ChaOne マニュアル

山田 篤 (Studio ARC)

2007 年 7 月 27 日

1 ChaOne とは

形態素解析の結果，各語に付与された発音形に対し，語の複合による語頭・語末音の変化（連濁や促音化）を処理する必要が生じる場合がある。たとえば、unidic において，書字形「一」を持つ数詞に対する発音形は「イチ」「イッ」「ヒト」「ヒ」「ヒー」がある。これらのうちから適切なものを文脈に応じて選択することは，テキスト音声合成などへの応用では重要な課題であるが，現状の形態素解析技術で実現するのは難しい。ChaOne は，このような音変化を扱うための形態素解析後処理ツールである。現在，対象としているのは，おもに数詞と助数詞である。このような音変化は一定の語の境界をまたいでは起こらない。そのため，音変化処理の前にチャンキング処理を行ない，中単位の範囲内で音変化の処理を行なうことを想定している。

2 ChaOne の構成

ChaOne は XSLT 1.0 (http://www.w3.org/TR/xslt) 及び EXSLT-Common (http://www.exslt.org/exsl/index.html) を用いて実装されており，これらに対応した XSLT 処理系を用いて使用することができる。

ChaOne の処理は以下の 5 つのステップからなり，それぞれ単独で動作させることもできる。

1. 前処理 (prep.xsl)
2. チャンキング (chunker.xsl)
3. 音変化処理 (phonetic.xsl)
4. アクセント結合 (accent.xsl)
5. 後処理 (postp.xsl)

また，Unix 上で pipe を用いて接続する時等のために，libxslt (http://xmlsoft.org/XSLT/) を用いて，コマンドラインから ChaOne を呼び出すための C で書かれたフロントエンドプログラム (chaone.c) も用意されている。

3 使用方法

3.1 XSLT 処理系を用いる場合

chaone.xsl をパラメータなしで呼び出した場合は，前処理から音変化処理までを行う。出力の文字コードは UTF-8 である。

`standalone` パラメータに以下の値を設定することにより，動作モードを変更することができる。

表 1: standalone パラメータの値

値	処理内容
prep	前処理のみ
chunker	チャンキングのみ
phonetic	音変化処理のみ
accent	アクセント結合のみ
postp	後処理のみ
pc	前処理からチャンキングまで
pcp	前処理から音変化処理まで（省略値）
pcpa	前処理からアクセント結合まで
gtalk	前処理から後処理まで（音声合成器 gtalk 用）

以下，代表的な XSLT 処理系を用いた操作方法を示す。

3.1.1 xsltproc

xsltproc は libxslt (http://xmlsoft.org/XSLT/) のコマンドライン用のツールである。

```
xsltproc --param standalone "'xxx'" -o output.xml chaone.xsl \
    input.xml
```

で，input.xml を処理した結果が output.xml に格納される。

3.1.2 Xalan-Java

Xalan-Java (http://xml.apache.org/xalan-j/) は Apache XML Project で開発されている XSLT プロセッサである。名前が示すとおり，JRE (Java 実行環境) が必要である。

```
java org.apache.xalan.xslt.Process -PARAM standalone "xxx" -IN \
    input.xml -XSL chaone.xsl -OUT output.xml
```

で，input.xml を処理した結果が output.xml に格納される。

3.1.3 Saxon

Saxon (http://saxon.sourceforge.net/) は XSLT 2.0 の Editor である Michael Kay が開発している XSLT 処理系である。Saxon の動作にも，JRE (Java 実行環境) が必要である。

```
java -jar saxon.jar input.xml chaone.xsl standalone="xxx" > \
    output.xml
```

で，input.xml を処理した結果が output.xml に格納される。

3.1.4 msxsl

msxsl は Microsoft 社による MSXML (Microsoft XML Core Services) を用いたコマンドライン用のツールである。MSXML は exslt に対応していないが，Microsoft 社の独自の拡張を用いて ChaOne を動作させることができる。Windows 上であっても，libxslt 等 EXSLT 対応の処理系を用いる場合は，MSXML は用いない。

```
msxsl -o output.xml input.xml chaone.xsl standalone="xxx"
```

で，input.xml を処理した結果が output.xml に格納される。

3.2 C フロントエンドを用いる場合

C フロントエンドのコンパイル及び実行には libxslt (http://xmlsoft.org/XSLT/) が必要である。

3.2.1 コンパイル

1. `./configure` を実行する。オプションは以下のとおり。指定しない場合は [ ] 内のデフォルト値が使われる。

   –with-kanjicode 文字コード [EUC-JP]

   –with-chaonedir ChaOne のインストール場所 [/usr/local/chaone]

2. 続いて `make` を実行する。

3. 続いて `make install` を実行する。

3.2.2 実行

- ヘルプの表示

  `chaone {-h | --help}`

- バージョンの表示

  `chaone {-v | --version}`

- 通常の使用法

  `chaone [options] [file]`

  入力ファイルを指定しなければ，標準入力から読み込む。出力は常に標準出力が用いられる。標準入力からの読み込み時に限り，XML 宣言を出力しない。

- 設定可能なオプション

  {-e | –encoding} 入出力の文字コードを，ISO-2022-JP，EUC-JP，Shift_JIS，UTF-8 のうちから指定する。

  {-s | –mode} モードを指定しなければ，前処理から音変化処理までが行われる。

  prep: 前処理のみを行う。

  chunker: チャンキングのみを行う。

  phonetic: 音変化処理のみを行う。

  accent: アクセント結合のみを行う。

  postp: 後処理のみを行う。

  pc: 前処理からチャンキングまでを行う。

  pcp: 前処理から音変化処理までを行う。

  pcpa: 前処理からアクセント結合までを行う。

  gtalk: 前処理から後処理までを行う（gtalk 用）。

{-d | –debug} stderr に中間結果を出力する。出力文字コードは UTF-8 固定。

4 処理内容及び入出力仕様

ChaOne は，unidic を用いた解析結果の XML 文書を入力としてとる。XSLT 処理系を用いる場合は，整形式でさえあれば，入力の文字コードは任意である。C フロントエンドを用いる場合は，コンパイル時または起動時に設定した文字コードに限定され，XML 宣言は不要である。

出力の文字コードは，XSLT 処理系を用いる場合はデフォルトで UTF-8，C フロントエンドを用いる場合は，コンパイル時または起動時に設定した文字コードになる。

ChaOne には 5 つの処理ステップがあり，前のステップの出力が次のステップの入力となる。以下，各ステップ毎に述べる。

4.1 前処理

前処理では，unidic を用いた形態素解析結果に対して，予約されたタグ (S, W1) 以外を空タグに置換する。S は文，W1 は短単位を表すタグである。空タグへの置換は，開始位置に空タグを挿入し，その W1len 属性として子孫の W1 の数を持たせる。また，S, W1 に名前空間が設定されていない場合は，設定する。S, W1 等の名前空間 URI は `http://www.unidic.org/chasen/ns/structure/1.0` とする。本文書中では，名前空間接頭辞として `cha:` を用いる。

さらに gtalk モードの場合に限り，英字からなる未知語を，ea_symbol_table.xml を用いて構成文字に分解する。

たとえば，unidic を用いた解析結果を

```
<cha:S xmlns:cha="http://www.unidic.org/chasen/ns/structure/1.0"
<num:A xmlns:num="http://www.unidic.org/numtrans/ns/structure/1.0" \
    type="decimal" origText="１１">
<cha:W1 orth="十" pron="{ジュー/ジュッ/トー/ジッ/ト}" pos="名詞-数詞" \
    lForm="{ジュウ/ジュウ/トオ/ジュウ/トオ}" lemma="十" iConType="Nj" \
    fType="{十促/十促/オ長添/十促/オ長添}" fForm="{基本/促音/長音添加/促
音/基本}形" aType="1" \
    aConType="C3">十</cha:W1>
<cha:W1 orth="一" pron="{イチ/ヒト/イッ/ヒ/イツ/ヒー}" pos="名詞-数詞" \
    lForm="{イチ/ヒト/イチ/ヒト/イチ/ヒト}" lemma="一" iConType="{N1/N1/N1///}" \
    fType="{チ促//チ促/イ長添//イ長添}" fForm="{基本形//促音形/基本形//長
音添加形}" \
    aType="{2/0/2/0/1,2/0}" aConType="C3">一</cha:W1>
</num:A>
```

```
<cha:W1 orth="本" pron="{ポ/ホ/ボ}ン" pos="接尾辞-名詞的-助数詞" lForm="
ホン" lemma="本" \
    iType="ホ混合" iForm="{半濁音/基本/濁音}形" fConType="B1S6SjShS,B1S8SjShS" \
    aConType="C3">本</cha:W1>
<cha:W1 orth="。" pron="" pos="補助記号-句点" lForm="" lemma="。">。</cha:W1>
</cha:S>
```

とする。この場合の chasenrc の出力フォーマット指定は,

```
(OUTPUT_FORMAT "<cha:W1 orth=\"%m\" pron=\"%?U/%m/%a0/\" \
    pos=\"%U(%P-)\"%?T/ cType=\"%T \"//%?F/ cForm=\"%F \"//%?I/ %i0 \
    //>%m</cha:W1>\n")
```

である。あるいは,

```
(OUTPUT_FORMAT "<cha:W1 orth=\"%m\" pron=\"%?U/%m/%a0/\" \
    pos=\"%U(%P-)\"%?T/ cType=\"%T \"//%?F/ cForm=\"%F \"//%?I/ %i0 \
    ///>\n")
```

でもよい。これを前処理にかけた結果は

```
<cha:S xmlns:cha="http://www.unidic.org/chasen/ns/structure/1.0">
<num:A xmlns:num="http://www.unidic.org/numtrans/ns/structure/1.0" \
    type="decimal" origText="１１" W1len="2"/>
<cha:W1 orth="十" pron="{ジュー/ジュッ/トー/ジッ/ト}" pos="名詞-数詞" \
    lForm="{ジュウ/ジュウ/トオ/ジュウ/トオ}" lemma="十" iConType="Nj" \
    fType="{十促/十促/オ長添/十促/オ長添}" fForm="{基本/促音/長音添加/促
音/基本}形" aType="1" \
    aConType="C3">十</cha:W1>
<cha:W1 orth="一" pron="{イチ/ヒト/イッ/ヒ/イツ/ヒー}" pos="名詞-数詞" \
    lForm="{イチ/ヒト/イチ/ヒト/イチ/ヒト}" lemma="一" iConType="{N1/N1/N1///}" \
    fType="{チ促//チ促/イ長添//イ長添}" fForm="{基本形//促音形/基本形//長
音添加形}" \
    aType="{2/0/2/0/1,2/0}" aConType="C3">一</cha:W1>
<cha:W1 orth="本" pron="{ポ/ホ/ボ}ン" pos="接尾辞-名詞的-助数詞" lForm="
ホン" lemma="本" \
    iType="ホ混合" iForm="{半濁音/基本/濁音}形" fConType="B1S6SjShS,B1S8SjShS" \
    aConType="C3">本</cha:W1>
<cha:W1 orth="。" pron="" pos="補助記号-句点" lForm="" lemma="。">。</cha:W1>
</cha:S>
```

となる。A タグが空タグに置換され，W1len 属性に元の子孫の W1 の数 2 が入っている。

4.2 チャンキング

音変化処理に必要な中単位を構成する。中単位のタグは W2 とする。現状は，人手で作成した規則 (chunk_rules.xml) を用いて，W1 の品詞に基づいた結合を行っている。さらに，辞書に登録されている複合語に対しては，W1 の w2Chunk 属性の値 (B, I, E) を用いた構成を行う。

先の前処理結果にチャンキングを施すと，

```
<cha:S xmlns:cha="http://www.unidic.org/chasen/ns/structure/1.0">
<num:A xmlns:num="http://www.unidic.org/numtrans/ns/structure/1.0" \
    type="decimal" origText="１１" W1len="2"/>
<cha:W2 orth="十一本" pos="名詞-普通名詞-副詞可能">
<cha:W1 orth="十" pron="{ジュー/ジュッ/トー/ジッ/ト}" pos="名詞-数詞" \
    lForm="{ジュウ/ジュウ/トオ/ジュウ/トオ}" lemma="十" iConType="Nj" \
    fType="{十促/十促/オ長添/十促/オ長添}" fForm="{基本/促音/長音添加/促
音/基本}形" aType="1" \
    aConType="C3">十</cha:W1>
<cha:W1 orth="一" pron="{イチ/ヒト/イッ/ヒ/イツ/ヒー}" pos="名詞-数詞" \
    lForm="{イチ/ヒト/イチ/ヒト/イチ/ヒト}" lemma="一" iConType="{N1/N1/N1///}" \
    fType="{チ促//チ促/イ長添//イ長添}" fForm="{基本形//促音形/基本形//長
音添加形}" \
    aType="{2/0/2/0/1,2/0}" aConType="C3">一</cha:W1>
<cha:W1 orth="本" pron="{ポ/ホ/ボ}ン" pos="接尾辞-名詞的-助数詞" lForm="
ホン" lemma="本" \
    iType="ホ混合" iForm="{半濁音/基本/濁音}形" fConType="B1S6SjShS,B1S8SjShS" \
    aConType="C3">本</cha:W1>
</cha:W2>
<cha:W2 orth="。" pos="補助記号-句点">
<cha:W1 orth="。" pron="" pos="補助記号-句点" lForm="" lemma="。">。</cha:W1>
</cha:W2>
</cha:S>
```

となる。

4.3 音変化処理

音変化処理では，W2 内での語頭，語末の音変化に注意しながら，W1 の並記された発音形の中から適切なものを選択するとともに，W2 に発音形を付与する。

語末の音変化は，後続する語との関係で決まる。unidic では，後続語が与える当該語への影響を語末変化結合型として記述している。このため，当該語の語彙素読み (lForm)・語彙素表記 (lemma)・発音形 (pron)・語末変化型 (fType)・語末変化形 (fForm) と，後続語の語末変化結合型 (fConType) を引数とする以下のような関数を定義し，その値により最尤の発音形を選択するような枠組みが考えられる。

```
F(lForm, lemma, pron, fType, fForm, fConType)
```

語頭の音変化についても，当該語の語頭変化型・語頭変化形と前接語の語頭変化結合型を用いて同様の定式化ができる。現状は，上記の関数部分は規則ベースのテーブル (FPAfn.xml 及び IPAfn.xml) で処理している。

先のチャンキング結果に音変化処理を施すと，

```
<cha:S xmlns:cha="http://www.unidic.org/chasen/ns/structure/1.0">
<num:A xmlns:num="http://www.unidic.org/numtrans/ns/structure/1.0" \
    type="decimal" origText="１１" W1len="2"/>
<cha:W2 orth="十一本" pos="名詞-普通名詞-副詞可能" pron="ジューイッポン">
<cha:W1 orth="十" pron="ジュー" pos="名詞-数詞" lForm="ジュウ" lemma="十" iConType="Nj" \
    fType="十促" fForm="基本形" aType="1" aConType="C3">十</cha:W1>
<cha:W1 orth="一" pron="イッ" pos="名詞-数詞" lForm="イチ" lemma="一" iConType="N1" \
    fType="チ促" fForm="促音形" aType="2" aConType="C3">一</cha:W1>
<cha:W1 orth="本" pron="ポン" pos="接尾辞-名詞的-助数詞" lForm="ホン" lemma="
本" \
    iType="ホ混合" iForm="半濁音形" fConType="B1S6SjShS,B1S8SjShS" \
    aConType="C3">本</cha:W1>
</cha:W2>
<cha:W2 orth="。" pos="補助記号-句点" pron="">
<cha:W1 orth="。" pron="" pos="補助記号-句点" lForm="" lemma="。">。</cha:W1>
</cha:W2>
</cha:S>
```

となる。助数詞「本」の語末変化結合型は”B1S6SjShS”と定義されており，このうち，数詞「一」との接続に関する部分を取り出すと”1S”となる。このとき，規則テーブルから F (イチ, 一, イッ, チ促, 促音形, 1S) = 1.0 が得られ，発音形として「イッ」が選ばれている。「本」の語頭変化についても同様である。

4.4 アクセント結合

アクセント結合では，アクセント句を構成し，アクセント句内でのアクセント結合を行う。アクセント句のタグは W2 とする。現在，アクセント句は人手で作成した規則 (ap_rule.xml) を用いて構成している。また，アクセント結合も，規則ベース (accent_rule.xml) で結合している。

さらに，gtalk モードの場合に限り，単漢字辞書 (kannjiyomi.xml) を用いて，未知語への単漢字読み付与を行う。

先の音変化処理結果にアクセント結合を施すと，

```
<cha:S xmlns:cha="http://www.unidic.org/chasen/ns/structure/1.0">
<cha:AP orth="" pron="" aType="0" silence="NON">
<num:A xmlns:num="http://www.unidic.org/numtrans/ns/structure/1.0" \
```

```
    type="decimal" origText="１１" W1len="2"/>
</cha:AP>
<cha:AP orth="十一本" pron="ジューイッポン" aType="3" silence="NON">
<cha:W2 orth="十一本" pos="名詞-普通名詞-副詞可能" pron="ジューイッポン" aType="3" \
    aConType="C3">
<cha:W1 orth="十" pron="ジュー" pos="名詞-数詞" lForm="ジュウ" lemma="十" iConType="Nj" \
    fType="十促" fForm="基本形" aConType="C3" aType="1">十</cha:W1>
<cha:W1 orth="一" pron="イッ" pos="名詞-数詞" lForm="イチ" lemma="一" iConType="N1" \
    fType="チ促" fForm="促音形" aConType="C3" aType="2">一</cha:W1>
<cha:W1 orth="本" pron="ポン" pos="接尾辞-名詞的-助数詞" lForm="ホン" lemma="
本" \
    iType="ホ混合" iForm="半濁音形" fConType="B1S6SjShS,B1S8SjShS" aConType="C3" \
    aType="">本</cha:W1>
</cha:W2>
</cha:AP>
<cha:AP orth="。" pron="" aType="" silence="SILE">
<cha:W2 orth="。" pos="補助記号-句点" pron="" aType="" aConType="">
<cha:W1 orth="。" pron="" pos="補助記号-句点" lForm="" lemma="。" \
    aType="">。</cha:W1>
</cha:W2>
</cha:AP>
</cha:S>
```

となる。

4.5 後処理

後処理は音声合成器 gtalk 向けのもので，それ以外の目的では使用しない。簡易版品詞体系 (pos_sys.xml) への変換と，名前空間の取り外しを行う。

先のアクセント結合結果に後処理を施すと，

```
<S>
<AP orth="" pron="" aType="0" silence="NON">
<A type="decimal" origText="１１" W1len="2"/>
</AP>
<AP orth="十一本" pron="ジューイッポン" aType="3" silence="NON">
<W2 orth="十一本" pos="名詞-普通名詞-一般" pron="ジューイッポン" aType="3" aConType="C3">
<W1 orth="十" pron="ジュー" pos="名詞-数詞" lForm="ジュウ" lemma="十" iConType="Nj" \
    fType="十促" fForm="基本形" aConType="C3" aType="1"/>
<W1 orth="一" pron="イッ" pos="名詞-数詞" lForm="イチ" lemma="一" iConType="N1" \
    fType="チ促" fForm="促音形" aConType="C3" aType="2"/>
<W1 orth="本" pron="ポン" pos="接尾辞-名詞的-一般" lForm="ホン" lemma="
本" iType="ホ混合" \
    iForm="半濁音形" fConType="B1S6SjShS,B1S8SjShS" aConType="C3" aType=""/>
</W2>
</AP>
<AP orth="。" pron="" aType="" silence="SILE">
<W2 orth="。" pos="補助記号-一般" pron="" aType="" aConType="">
<W1 orth="。" pron="" pos="補助記号-一般" lForm="" lemma="。" aType=""/>
</W2>
```

```
</AP>
</S>
```

となる。

5 ファイル一覧

表 2: ファイル一覧

ファイル名	内容
chaone.xsl	XSLT メインファイル
chaone4gtalk_win.xsl	Windows 版 gtalk 用ローダ
prep.xsl	前処理用 XSLT ファイル
chunker.xsl	チャンキング用 XSLT ファイル
phonetic.xsl	音変化処理用 XSLT ファイル
accent.xsl	アクセント結合用 XSLT ファイル
postp.xsl	後処理用 XSLT ファイル
ea_symbol_table.xml	英字変換テーブル
chunk_rules.xml	チャンキング用規則
FPAfn.xml	語末音変化用テーブル
IPAfnxml	語頭音変化用テーブル
kannjiyomi.xml	単漢字辞書テーブル
ap_rule.xml	アクセント句構成規則
accent_rule.xml	アクセント結合規則
pos_sys.xml	gtalk 用品詞体系変換テーブル
manual.pdf	マニュアル
license.txt	ライセンス文書
以下のファイルは Windows 用パッケージには含まれない	
Makefile.in	Makefile のスケルトン
configure	Makefile 生成スクリプト
configure.in	configure のスケルトン
chaone.c	C フロントエンドプログラムソースファイル

6 著作権

ChaOne 及び本マニュアルは，山田篤 (Studio ARC) がすべての権利を保有する。