス

The web-based clicker system is available for mobile devices such as a smartphone or tablet. Furthermore, the web-based clicker system can collect description answers, but it is difficult for the clicker system with exclusive devices. When we want to collect various opinions or when it is difficult to prepare for choices, it is convenient to use a description question. However, we cannot grasp the abstract of the answers immediately because it is necessary to analyze answers when we use a description question. In this paper, we report a web-based clicker system which can grasp description answers by implementation of text mining.

Keyword: Text Mining, Clicker System, Smart Device

### 1. はじめに

文部科学省中央教育審議会が2012年8月に取りまとめた「新たな未来を築くための大学教育の質的転換に向けて～生涯学び続け、主体的に考える力を育成する大学へ～（答申）」の公表以降、各大学においてアクティブ・ラーニング(能動的学習)をどのように進めていくか様々な実践が試みられている。

アクティブ・ラーニングを実現する手法は様々なあるが、比較的容易にアクティブ・ラーニングを導入できるツールとしてオーディエンス・レスポンス・システム(通称：クリッカー)が注目されている。クリッカーは教師と学生間の双方向コミュニケーションを可能とするシステムであり、アメリカの大学では広く普及している。一般的なクリッカー(以下、クリッカー専用機と呼ぶ)を利用する際に必要となるのは、専用のソフトウェア(プラグイン)をインストールし、USB レシーバーを挿したパソコンと、専用のリモコン端末である。クリッカー専用機利用時には、学生にリモコ

---

\* ICT 基盤センター  
〒852-8521 長崎市文教町1-14  
Center for Information and Communication  
Technology  
〒852-8521 1-14, Bunkyo-machi, Nagasaki  
E-Mail : amnesia@nagasaki-u.ac.jp

ン端末を配布した上で、教師がパソコンで操作を行い、学生がスクリーンに映し出された質問にリモコン端末を用いて回答するという流れになる。回答結果はすぐにスクリーンに表示され、学生参加型の授業を手軽に実現できるというのがクリッカー専用機の大きなメリットである。しかしながら、クリッカー専用機には以下のように幾つかのデメリットも存在する。

1. 同時に利用する授業の数だけクリッカー専用機のセット(1セット数十万)を用意する必要がある。
2. 授業に参加する学生の数だけリモコン端末を用意する必要がある。
3. リモコン端末を授業の開始時に学生に配布し、終了時に回収する必要がある。

クリッカー専用機を利用したアクティブ・ラーニングを大規模に導入する場合、1, 2は予算上の障害となる可能性がある。また、3は教師にとって大きな負担となり、配布や回収が面倒だからクリッカー専用機を利用したくないという意見も聞かれた。そこで、筆者らはクリッカー専用機のデメリットを解消し、より身近にアクティブ・ラーニングを導入できるツールの提供を目指してスマートデバイスを端末として利用する Web クリッカーシステムの開発を行った。筆者らが開発した Web クリッカーシステムの特徴は以下の通りである。

1. スマートフォンやタブレットなどを端末として利用できるため、教師が端末を準備し配布や回収を行う必要がない。
2. Web ブラウザがあれば動作し、利用に際して特殊なソフトウェアやアプリのインストールは不要。
3. Web アプリケーションとして作成しているので、同時に利用できる授業数、学生数に制限がない。

筆者らが開発した Web クリッカーシステムは、クリッカー専用機のデメリットの多くを解消しており、筆者らが所属する長崎大学で

はクリッカー専用機に代わるツールとして期待されている。

さらに、スマートデバイスに対応した Web クリッカーシステムの副次的なメリットとして記述式の回答を収集することができるという点をあげることができる。長崎大学で導入されているクリッカー専用機ではリモコン端末の仕様から記述式の回答を入力させることが不可能だったのが、スマートデバイスを利用することによって簡単に記述式の回答を収集することが可能となった。記述式の質問は広く意見を収集したい場合や、選択肢を用意するのが難しい場合に便利である。また、記述式の回答には学生の生の意見が込められており、有意義な情報を得られることが多い。記述式の質問が利用できるようになり、クリッカー活用の範囲が大きく広がったといえる。しかしながら、記述式の質問にもデメリットがある。実際に記述式の質問を利用してみるとわかるのだが、記述式の回答は選択式の回答と違って単純に集計できず、内容を把握するためには、一通り回答内容に目を通す必要がある。これはクリッカーの即時性を損なう場合が多く、利用者からも改善を望む意見があがっていた。そこで筆者らが着目したのが、自然文の解析に用いられるテキストマイニング技術である。テキストマイニング技術を利用することによって、記述式の回答を定量的なデータとして扱うことが可能となる。

本論文では、Web クリッカーシステムにテキストマイニング機能を実装することにより、記述式の回答の把握を容易にしたクリッカーシステム構築の試みについて報告するものである。

## 2. Web クリッカーシステムの概要

本章では、筆者らが開発を行っている Web クリッカーシステムの概要について説明する。テキストマイニング機能の実装については第3章で説明する。本システムの基本的なアイ

デアや仕組みは筆者らが 2013 年に開発したシステム<sup>2)</sup>を継承しているが、テキストマイニング機能の実装に合わせてシステムを一新している。

## 2.1 システムの開発・運用環境

本システムのアーキテクチャとしては運用コスト、可搬性を考慮し LAPP(Linux + Apache + PostgreSQL + PHP)を採用している。オペレーティング・システム(OS)は CentOS 6.6 を採用し、Web サーバーおよびデータベース管理システムは OS 付属の Apache 2.2 と PostgreSQL 8.4 を使用した。PHP については、OS 付属のものが PHP 5.3.3 と古かったため、Remi レポジトリ<sup>2)</sup>から PHP 5.4 をインストールして使用した。

システムの開発環境としては、開発効率やメンテナンス性を考慮し、サーバサイドおよびクライアントサイドともにフレームワークを使用している。サーバサイドのフレームワークとして Symfony<sup>3)</sup> 2.5 を、クライアントサイドのフレームワークとして Sencha ExtJS<sup>4)</sup> 5.1 と Sencha Touch<sup>4)</sup> 2.4 を採用した。クライアントサイドのフレームワークについては、クリッカーの設定やパソコンからの接続時には Sencha ExtJS を、スマートフォンやタブレットからの回答時にはスマートフォンにより特化した Sencha Touch を利用するように Web ブラウザを判別して使い分けている。

## 2.2 システムの利用環境

本システムを利用するために必要となるのは HTML5 に対応した Web ブラウザのみである。なお、スマートフォンやタブレット等のブラウザを利用する場合、デバイス毎の Viewport のサイズに注意する必要がある。本システムでは、設定、質問の編集および結果の閲覧などの教師利用の場合は横幅 768 ピクセル以上、アンケートへの回答など学生利用

の場合は横幅 320 ピクセル以上の Viewport サイズを想定している。

## 2.3 アクセスキー

本システムの大きな特徴のひとつにアクセスキーを用いた回答方式がある。利用方法について説明する前にアクセスキーについて解説しておく。本システムでは、学生は常に同一の URL に接続し、教師から提示されたアクセスキーと呼ぶ 4~6 桁の数字を入力して回答画面に移動するようになっている。アクセスキーを入力すればアンケートの選択等は不要で、すぐに教師が指定したアンケートに回答することができる。図 1 はアクセスキーを用いた回答の手順を示したものである。

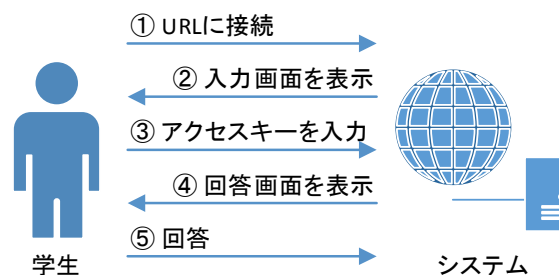


図 1 アクセスキーを用いた回答の手順

本システムが採用しているアクセスキー方式には以下のようなメリットがある。

1. 常に同一の URL に接続すればよいので、URL をブックマークやお気に入りに登録したり、二次元バーコードで提供したりすることが容易である。
2. 単純なアクセスキーを入力するだけですぐに回答を開始できるので、クリッカーの即時性を損なわない。

なお、アクセスキーについてはアンケートの回答開始時にシステムがユニークな値を割り当て、アンケートの終了時に解放するようにしている。また、アンケートが回答可能状態で放置された場合への対応としてアクセスキーには有効期限を設けており、期限を過ぎたアクセスキーは自動的に解放される仕組みとしている。アクセスキーの有効期限は桁数によって 3 日~30 日としている。

## 2.4 システムの利用方法

本節では、システムの利用方法について簡潔に説明する。図2は本システムの利用の手順を示したものである。

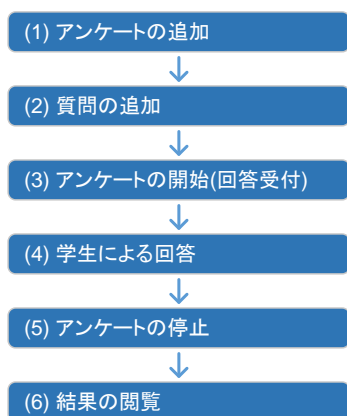


図2 システム利用の手順

(1)～(3), (5), (6)は教師が実行する手順, (4)は学生が実行する手順となる。以下, それぞれの手順について説明する。

### (1) アンケートの追加

図3は教師が初めてシステムにアクセスした場合の画面である。



図3 システムアクセス時の画面

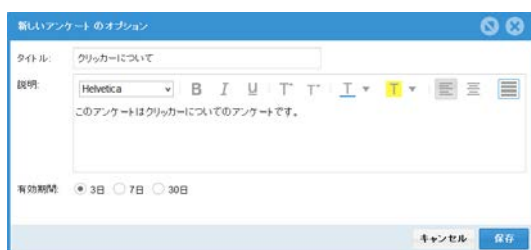


図4 アンケートのオプションダイアログ

図3の右上の丸い点線で囲んだ+ボタンをクリックすると図4のようなアンケートのオプションダイアログが表示される。ダイアログでは、アンケートのタイトル、説明および有効期間を設定することができる。

### (2) 質問の追加

図5はアンケートが追加された画面である。アンケートのタイトルバー上の丸い点線で囲んだペンのボタンをクリックすると図6のようなアンケート編集画面に移動する。



図5 アンケート追加後の画面



図6 アンケート編集画面

図6の右上の丸い点線で囲んだ+ボタンをクリックすると図7のような追加する質問の種類を選択するダイアログが表示される。



図7 質問の追加ダイアログ

利用できる質問の種類は、多肢選択式、複数選択式、二択式、記述式、短答式の5種類である。短答式は改行なし、記述式は改行ありのテキストで回答する質問としている。

種類を選択すると図8のような質問のオプションダイアログが表示され、質問のタイトル、質問文、選択肢などを設定することができる。



図 8 質問のオプションダイアログ

### (3) アンケートの開始

図 9 のようなアンケート欄のボタンの中にある「開始」ボタンをクリックすることによってアンケートを開始することができる。

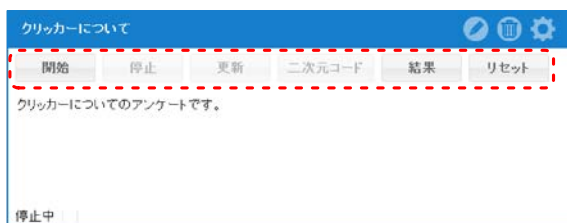


図 9 アンケートのボタン

アンケートを開始すると「二次元コード」ボタンが利用可能となり、二次元コードとアクセスキーを画面に表示できる。図 10 は画面に表示された二次元コードの例である。



図 10 二次元コード画面

### (4) 学生による回答

学生は、二次元コードから回答画面に接続し、二次元コードの上に表示されているアク

セスキーを入力して回答を開始する。図 11 はアクセスキー入力画面である。



図 11 アクセスキー入力画面

### (5) アンケートの停止

図 9 の「停止」ボタンをクリックすることによって、アンケートを停止することができる。アンケートを停止するとアクセスキーが解放され学生は回答できなくなる。

### (6) 結果の閲覧

図 9 の「結果」ボタンをクリックすると、実施したアンケートの結果を確認することができる。図 12 は回答の集計結果をグラフ表示したものである。



図 12 回答の集計結果のグラフ

結果画面では、タブによってグラフの種類を切り替えたり、回答データを表示したりすることができる。

### 3. テキストマイニング機能の実装

本章では、本システムへのテキストマイニング機能の実装について詳述する。

#### 3.1 形態素解析エンジンについて

本システムでは、テキストマイニング機能を実現するための形態素解析エンジンとしてオープンソースソフトウェアの MeCab<sup>6)</sup> を利用している。また、MeCab で用いる辞書には MeCab の公式ページで推奨されている IPA 辞書<sup>6)</sup> を採用した。なお、MeCab のインストールにあたってはオープンソースの全文検索エンジン Groonga<sup>7)</sup> のパッケージを利用した。図 13 は MeCab をインストールする際のコマンドである。

```
# rpm -ivh http://packages.groonga.org/centos/
groonga-release-1.1.0-1.noarch
# yum install -y mecab mecab-ipadic mecab-devel
```

図 13 MeCab のインストール

#### 3.2 PHP 用 MeCab ライブラリ

PHP から MeCab を利用するためのライブラリとして php-mecab<sup>8)</sup> を利用した。php-mecab のインストールは、PECL(PHP Extension Community Library)を用いて行った。図 14 は php-mecab をインストールする際のコマンドである。

```
# pear channel-discover pecl.opendogs.org
# pear install opendogs/mecab-0.5.0
```

図 14 php-mecab のインストール

図 15 は php-mecab を利用したプログラムの例である。

```
1: <?php
2: $mecab = new MeCab_Tagger();
3: $s = 'MeCabを利用して形態素解析を行う';
4: echo implode(', ', $mecab->split($s));
5: echo "\n<br /><br />\n";
6: $node = $mecab->parseToNode($s);
7: while($node){
8:   echo $node->getFeature();
9:   echo "\n<br />\n";
10:  $node = $node->getNext();
11: }
```

図 15 プログラムの例

図 15 のプログラムでは、「MeCab を利用して形態素解析を行う」という文章を 4 行目の split というメソッドを用いて形態素に分解し、さらに 8 行目の getFeature というメソッドで各形態素の情報を取得している。図 16 はこのプログラム実行結果である。

```
1: MeCab,を,利用,し,て,形態素,解析,を,行う
2:
3: BOS/EOS,*,*,*,*,*,*,*
4: 名詞,一般,*,*,*,*,*
5: 助詞,格助詞,一般,*,*,*,を,ヲ,ヲ
6: 名詞,サ変接続,*,*,*,*,利用,リヨウ,リョー
7: 動詞,自立,*,*,サ変・スル,連用形,する,シ,シ
8: 助詞,接続助詞,*,*,*,*,て,テ,テ
9: 名詞,一般,*,*,*,*,形態素,ケイタイソ,ケイタイソ
10: 名詞,サ変接続,*,*,*,*,解析,カイセキ,カイセキ
11: 助詞,格助詞,一般,*,*,*,を,ヲ,ヲ
12: 動詞,自立,*,*,五段・ワ行促音便,基本形,行う,オコナウ,オコナウ
13: BOS/EOS,*,*,*,*,*,*,*
```

図 16 プログラムの実行結果

図 16 の実行結果の 1 行目が文章を形態素に分解した結果となっており、カンマ区切りで各形態素が出力されている。4~12 行目は各形態素の情報となっており、この情報から品詞や活用前の原型を知ることができる。

#### 3.3 テキストマイニング機能の実装

テキストマイニング機能の実装にあたっては、クリッカーの即時性を損ねないように、また、テキストマイニングについての専門的知識がないユーザーでも簡単に利用できるよう留意し、以下のような実装を行った。

1. テキストマイニングの対象は短答式および記述式質問の回答データとする。
2. 対象とする品詞は名詞および動詞とし、名詞および動詞として分類された形態素の出現回数をカウントする。
3. 動詞については形態素の情報から品詞の原型を取得してカウントするものとする。
4. テキストマイニング機能は結果画面の回答データ欄に設けた「名詞」、「動詞」タブから確認できるようにする。



図 17 はテキストマイニング機能を実装する前、図 18 はテキストマイニング機能を実装した後の画面である。

番号	回答	回答日時
1	忘れる	2015/02/18 09:56
2	忘れたらヤバイ	2015/02/18 09:57
3	忘れてしまう	2015/02/18 09:58
4	メモ帳なくしたら全部わからなく...	2015/02/18 09:58

図 17 テキストマイニング機能実装前

番号	回答	回答日時
1	忘れる	2015/02/18 09:56
2	忘れたらヤバイ	2015/02/18 09:57
3	忘れてしまう	2015/02/18 09:58
4	メモ帳なくしたら全部わからなく...	2015/02/18 09:58

図 18 テキストマイニング機能実装後

実装前は、単に回答データを一覧で表示する「回答データ」というタブのみであったが、実装後は、「名詞」、「動詞」というタブを追加し、各品詞の集計を簡単かつすぐに実行できるようにした。

タブがクリックされたときシステム内部では図 19 のような処理を行っている。

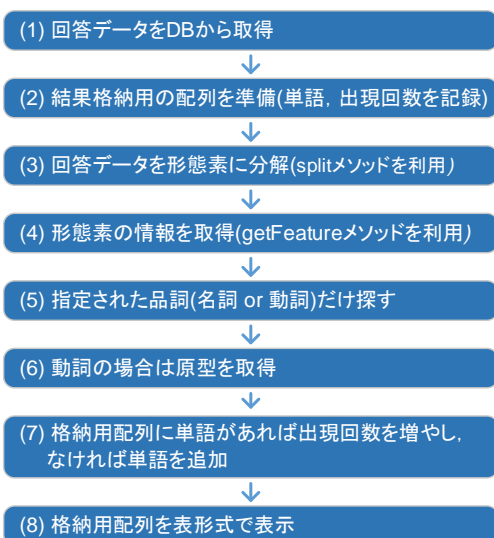


図 19 システム内部の処理の流れ

なお、形態素解析処理等の実行タイミングは、システムの負荷を考慮し結果ページの読み込み時ではなくタブのクリック時に行うようにしている。

### 3.4 テキストマイニング機能の利用例

筆者らが担当する情報系科目で行ったアンケートに対して本システムのテキストマイニング機能を利用した場合の例を示す。記述式のアンケートを利用して「あなたのパスワードの管理方法のデメリットを教えてください。」という質問を作成した。学生から得られた回答は約 50 件であった。

図 20, 21 はそれぞれ「名詞」、「動詞」タブをクリックして処理を実行し、出現回数順にソートした結果である。なお、システム上は全ての結果が表示されるが図 20, 21 では 7 項目までキャプチャしている。

名詞	出現回数
パスワード	14
こと	9
性	7
場合	6
可能	5
ログイン	4
すべて	4

図 20 名詞タブの集計結果

動詞	出現回数
忘れる	34
する	18
しまう	17
なる	13
ある	12
できる	8
終わる	6

図 21 動詞タブの集計結果

図 20 の名詞の集計結果をみると「パスワード」が最も多く、パスワードに関連した「ログイン」という単語も利用されているの

がわかる。図 21 の動詞の集計結果をみると「忘れる」が最も多く、学生がパスワードを忘れてしまうことが一番のデメリットだと考えていることがわかる。ただし、名詞、動詞ともに集計結果だけをみても意味が把握しにくい単語も含まれており、これについては今後改善が必要である。

なお、今回の例ではテキストマイニング機能の実行にかかった時間は1秒未満で、瞬時に集計結果を得ることができた。このことから、一般的な利用においてはクリッカーの即時性を損なうことなく活用できるものと考えられる。

#### 4. まとめ

本論文では、筆者らが開発を行っている Web クリッカーシステムにテキストマイニング機能を実装する試みについて報告した。

クリッカーの即時性を損なわず、かつ簡単に利用できるよう留意してテキストマイニング機能の実装を行った結果、これまですぐに把握することが難しかった記述式質問の回答を瞬時に把握できるシステムを構築することができたものと考えている。本機能の実装により、これまで授業中などの短い時間で瞬時に把握することが難しかった記述式回答のデータをある程度把握できるようになったので、授業などでさらに本システムの活用が進むものと考えている。

また、今後の予定としては、以下のような改善を行っていきたいと考えている。

1. 形態素解析時の解析オプションの追加
2. 形態素からの原文表示機能の追加
3. 一覧表示でのフィルタ機能の追加
4. ユーザー辞書利用機能の追加
5. 係り受け解析機能の追加

#### 参考文献

- (1) 古賀掲維, 柳生大輔, 野崎剛一: “スマートデバイスでの利便性を追求した Web

システム版クリッカーの開発”, 大学情報システム環境研究, VOL.16, pp.43–50 (2013-07)

- (2) Les RPM de Remi - Repository :

<http://rpms.famillecollet.com/>

- (3) Symfony :

<http://symfony.com/>

- (4) Sencha ExtJS / Touch :

<http://www.sencha.com/>

- (5) MeCab :

<http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>

- (6) IPA 辞書 :

<http://sourceforge.jp/projects/ipadic/>

- (7) Groonga :

<http://groonga.org/>

- (8) php-mecab :

<https://github.com/rsky/php-mecab/>

※URL は全て 2015 年 2 月 13 日確認

#### 著者略歴



**古賀掲維** 1994 年長崎大学工学部卒業, 1996 年同大学院工学研究科構造工学専攻修了, 1996 年 4 月長崎大学工学部助手, 2002 年 4 月同大学教育機能開発センター講師, 2006 年 3 月同助教授, 2007 年 4 月同准教授, 2011 年 5 月同情報メディア基盤センター准教授, 2014 年 4 月 ICT 基盤センター准教授, 修士 (工学).

**野崎剛一** 1975 年九州大学工学部卒業, 同年 4 月長崎大学助手, 1980 年同講師 1986 年 4 月~1987 年 1 月テネシー州立大学 客員研究員, 2003 年長崎大学総合情報処理センター助教授, 2005 年同情報メディア基盤センター教授, 2014 年 4 月 ICT 基盤センター教授, 博士 (工学).