コーパスコンコーダンサ 『ChaKi.NET』 のプロジェクト機能

浅原 正幸 (国立国語研究所) *
森田 敏生 (総和技研)

Project Functions on 'ChaKi.NET'

Masayuki Asahara (NINJAL)
Toshio Morita (Sowa Research Co.,Ltd.)

要旨

本稿では、ChaKi.NET の新しい機能であるプロジェクト機能を紹介する。プロジェクト機能は複数のテキストを可視化する機能である。文単位でアラインメントされたテキスト対を相互に可視化することが可能である。発表では、

- BCCWJ-Trans (BCCWJ に対する対訳付与) の複数言語の可視化
- BCCWJ 長単位・短単位の可視化
- BCCWJ の読み時間の可視化 (テキスト出現順と、視線走査順の 2 種類の分析)

についてデモを行う。

1. はじめに

本稿ではコーパスコンコーダンサ 『ChaKi.NET』 (Matsumoto et al. (2005)) の新しい機能であるプロジェクト機能について解説する。

コーパスに対して様々なレベルのアノテーションが施されている。形態論情報・係り受け構造を基本として、言語学的に多様なレベルのアノテーションを重ね合わせるためのデータ形式拡張 CaboCha フォーマット (松吉ほか (2014b)) が提案されている。このフォーマットでは、アノテーションを文字列範囲・リンク（有向・無向）・同値類に抽象化した。ChaKi.NET はこの 3 つの種類のアノテーションを可視化する機能を有している。

一方、複数のレイヤーのテキストからなるコーパスなどが整備されつつある。例えば『現代日本語書き言葉均衡コーパス』(以下 BCCWJ; Maekawa et al. (2014)) は長単位と短単位の 2 つの形態論情報を保持しており、基底となる形態論情報を 2 つ有している。自然言語処理の分野では機械翻訳のための訓練データ・評価データとして対訳コーパスが整備されている。また歴史コーパスや方言コーパスの場合、現代語訳や標準語訳が整備されることが考えられる。

ChaKi.NET のプロジェクト機能は 2 つ以上のレイヤーからなるコーパスを格納し、その 2 つのレイヤーを可視化する機能である。以下では、プロジェクト機能の概要について解説するとともに、活用例として、対訳コーパス・BCCWJ の長単位/短単位・読み時間の可視化などについて紹介する。

* masayu-a@ninjal.ac.jp

2.　プロジェクト機能の概要

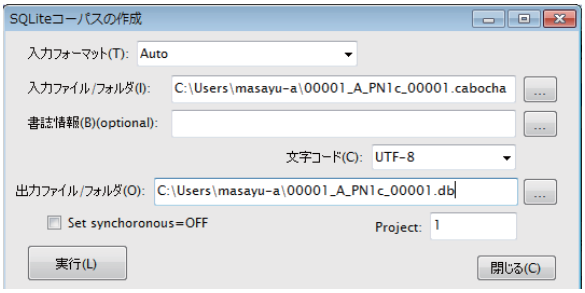
ChaKi.NET は形態素情報を格納する word テーブルや各アノテーションタグを階層化することができるプロジェクト (Project) という概念が存在する。デフォルトでは全ての要素が ID=0 のプロジェクトに存在するが、このデフォルトプロジェクト以外のプロジェクトを作成することにより、形態論情報やアノテーションをプロジェクト毎にグルーピングすることができる。

2.1　プロジェクトを指定したデータの格納・検索・可視化

以下ではプロジェクト機能の利用方法について概説する。

2.1.1　プロジェクトを指定したデータの格納

まずプロジェクトを指定したデータの格納方法について述べる。通常のデータ作成方法と同様に拡張 CaboCha フォーマット (松吉ほか (2014b)) もしくは CoNLL-U フォーマット(1)のデータを準備する。[ツール]→[SQLite コーパスの作成] よりコーパス作成の画面を立ち上げ、[出力ファイル/フォルダ] に既存の sqlite db ファイルを指定する。右下の [Project:] の値をデフォルトの 0 以外の値を指定することにより、既存のデータと別のレイヤーのデータを格納することができる。



2.1.2　プロジェクトを指定したデータの検索

メインツールバーの右端にある “Proj” 欄に Project ID を指定すると、検索時に Project ID に合致する結果のみを得ることができる。

検索結果に対して DependencyEdit を行うとき、検索に用いた Project ID、すなわちその結果が属している Project ID が DependencyEdit に伝えられる。その DependencyEdit において行われる編集 (アノテーションタグの追加・削除等）は、その Project に対して行われる。

2.1.3　2 画面モード

メニュー [表示] → [KWIC 画面を分割] により、KWIC View が上下 2 画面に分割される。

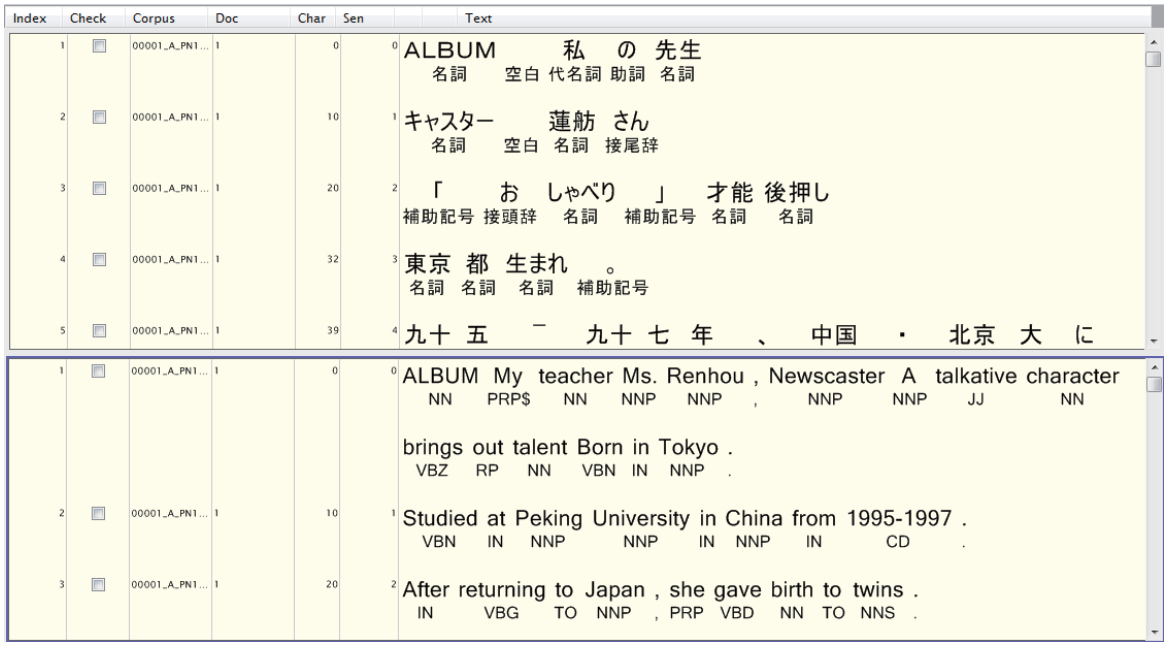
2 画面モード時に上下いずれかの KWIC View をクリックすると青い縁取りが現れる。これは、その View がカレント View であることを示す。検索コマンドの結果はカレント View に表示されるので、カレントを切り替えながら異なる検索条件で検索を行うと、検索結果同士を

(1) http://universaldependencies.github.io/docs/format.html　但し、ChaKi.NET は CoNLL-U フォーマットのうち Multiword token 表現については対応していない。

上下画面で比較することができる。

真中のスプリッター（分割線）をマウスドラッグすることにより分割位置を調整できます。元の 1 画面表示に戻すにはもう一度同じコマンドを実行する。

典型的な使い方として、上の View に Project=0 の検索結果を、下の View に Project=1 の検索結果を表示することにより複数の Project 内容を対比することなどが想定されている。



2 画面モード時に、片側の View のカレント行（1 行の全体がグレー背景になっている状態）が変更されると、その行の文番号と同一の文番号の行がもう一方の View にも存在すれば、その行が自動的にカレント行となります。この時、行が見えていない状態であれば見えるようにスクロールも行われます。Up, Down, PageUp, PageDown キーによりカレント行を変更した場合もこの自動同期が働きます。

一方、この同期機能はスクロールバーを操作するだけでは動作しません。これは、文の順序が上下の View で必ずしも一致しているとは限らないため（ソートを行った場合など）です。

2.2　形態素間マッピング

2.2.1　形態素間マッピングのインポート

ChaKi.NET には、Word と Word との間の対応関係を示すための特別なテーブル “word_word” が存在しており、対応する Word 間の対応関係を格納することができるようになっている。このテーブルは、異なる Project 間で Word と Word との対応関係を記述するのに使用することが想定されている。例えば、

- Project 0 に短単位での Word の並びが格納されていて、他の Project には長単位などそれとは異なる単位の Word の並びが格納されている
- Project 0 に日本語、Project 1 に英語というように対訳データを格納し、対応する Word をマークアップする

- Project 0 に通常の語順での Word の並びが格納されていて、他の Project には「読み順」などそれとは異なる語順の Word の並びが格納されている

などの使い方が考えられる。

Word 間マッピングをインポートするコマンドは、コマンドラインから “ImportWordRelation.exe” を実行する。下記に Usage を示す。

```
Usage: ImportWordRelation [Options] <InputFile> <Output>
Options (default):
  [-C] Do not pause on exit (false)
  [-b] Make relations bi-directional (false)
  [-a] Do not clear the mapping table; append mode (false)
InputFile - TSV File
Output    - .db file for SQLite / .def file for Others
```

入力は Project, Sentence, WordNo の 3 つ組を基本として、From-word, To-word を横に並べた Tab-separated 形式となる。すなわち、各行は、

From-word の Project・From-word の Sentence No・From-word の Word No・To-word の Project・To-word の Sentence No・To-word の Word No というカラムから成る。

関係は、デフォルトでは From-word から To-word の一方向だが、“-b” オプションを付けることで双方向とすることも可能である。この場合、1 つの入力それぞれについて、方向を逆にした 2 つのレコードが挿入される。

以下に日英対訳の場合の入力ファイルの例を示す。

日本語側入力ファイル (拡張 CaboCha フォーマット):

```
* 0 1D 0/0 0
ＡＬＢＵＭ　　名詞, 普通名詞, 一般,*,*,*,*,*,*, ＡＬＢＵＭ, ＡＬＢＵＭ
　　空白,*,*,*,*,*,*,*,*,*,*,
* 1 2D 0/0 0
私　　代名詞,*,*,*,*,*,*,*,*, 私, 私
の　　助詞, 格助詞,*,*,*,*,*,*,*, の, の
* 2 -1Z 0/0 0
先生　　名詞, 普通名詞, 一般,*,*,*,*,*,*, 先生, 先生
#! SEGMENT_S Apposition 0 5 ""
#! SEGMENT_S Apposition 6 10 ""
#! GROUP_S Apposition 0 1  ""
EOS
* 0 1D 0/0 0
キャスター　　名詞, 普通名詞, 一般,*,*,*,*,*,*, キャスター, キャスター
　　空白,*,*,*,*,*,*,*,*,*,*,
* 1 -1Z 0/0 0
```

```
蓮舫　　名詞,固有名詞,人名,一般,*,*,*,*,*,蓮舫,蓮舫
さん　　接尾辞,名詞的,一般,*,*,*,*,*,*,さん,さん
EOS
```

英語側入力ファイル (CoNLL-U フォーマット):

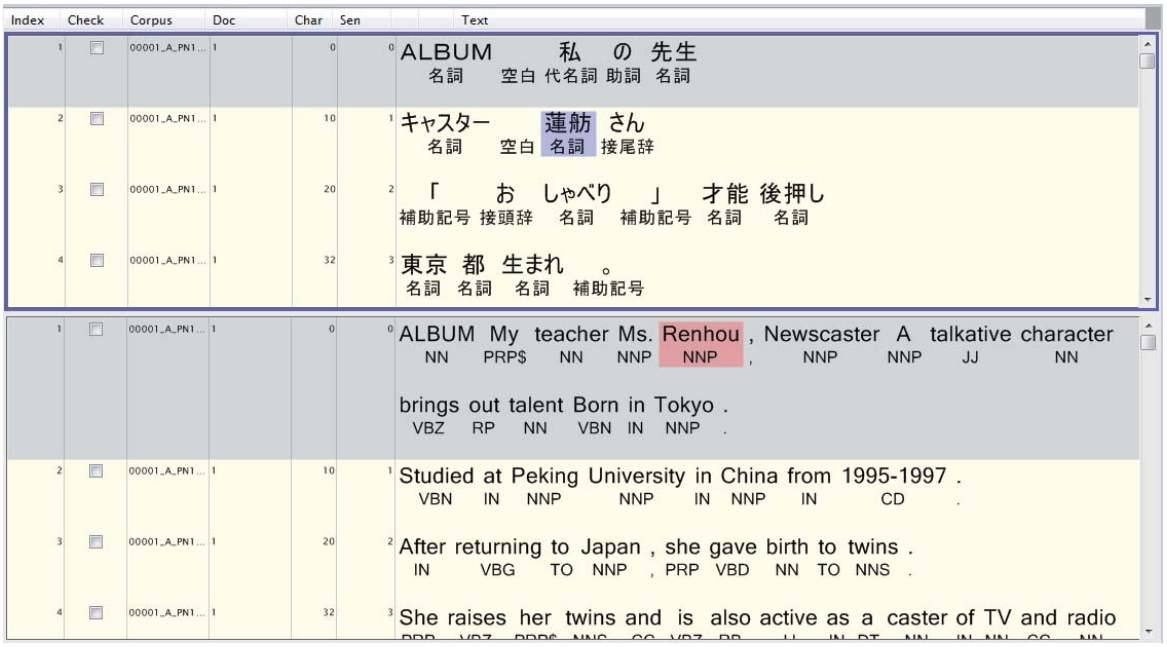
```
1   ALBUM   _   NN   NN   _   11   tmod   _   _
2   My   _   PRP$   PRP$   _   5   poss   _   _
3   teacher   _   NN   NN   _   5   nn   _   _
4   Ms.   _   NNP   NNP   _   5   nn   _   _
5   Renhou   _   NNP   NNP   _   11   nsubj   _   _
6   ,   _   ,   ,   _   5   punct   _   _
7   Newscaster   _   NNP   NNP   _   10   nn   _   _
8   A   _   NNP   NNP   _   10   nn   _   _
9   talkative   _   JJ   JJ   _   10   amod   _   _
10   character   _   NN   NN   _   5   conj   _   _
11   brings   _   VBZ   VBZ   _   0   null   _   _
12   out   _   RP   RP   _   11   prt   _   _
13   talent   _   NN   NN   _   11   dobj   _   _
14   Born   _   VBN   VBN   _   13   partmod   _   _
15   in   _   IN   IN   _   14   prep   _   _
16   Tokyo   _   NNP   NNP   _   15   pobj   _   _
17   .   _   .   .   _   11   punct   _   _
```

word_word 対応ファイル (ImportWordRelation.exe 入力ファイル):

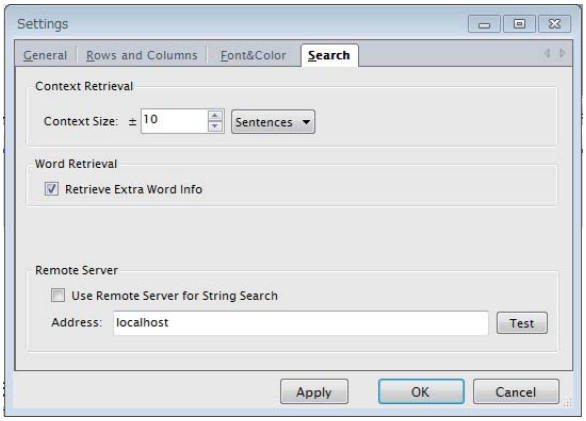
```
0       0       0       1       0       0
0       0       2       1       0       1
0       0       3       1       0       1
0       0       4       1       0       2
0       1       0       1       0       6
0       1       2       1       0       4
0       1       3       1       0       3
```

2.2.2　形態素間マッピングの可視化

形態素間マッピングの情報は KwicView の 2 画面モード上で可視化することができる。



ImportWordRelation.exe により Word-Word マッピングをコーパスにインポートしてある場合は、マッピングの From 側に一致する Word 上にマウスを置いたときに Word 背景が青色となり、同時に To 側に対応する Word の背景が自動的に赤色になる。つまり、青色背景の Word から赤色背景の Word へのマッピングが存在することを、Word 上にマウスを持っていくことにより確認することができる。 但し、word-word マッピング情報は、設定ダイアログ（メニューの [オプション]→[設定] で表示されるダイアログ）の [Search] タブにおいて、[Retrieve Extra Word Info] が ON になっていないと検索時に読み込まれないことに注意すること（デフォルトでは OFF）。



3. 活用例

3.1 BCCWJ-Trans 対訳の可視化

前節までの例では、BCCWJ に対する対訳 BCCWJ-Trans に基づいて紹介した。表 1 に BCCWJ-trans の概要について示す。今回、デモ用に形態素単位の対応を 1 サンプルにのみ付与したが、現在のところ全データに対して形態素単位の対応が付与されているわけではない。

今後、形態素単位の対応を付与していきたい。

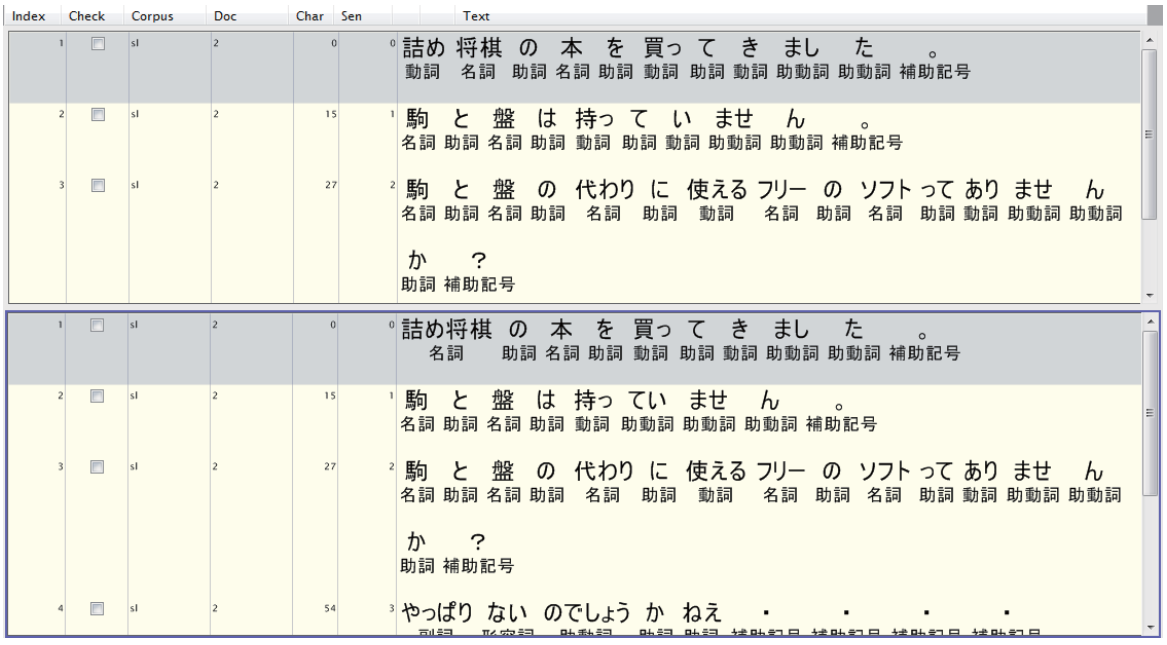
表 1　BCCWJ-Trans の概要

言語	文書数	文数	下訳	摘要
英語	6	319	有	OY 1, OC 1, PN 1, PB 1, PM 1, OW 1
中国語 (簡)	6	319	有	OY 1, OC 1, PN 1, PB 1, PM 1, OW 1
イタリア語	16	436	無	OY 6, OC 6, PN 1, PB 1, PM 1, OW 1
インドネシア語	10	337	無	OY 3, OC 3, PN 1, PB 1, PM 1, OW 1

文数は日本語側のもの。文書はアノテーションの優先順位順に選択。
OY “ブログ”, OC “知恵袋”, PN “新聞”, PB “書籍”, PM “雑誌”, OW “白書”。

3.2　BCCWJ の長単位・短単位の可視化

BCCWJ の DVD には長単位・短単位の 2 種類の形態素単位の形態論情報が付与されている。これまでの ChaKi.NET はどちらか一方の形態素単位による検索しかできなかった。プロジェクト機能を用いて、長単位・短単位を別のプロジェクトに格納することにより、それぞれの形態論情報による検索・可視化が可能になる。下図は KwicView の 2 画面モードにより 2 つの形態素単位を可視化したものである。



3.3　BCCWJ の読み時間の可視化

現在 BCCWJ に対する読み時間の付与を進めており、さまざまな可視化手法について検討している (浅原・森田 (2013), 浅原ほか (2014a))。プロジェクト機能を用いることにより、視線走査装置によって得られた読み時間の可視化をすることができる。具体的には読んでいるテキストの線形順序と視線が走査した形態素順序の 2 種類の順序を別のプロジェクトに格納する。以下では、視線走査実験結果の可視化について紹介する。

視線走査装置は、被験者がディスプレイ画面上のどの文字を注視しているのかを取得することができる機材である。この視線走査装置を用いて、視線停留箇所と停留時間を計測することにより、読文速度を取得することができる。視線走査装置として、SRResearch 社の EyeLinkCL シリーズを用いる。テキストは横書き、等幅フォントを用い、5 行単位で呈示する。自己ペース読文法と同様に、1 文書毎に内容を問う Yes/No Question に回答させる。

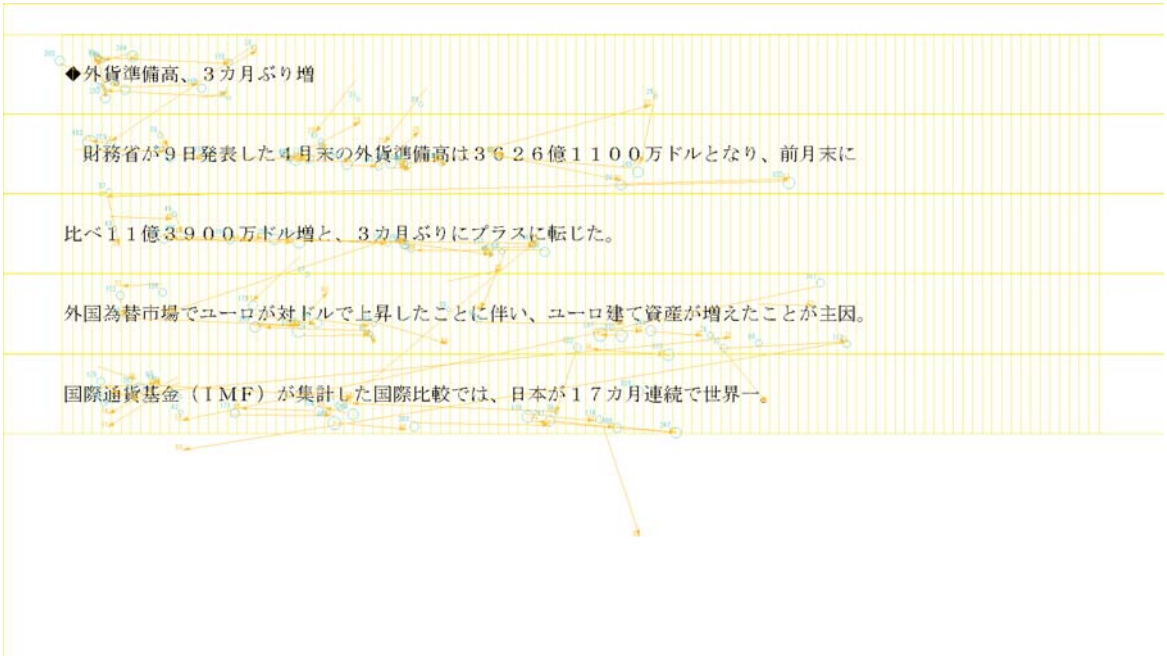


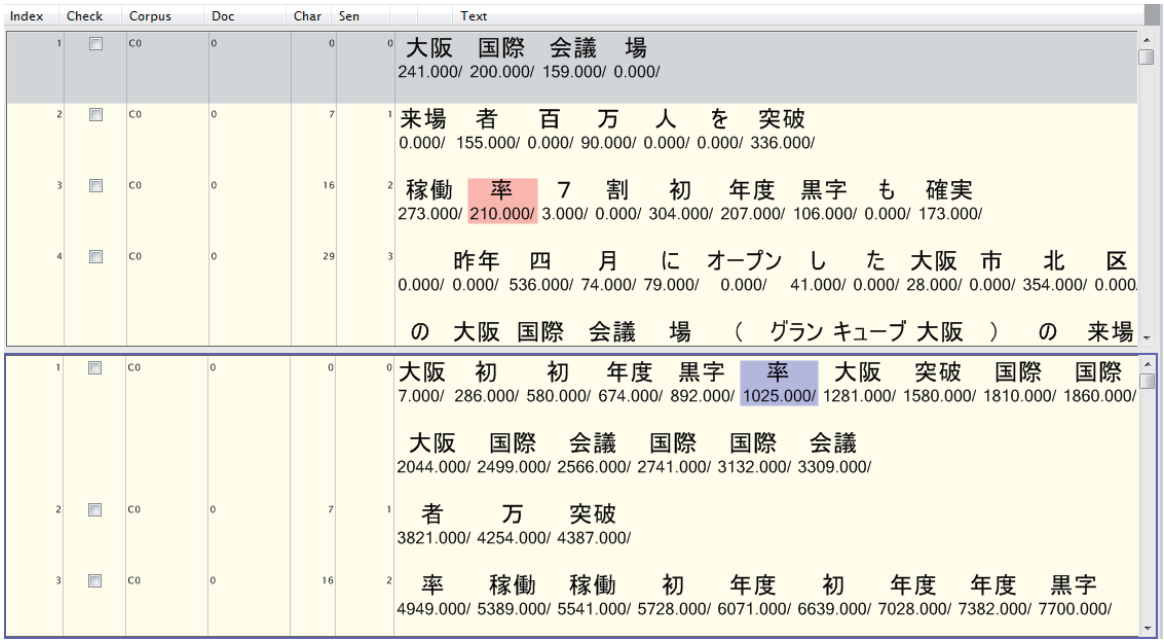
図 1　視線走査実験結果

図 1 に視線走査実験結果を示す。呈示する各文字の 1/2 幅毎に interest area (図中黄色の grid で表示) と呼ばれる領域を設定する。各 interest area 毎に視線停留箇所と停留時間、サッケード眼球運動の通過などが付与される。この interest area が設定されている半文字単位の情報に BCCWJ に付与されている短単位形態論情報、長単位形態論情報、文節境界情報を重ね合わせることにより、それぞれの単位での分析を行う。この実験法で得られるデータは読み戻しができ、かつ周辺視野により隣接する形態素・文節が読まれることもあり、全ての文節が必ず一度は読まれるわけではない。被験者は自由に読み戻し・読み飛ばしが可能である。元文書の語順に沿って分析するために次の指標が用いられる。

- First pass time
  最初に「分析単位」に視線が停留してから、他の「分析単位」に出るまでの間の視線停留時間の合計
- Total time
  「分析単位」内の視線停留時間の合計
- Regression path time
  最初に「分析単位」に視線が停留してから、より右側 (もしくは下側) の「分析単位」に

出るまでの間の視線停留時間の合計 (左側 (もしくは上側) へ停留している停留時間は累計される)

以下の図は、KwicView の上画面に読んでいるテキストの線形順序の形態素に読み時間の First pass time を付与したものを、下画面に視線走査順序の形態素に実験開始時刻を 0.000 ミリ秒とした場合の視線停留開始時刻を示したものである。下画面側の形態素（青地）にマウスカーソルを合わせることにより、当該形態素のテキスト中の位置（赤字）を示す。



4. おわりに

本稿では、コーパスコンコーダンサ ChaKi.NET のプロジェクト機能の概要と活用事例について紹介した。プロジェクト機能は ChaKi.NET Version 2.8 Revision 496 以降[(2)]で利用可能である。

謝辞

本研究の一部は科研費基盤 (B)「言語コーパスに対する読文時間付与とその利用」、国語研基幹型共同研究プロジェクト「コーパスアノテーションの基礎研究」および国語研「超大規模コーパス構築プロジェクト」によるものです。

参考文献

浅原正幸・森田敏生 (2013).「コーパスコンコーダンサ『ChaKi.NET』の連続値データ型」第 4 回コーパス日本語学ワークショップ予稿集, pp. 249–256.

浅原正幸・池本優・森田敏生 (2014a).「コーパスコンコーダンサ『ChaKi.NET』の連続値データ型 (2) —読み時間の表示—」 第 5 回コーパス日本語学ワークショップ予稿集, pp. 39–48.

(2) http://sourceforge.jp/projects/chaki/releases/

松吉俊・浅原正幸・飯田龍・森田敏生 (2014b).「拡張 CaboCha フォーマットの仕様拡張」第 5 回コーパス日本語学ワークショップ予稿集, pp. 223–232.

Maekawa, Kikuo, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den (2014). “Balanced Corpus of Contemporary Written Japanese.” *Language Resources and Evaluation*, 48, pp. 345–371.

Matsumoto, Yuji, Masayuki Asahara, Kou Kawabe, Yurika Takahashi, Yukio Tono, Akira Ohtani, and Toshio Morita (2005). “Chaki: An annotated corpora management and search system.” *Proc. of the Corpus Linguistics Conference Series (Corpus Linguistics 2005)*.