日本語形態素解析システム
JUMAN++ version 1.00

京都大学大学院情報学研究科
黒橋・河原研究室

平成 28 年 9 月

Version 1.00 September 2016

目 次

1　はじめに

日本語文の解析では，文を単語に区切りその品詞，読み，活用などを明らかにする形態素解析が最初に必要な処理となります．日本語文の形態素解析を行うシステムとして，これまでに JUMAN，Chasen，MeCab をはじめ多数のシステムが開発され，広く利用されてきました．それらのシステムはそれなりに高い解析精度を実現しているものの，機械翻訳，情報検索，対話システムなどの上位のアプリケーションの誤りを分析すると，それが形態素解析の誤りに起因することが少なくありませんでした．従来の形態素解析は，局所的な文法的制約や，数万文程度の訓練データ中の局所的な単語並びの傾向を学習したもので，広い文脈での意味的整合性は考慮しておらず，それが時に起こりうるトンチンカンな解析誤りの原因でした．

このような問題を解決するために，JUMAN++では，RNN(recurrent neural network) に基づく言語モデルを利用しています．RNN 言語モデルでは，単語の密ベクトルに基づく再帰的な文脈の意味表現を用いることにより，次に出現する意味的に妥当な単語を予測します．これと，京都大学テキストコーパス（新聞記事）および京都大学ウェブリードコーパス（ウェブテキスト）から学習した基本モデルを組み合わせることにより，文全体として意味的に妥当な単語分割を実現しました．従来システムとの精度の比較は参考文献 [6, 7] で報告しています．

JUMAN++の文法および辞書は，JUMAN の標準文法と辞書を継承しています．3 万語程度の基本辞書についてさまざまな語彙情報・意味情報を人手で正確に整備し，その範囲を超えるものについては Wikipedia やウェブコーパスからの自動語彙獲得を行っています．

さらに，JUMAN++では部分アノテーションの枠組みを取り入れています．高精度な形態素解析システムであっても，実際のテキストの解析における誤りは避けられません．解析誤りを発見した場合，正しい単語区切りの情報を与えた部分文字列を登録することにより，そのような誤りを繰り返さないようにシステムの再学習を行うことができます．今後，このような部分アノテーションデータをユーザ・コミュニティで共有することにより，JUMAN++をより高精度にしていくことを計画しています．

JUMAN++は，おもに CREST「知識に基づく構造的言語処理の確立と知識インフラの構築」(研究代表者:黒橋禎夫) において開発を行いました．CREST の関係各位に感謝致します．また，JUMAN++は JUMAN で整備された文法・辞書を引き継いでおり，これまで JUMAN の開発に協力頂いた方々の貢献に負うところが少なくありません．一人一人のお名前を挙げることはできませんが，ここで改めて感謝を申し上げます．

平成 28 年 9 月

本システムに関するお問い合わせは以下にお願いします．
京都大学大学院情報学研究科 知能情報学専攻 黒橋・河原研究室
Email: nl-resource@nlp.ist.i.kyoto-u.ac.jp

2 JUMAN++の使用方法

2.1 動作環境

JUMAN++ を使用するには次の環境が必要である．

動作環境

- OS: Linux (CentOS 6.7 で動作を確認)
- 必要メモリ: 4GB 以上
- ディスク容量: 2GB 以上

必須ツール・ライブラリ

- gcc (4.9 以降)
- Boost C++ Libraries (1.57 以降) [1]

推奨ライブラリ（導入することで，動作を高速化することができる）

- gperftool [2]
- libunwind [3](gperftool を 64bit 環境で動作させる場合に必要)

2.2 インストール

JUMAN++ を使用するには，以下の手順で配布アーカイブをダウンロード，インストールする必要がある．インストールされるものは，JUMAN++本体，JUMAN++ のシステム標準辞書，システム標準モデル（訓練済みのパラメタ），言語モデルである．

```
% wget http://lotus.kuee.kyoto-u.ac.jp/nl-resource/jumanpp/jumanpp-1.00.tar.xz
  (約 1.3GB)
% tar xJvf jumanpp-1.00.tar.xz
% cd jumanpp-1.00
% ./configure
% make
% sudo make install
```

デフォルトでは /usr/local/ にインストールされる．インストール先を指定する場合は，./configure に--prefix=/path/to/somewhere/オプションを付加する．

[1] http://www.boost.org/
[2] https://github.com/gperftools/gperftools
[3] http://www.nongnu.org/libunwind/

2.3 使用方法

JUMAN++の実行ファイルは jumanpp という名前である．標準入力から解析するテキストを読み込み，解析結果を標準出力に出力する．入力は UTF-8 でエンコードされた一行一文のテキストを仮定している [4]．文章を解析する場合には，あらかじめ文ごとに改行で区切ったテキストを入力とする．半角の# で始まる行が入力された場合には，解析は行わずコメント行として扱い，バージョン情報を付加して出力する．ただし，##JUMAN++で始まる行は 2.7 節で述べる実行時設定変更コマンドとして解釈する．

```
% cat cake.txt
# S-ID: 00000000-01
ケーキを食べる
% cat cake.txt | jumanpp
# S-ID: 00000000-01 JUMAN++:1.00
ケーキ けーき ケーキ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:ケーキ/けーき カテゴリ:人工物-食べ物 ドメイン:料理・食事"
を を を 助詞 9 格助詞 1 * 0 * 0 NIL
食べる たべる 食べる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:食べる/たべる ドメイン:料理・食事"
EOS
```

以下のオプションが用意されている．

-s, --specifics N	N-Best 解を詳細出力形式で出力（2.4 節）
-B, --beam width	解析に利用する Beam 幅（4.1 節） (default: width = 5)
--partial	単語境界の一部が明示されたテキストの解析を行う（5.3 節）
--force-single-path	同一スパンで同一スコアの形態素が複数ある場合も常に 1 つだけ表示する．（4.1 節）
-v, --version	バージョンを表示
--debug	デバッグ出力を表示する
-h, --help	ヘルプを表示

2.4 出力形式

JUMAN 形式 (default)

各行が 1 つの形態素を表す．形態素の各項目は半角スペース区切りで表される．各項目が表す内容は以下の通りである．

表層形 読み 見出し語 品詞大分類 品詞大分類 ID 品詞細分類 品詞細分類 ID 活用型 活用型 ID 活用形 活用形 ID 意味情報

[4]現在のバージョンでは，システム標準の辞書は，全角文字で登録されているため，入力を全角文字に統一して解析を行うことを推奨する．

出力例:

```
家 いえ 家 名詞 6 普通名詞 1 * 0 * 0 "代表表記:家/いえ 漢字読み:訓 カテゴリ:組織・団体; 場所-施設 ドメイン:家庭・暮らし"
に に に 助詞 9 格助詞 1 * 0 * 0 NIL
は は は 助詞 9 副助詞 2 * 0 * 0 NIL
いる いる いる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:鋳る/いる"
@ いる いる いる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:居る/いる"
@ いる いる いる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:射る/いる"
EOS
```

'EOS' は文区切りを表す．意味情報は半角スペースを区切りとしている．なお，品詞細分類や活用型などが定義されていない形態素の場合，その欄は '*' となり，対応する ID 欄は '0' となる．意味情報に表示する事項がない場合は 'NIL' となる．同じスパンにスコアが同一の形態素候補が複数ある場合は，先頭に '@' をつけて二つ目以降の候補を表示する．半角スペースに対する出力はバックスラッシュで半角スペースをエスケープし，次のように表示される．

```
\␣␣\␣␣\␣␣特殊␣1␣空白␣6␣*␣0␣*␣0␣"代表表記:␣/␣"
```

詳細出力形式 (-s)

詳細出力形式は，N-best 解の出力に対応し，各形態素が他のどの形態素と接続するかという情報が付随する出力形式である．JUMAN 形式と同様に各行が 1 つの形態素を表し，形態素の各項目はタブで区切られている．この形式では入力にタブが存在した場合，"\t" に置き換えて表示する．1 行目にはコメント行として N-best 解の各スコアを表示する．各項目が表す内容は以下の通りである．

行頭記号 形態素 ID 前方向に接続する形態素 ID の列 開始文字位置 終了文字位置 表層形 代表表記 読み 見出し語 品詞大分類 品詞大分類 ID 品詞細分類 品詞細分類 ID 活用型 活用型 ID 活用形 活用形 ID 意味情報

行頭記号には以下の 3 種類がある．

- # コメント行
- - 形態素行
- EOS 文区切り

出力例:

```
# MA-SCORE  rank1:-9.52349 rank2:-9.59653 rank3:-9.72774 rank4:-9.75929 rank5:-10.6167
-  21  0  0  1  ここ  ここ/ここ  ここ  ここ  指示詞  7  名詞形態指示詞  1  *  0  *  0  特徴量スコア:-1.92711|言語モデルスコア:-0.74683|形態素解析スコア:-2.67394|ランク:1;2;3;4;5
-  44  21  2  2  に  に/に  に  に  助詞  9  接続助詞  3  *  0  *  0  特徴量スコア:-0.294647|言語モデルスコア:-0.286072|形態素解析スコア:-0.580719|ランク:5
-  43  21  2  2  に  に/に  に  に  助詞  9  格助詞  1  *  0  *  0  特徴量スコア:0.755723|言語モデルスコア:-0.286072|形態素解析スコア:0.469651|ランク:1;2;3;4
-  93  44;43  3  5  はいる  入る/はいる はいる  はいる  動詞  2  *  0  子音動詞ラ行  10  基本形  2  自他動詞:他:入れる/いれる|反義:動詞:出る/でる|特徴量スコア:-0.741122|言語モデルスコア:-1.41253|形態素解析スコア:-2.15365|ランク:3
-  70  44;43  3  3  は  は/は  は  は  助詞  9  副助詞  2  *  0  *  0  特徴量スコア:0.368752|言語モデルスコア:-0.264521|形態素解析スコア:0.104231|ランク:1;2;4;5
```

```
- 137 70 4 5 いる いる/いる いる いる 接尾辞 14 動詞性接尾辞 7 母音動詞 1 基本形 2 特徴量スコア:-1.35178|言語モデルスコア:-1.27911|形態素解析スコア:-2.63089|ランク:4
- 136 70 4 5 いる 鋳る/いる いる いる 動詞 2 * 0 母音動詞 1 基本形 2 特徴量スコア:-1.01211|言語モデルスコア:-0.991401|形態素解析スコア:-2.00351|ランク:1;5
- 135 70 4 5 いる 居る/いる いる いる 動詞 2 * 0 母音動詞 1 基本形 2 特徴量スコア:-1.01211|言語モデルスコア:-0.991401|形態素解析スコア:-2.00351|ランク:1;5
- 134 70 4 5 いる 射る/いる いる いる 動詞 2 * 0 母音動詞 1 基本形 2 特徴量スコア:-1.01211|言語モデルスコア:-0.991401|形態素解析スコア:-2.00351|ランク:1;5
- 133 70 4 5 いる 要る/いる いる いる 動詞 2 * 0 子音動詞ラ行 10 基本形 2 特徴量スコア:-1.11446|言語モデルスコア:-0.991401|形態素解析スコア:-2.10586|ランク:2
- 132 70 4 5 いる 煎る/いる いる いる 動詞 2 * 0 子音動詞ラ行 10 基本形 2 ドメイン:料理・食事|特徴量スコア:-1.11446|言語モデルスコア:-0.991401|形態素解析スコア:-2.10586|ランク:2
EOS
```

意味情報の区切りは JUMAN 型式と異なり，| を区切りとしている．代表表記は JUMAN 形式では意味情報の 1 つとして表示していたが，詳細出力形式では意味情報とは独立に表示する．また，詳細出力形式の場合のみ意味情報に**ランク**が出力されており，その形態素が N-best 解のうち何番目の解に出現したかを表す．同じスパンに同一スコアの形態素候補が複数ある場合は，同ランクの形態素を複数個表示する．意味情報中の**特徴量スコア**，**言語モデルスコア**，**形態素解析スコア**は，解析時にその形態素に付与されたスコアを表す．特徴量スコアは基本モデルの出力するスコア，言語モデルスコアは RNN 言語モデルの出力するスコア，形態素解析スコアはその合計値を表す．JUMAN++の解析アルゴリズムおよびスコアの詳細については 4 節にて述べる．

2.5 サーバ・クライアントモードでの使用

JUMAN++ をサーバモードであらかじめ起動しておくことで，起動時のオーバーヘッドを回避することができる．JUMAN++ をサーバ･クライアントモードで実行するスクリプトは，配布アーカイブの展開先にある `script` ディレクトリ内に含まれている `script/server.rb`，`script/client.rb` である．

JUMAN++ をサーバモードで起動する方法

server.rb スクリプトの `--cmd` オプションに，JUMAN++ を起動するコマンドを渡して実行する．デフォルトでは TCP ポート 12000 を利用する．ポートを変更する場合は`--port 1234` のように指定する．

```
$ ruby script/server.rb --cmd "jumanpp -B 5" --host host.name --port 1234
```

JUMAN++ をクライアントモードで実行する方法

client.rb スクリプトを使う．サーバを他のホストで起動している場合にはホスト名を`--host <hostname>` で指定する．デフォルトでは ポート 12000 を利用するが，変更する場合には `--port 1234` のように指定する．

```
$ echo "ケーキを食べる" | ruby script/client.rb --host host.name --port 1234
```

```
ケーキ けーき ケーキ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:ケーキ/けーき カテゴリ:人工物-食
べ物 ドメイン:料理・食事"
を を を 助詞 9 格助詞 1 * 0 * 0 NIL
食べる たべる 食べる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:食べる/たべる ドメイン:料理・
食事"
EOS
```

2.6 Python ラッパー

python モジュール “pyknp” を用いることにより，python から JUMAN++を利用することができる．

- インストール方法

```
% wget http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/knp/pyknp-0.3.zip
% unzip pyknp-0.3.zip
% cd pyknp-0.3

% sudo python setup.py install [--prefix=path]
```

- サンプルプログラム
  pyknp のサンプルプログラムを以下に示す．このプログラムを実行すると,「ケーキを食べる」という文の解析結果が表示される．以下のプログラムは python 2.7 用であるが，pyknp は python 2, python 3 の両系統に対応している．以下のサンプルプログラムは，配布アーカイブの展開先にある sample/python_juman.py ディレクトリ内にある．

```
#-*- encoding: utf-8 -*-
from pyknp import Jumanpp
import sys
import codecs
sys.stdin = codecs.getreader('utf_8')(sys.stdin)
sys.stdout = codecs.getwriter('utf_8')(sys.stdout)

# Use Juman++ in subprocess mode
jumanpp = Jumanpp()
result = jumanpp.analysis(u"ケーキを食べる")

for mrph in result.mrph_list():
    print u"見出し:%s" % (mrph.midasi)
```

このモジュールに関するより詳しい情報は “pyknp” の Readme を参照してほしい．また，構文・格解析システム KNP と組合わせて使用することもできる．配布アーカイブの展開先にある sample/python_knp.py を参照のこと．

2.7 実行時設定変更コマンド

サーバモードでの実行時や，解析する文ごとに設定を変更したい場合に，JUMAN++を再実行することなしに，設定を変更するためのコマンドが用意されている．“##JUMAN++” で始まる行を入力された場合には，解析や表示に関する設定を変更するコマンドとして解釈し，実行する．コマンドとしては以下の４つを解釈する．

- `##JUMAN++ set-lattice N`
  詳細出力形式で表示する N-best 解の個数を `N` に変更する．
- `##JUMAN++ set-beam width`
  解析に用いるビーム幅を `width` に変更する．
- `##JUMAN++ set-force-single-path`
  同一スパン同一スコアの形態素を表示しない．
- `##JUMAN++ unset-force-single-path`
  同一スパン同一スコアの形態素を表示する．

3　日本語形態素解析に用いる言語資源

本システムで用いる日本語の形態素文法および辞書，解析器の訓練に用いたコーパスについて説明する．

3.1　システム標準文法

JUMAN++では JUMAN で用いられている文法を継承し，利用している．具体的には，品詞分類定義ファイル，活用関係定義ファイル，活用定義ファイルを利用しており，ここではこれをシステム標準文法と呼ぶ．

システム標準文法は，益岡・田窪文法 [1] を参照し，それを拡張して作成された．

1. **品詞分類定義ファイル：**(cf. `JUMAN.grammar`)

   品詞分類定義ファイルでは，本システムで用いる品詞およびその品詞細分類の名称を定義する．

   益岡・田窪文法に「特殊」(句読点・記号・括弧など) を加え，接辞を「接頭辞」「接尾辞」に分けて，14 種類の品詞を定義した．

2. **活用関係定義ファイル：** (cf. `JUMAN.kankei`)

   活用関係定義ファイルは，品詞分類定義ファイルによって活用すると定義された品詞または品詞細分類それぞれに対してそれが取り得る活用型の一覧を定義する．

3. **活用定義ファイル：** (cf. `JUMAN.katuyou`)

   活用定義ファイルは，活用関係定義ファイルによって定義された個々の活用型に対して，それが取り得る活用形の名前およびその活用語尾を定義する．

   益岡・田窪文法に対して，文語的表現・口語的表現・敬語表現に対応できるように拡張を行い，21 の一般的な活用型と 7 の特殊な活用型を定義した．

システム標準文法に対して変更を行った際には，変更後の文法に対応した辞書，コーパスを用意し，モデルの訓練を行う必要がある．辞書の更新方法は 3.2 節で，コーパスについては 3.3 節で，モデルの訓練方法は 5 節で述べる．

3.2　辞書

辞書は，JUMAN++が解析に使用する形態素の定義を記述したものである．本システムでは辞書をあらかじめコンパイルし，解析で利用する型式（dic.bin, dic.da 等）に変換してから用いる [5]．ただし，本システムではコンパイル済みの辞書をリソースファイルに含めて配布しているため，通常はユーザがコンパイルを行う必要はない．

本システムでは付属するシステム標準辞書の他，ユーザが作成した辞書を利用することができる．システム標準辞書については 3.2.2 節で，ユーザ辞書の追加については 3.2.4 節で説明する．

[5] JUMAN では複数のコンパイル済みの辞書を扱うことができたが，本システムでは辞書全体を 1 つのコンパイル済み辞書に変換する必要がある．

3.2.1 辞書の記述法

辞書はリスト構造を用いて記述する．辞書は，‘dic’ という拡張子をもつファイルに格納する．辞書は複数のファイルに分割されていてもよい．辞書ファイルでは，各行中のセミコロン ‘;’ 以降の文字列はその行末までがコメントとみなされる．

各形態素は，複数の見出し語を持ってもよい．複数の見出し語が記述されている場合，見出し語以外の形態素情報は共有される．次に辞書の形態素定義の記述方法を BNF で示す．

〈形態素定義〉 ::= (〈#品詞名〉〈形態素情報の並び〉) |
(〈#品詞名〉(〈#品詞細分類名〉〈形態素情報〉))
〈形態素情報〉 ::= (〈見出し語情報〉〈読み情報〉〈活用型情報〉〈意味情報〉)
〈見出し語情報〉 ::= (見出し語 〈見出し語内容の並び〉)
〈見出し語内容の並び〉 ::= 〈#見出し語表記〉 | 〈#見出し語表記〉〈見出し語内容の並び〉
〈読み情報〉 ::= (読み 〈#読み表記〉)
〈活用型情報〉 ::= (活用型 〈#活用型名〉) | NIL
〈意味情報〉 ::= (意味情報 〈#意味記述〉) | NIL

- 〈#品詞名〉,〈#品詞細分類名〉は,「品詞分類定義ファイル」で定義されていなければならない．また，品詞細分類が定義されている品詞に対しては品詞細分類名を指定しなければならない．
- 〈活用型情報〉は,〈#品詞名〉または〈#品詞細分類名〉が活用すると定義されている時は省略できない．
- 〈#見出し語表記〉は形態素の表層の形として表現する．活用する形態素の表層の形は，その基本形を書かねばならない．
- 〈#読み表記〉には形態素の読みを記述する．活用する形態素については，その基本形の読みを書かねばならない．
- 〈#意味記述〉には意味情報を記述する．意味情報には二重引用符 (") で囲まれたテキストを自由に用いることができる．二重引用符に囲まれた範囲では任意の文字が使用可能である．二重引用符で囲まれたアトムの中の二重引用符は ‘\"’ によって記述可能である．記述されたデータは二重引用符を含んでそのままテキストデータとして扱われる．長さについての制限は設けていない．

  意味情報の表記法は次のとおり.「:」は意味情報の階層を示すものとし，最初の階層が同じものは一つの語に対して一つしか与えず，最初の階層が同じものが複数ある場合は 2 階層目以降を「;」で並列に並べることとする．以下に例を示す.

  加える → 反義:動詞:引く; 動詞:減らす
  貯金 → カテゴリ:人工物-金銭; 抽象物

3.2.2 システム標準辞書

システム標準辞書には基本語彙辞書，オノマトペ辞書，Wikipedia 辞書，Wiktionary 辞書，Web コーパス辞書の 5 種類の辞書が含まれている．

基本語彙辞書

基本語彙辞書は基本的に JUMAN の辞書を引き継いで用いている．基本語彙辞書は，主なものとして，内容語辞書（ContentW.dic）約 3 万語，固有名詞辞書（Noun.koyuu.dic）約 8 千語，機能語辞書（Postp.dic，Suffix.dic など），連濁辞書（Rendaku.dic）などからなる．内容語辞書を 3 万語規模としているのは，新語・専門用語等への対応は人手で行うのではなく自動獲得によって行うべきであるとの考えに基づいている．

なお連濁辞書は，連濁により濁音化した形態素の辞書であり，内容語辞書から自動生成したものである．内容語辞書に含まれるカタカナ語を除く形態素のうち，読みに濁音が含まれない名詞および活用語について一文字目の清音を濁音化した語を生成し，辞書として構築する．ただし，濁音が含まれる形態素でも意味情報に “濁音可” とついている語 (例: はしご) については濁音化を行う．
連濁辞書を用いた解析例

```
% echo "上海ガニ" | jumanpp
上海 しゃんはい 上海 名詞 6 地名 4 * 0 * 0 "代表表記:上海/しゃんはい地名:国:中国:市"
ガニ がに ガニ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:蟹/かに カテゴリ:動物; 人工物-食べ物 ドメイン:料理・食事 濁音化"
EOS
```

オノマトペ辞書（Onomatopeia.dic）

非反復形のオノマトペを自動生成し登録した辞書．品詞は副詞としている．登録する非反復形のオノマトペは以下のパターンに適合した文字列である．パターン中のＨ，Ｋ，Ｙはそれぞれ平仮名，片仮名，ヤ行拗音字 (平仮名，片仮名を含む) を表す．

- ＨっＨり　　例） もっさり，ざっくり
- ＨっＨＹり　例） ぐっちょり，ぺっちゃり
- ＫッＫリ　　例） モッサリ，ドッサリ
- ＫッＫＹリ　例） ズッチョリ，ポッチャリ
- ＫＫっと　　例） ピタっと，キュっと
- ＫＫッと　　例） ピタッと，ホロッと

Wikipedia 辞書（Wikipedia.dic）

日本語 Wikipedia（2016/06/01 版）のエントリのうち，1 形態素である可能性が高いもので，かつ基本語彙辞書に含まれない語を自動的に選択し，構築した辞書（約 83 万語）となっている．品詞細分類および意味情報は次のように決定，付与した．

- **品詞細分類**：品詞細分類としては普通名詞，人名，地名，組織名の可能性を考えている．上位語の主辞の JUMAN カテゴリ，Wikipedia の記事カテゴリを用いて，品詞細分類を決定している．
- **意味情報**：Wikipedia 辞書内の全ての形態素に「自動獲得:Wikipedia」を付与する．定義文から獲得された上位語があれば,「Wikipedia 上位語」を付与し，また，リダイレクトがあれば「Wikipedia リダイレクト」を付与している．

Wiktionary 辞書（Wiktionary.dic）

日本語 Wiktionary（2016/06/01 版）のエントリにおいて，活用が明示されている動詞･形容詞で，1 形態素である可能性が高いもののうち，基本語彙および Wikipedia 辞書に含まれない語，約 2,000 語を自動的に選択し，Wiktionary 辞書として構築した．Wiktionary に記載されている品詞や活用型はシステム標準の体系と異なるため，変換し用いている．意味情報には「自動獲得:Wiktionary」を付与する．

Web コーパス辞書（Web.dic）

Web コーパス辞書は，未知語の自動獲得（品詞，活用形の推定を含む）を行い，その結果から自動的に構築した辞書（約 1 万 1 千語）となっている．この中には「ググる」「ようつべ」「ドラえもん」などの語がある． これらの形態素の意味情報には「自動獲得:テキスト」を付与する．

与えられたコーパスから未知語の自動獲得を行うプログラムは以下で公開している．

https://github.com/murawaki/lebyr

3.2.3 システム標準辞書で用いられている主な意味情報

意味情報には主に以下の情報が含まれる．

代表表記

同じ語の表記バリエーションであることを扱うために，代表表記を設定し，これを意味情報に含めた．これによって，日本語処理の表記揺れの問題を，形態素解析を行うだけである程度取り除くことが可能となる．

また，平仮名書き等による曖昧性も，適切な範囲で複数の可能性（代表表記）を挙げるようにしており，以後の構文解析・意味解析などへの適切な入力を提供することができる．詳細は付録 B.1 を参照．

意味カテゴリ

「人」「動物」「植物」「人工物」「抽象物」などの意味カテゴリ 22 種を名詞の意味情報として付与した．詳細は付録 B.2 を参照．

ドメイン

意味カテゴリは語の上位下位関係に基づくものであり，これを意味の縦糸と考えるとすれば，意味の横糸として,「文化・芸術」「スポーツ」「健康・医学」「科学・技術」などのドメイン 12 種を設定し，これを語の意味情報として付与した (主に名詞，一部 動詞，形容詞)．

これにより，たとえば「土俵」という語には「カテゴリ:場所，ドメイン:スポーツ」,「医者」には「カテゴリ:人，ドメイン:健康・医学」という意味情報が付与されている．詳細は付録 B.3 を参照．

固有名詞

基本語彙辞書中の人名，地名については種々の意味情報を整理・付与した．詳細は付録 B.4 を参照．

見出し語間の意味関係

見出し語間の意味関係を整理し，内容語辞書約 3 万語の見出し語について網羅的に情報を付与した．付与した意味関係は，可能動詞，尊敬動詞・謙譲動詞，自動詞・他動詞，授受動詞，反義，名詞派生などの種々の派生，等の関係である．例えば可能動詞の対応は次のように記述されている．

> **可能動詞**
> 走れる → 可能動詞:走る/はしる
> 増やせる → 可能動詞:増やす/ふやす

その他の具体例を付録 B.5 に示す

読みの音訓情報　常用漢字 1 文字の見出し語に対して読みの音，訓の情報を与えた．

> **例)**
> 場（じょう） → 漢字読み:音
> 場（ば） → 漢字読み:訓
> 米（まい） → 漢字読み:音
> 米（こめ） → 漢字読み:訓

3.2.4　ユーザ辞書の追加

ユーザが辞書を追加し，解析を行うには辞書のコンパイルを行う必要がある．辞書のコンパイルに必要なスクリプトは，配布アーカイブの展開先にある `dict-build` ディレクトリ内にある．

ユーザ辞書をシステムに追加するには，ユーザ辞書を 3.2.1 節に従って作成し，そのファイルを `dict-build/userdic/` 以下に追加する．同ディレクトリ内に複数のユーザ辞書があってもよい．その後 `dict-build` ディレクトリ内で以下の手順を実行し，辞書のコンパイルとインストールを行う．インストール先を指定する場合は，`install.sh` に`--prefix /path/to/somewhere/`オプションを付加する．

```
% make
% sudo ./install.sh
```

3.3　コーパス

システム標準モデル（2.2 節）は以下の 3 種類のコーパスを用いて訓練を行った．辞書に大きな変更を加える場合には，同様のコーパスを用意して再度訓練を行う必要がある．訓練の方法については 5 節で述べる．

1. 京都大学テキストコーパス
   毎日新聞の記事に人手で形態素・構文情報を付与したコーパス．配布サイト [6] からダウンロードし，コーパス付属のスクリプトにより変換を行う．利用には別途毎日新聞 1995 年版 CD-ROM が必要となる．詳細は付属の Readme を参照のこと．標準では EUC-jp でエンコードされているため，システム標準モデルの訓練では事前に UTF-8 へ変換して用いた．

[6] `http://nlp.ist.i.kyoto-u.ac.jp/index.php?`京都大学テキストコーパス

2. 京都大学ウェブ文書リードコーパス
   さまざまなウェブ文書のリード (冒頭)3 文に人手でアノテーションを行ったコーパス．配布サイト [7] からダウンロードが可能.

3. Web テキストコーパス
   Web 上から収集したテキスト 1,000 万文を JUMAN++により自動的に解析したテキストコーパス．標準モデルに含まれる言語モデルは，このコーパスで学習を行い，上記 2 つのタグ付きコーパスで再度訓練したモデルである．

[7] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KWDLC

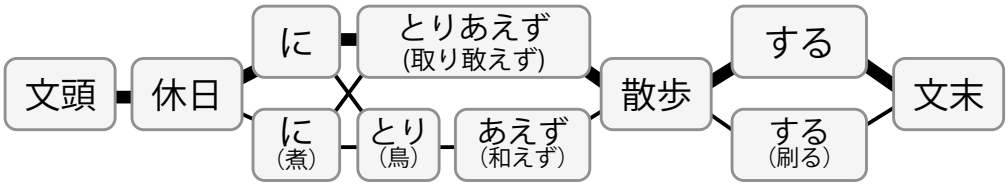

図 1: 入力文:“休日にとりあえず散歩する” に対する形態素ノードおよびパスの例．縦に複数存在する形態素ノードは同一スパンに複数の形態素ノードが存在することを示し，太いパスが選択されたパスを表す．

4 日本語形態素解析処理

JUMAN++ は言語モデルを利用した形態素解析システムである．言語モデルとして **Recurrent Neural Network (RNN)** 言語モデルを用いることにより，単語の並びの意味的な自然さを考慮した解析を行い，JUMAN や MeCab に比べ大きく性能が向上している [6]．

4.1 形態素解析アルゴリズム概要

本システムは，テキストを標準入力から一行ごとに読み込んで入力とし，形態素解析アルゴリズムにより選択された形態素からなるラティス状の構造を出力する．形態素解析アルゴリズムは，与えられた入力文字列に対する形態素ノードの生成と，文頭から文末までのスコアを最大化する形態素パスの探索を行う（図 1）．スコアとして，各形態素に与えられる次の二種類のスコアを足しあわせて用いる．

特徴量スコア: 形態素の並び（1-3 gram）に対する品詞や活用，形態素の文字列長などを表す特徴量に対し，基本モデルが与えるスコア．

言語モデルスコア: 形態素の並びの自然さに対して RNN 言語モデル が与えるスコア．

本システムでは言語モデルスコアの計算に RNN 言語モデルを用いる．RNN 言語モデルではある形態素以前に出現した形態素についての情報をベクトルとして保持する．本システムでは，このベクトルと形態素ノードにたどり着くまでに通ったパスの情報をまとめてコンテクストと呼ぶ．

解析アルゴリズムの概要は次のとおりである．

- 入力文字列中の各位置で始まるすべての形態素を辞書引きし，それぞれについて形態素ノードを生成する．これに加えて，数詞や未定義語などの形態素ノードを動的な形態素ノード生成処理により生成する．また，文の先頭と末尾には，それぞれ,「文頭」および「文末」と呼ばれる仮想的な形態素ノードを生成する．
- 得られた個々の形態素ノードに対して，スコアが上位のコンテクストをビーム幅 B 個分保持し，文頭から文末までビームサーチを行う．

形態素解析の一つの解析結果は文頭と文末を結ぶ形態素パスであり，その総スコアは，それに含まれる各形態素の特徴量スコアと言語モデルスコアの合計である．N-best 解を出力する際には，形態素パスの内，上位 N 位までのスコアを持つパスを N-best 解として出力する．`--force-single-path` オプションが指定されている場合には，スコアが同一のパスが複数ある場合にも，1 つのパスのみを選択し出力する．

4.2 動的な形態素ノード生成処理

本システムでは実テキストに現れるくだけた表現等に対応するために，自動的な語彙獲得に加えて辞書引き時および辞書引き後に動的に形態素を認識し，形態素ノードを生成する．辞書引き時には，長音記号・小書き文字による異表記の認識を行い，辞書中と異なる表記でも形態素ノードを生成する．辞書引き後には，数詞の連結，反復形オノマトペの認識，未定義語の認識を行い形態素ノードを生成する．

4.2.1 長音記号・小書き文字による異表記の認識

長音記号，小書き文字の挿入による異表記

長音記号「ー」,「〜」や，小書き文字「ぁ」,「ぃ」,「ぅ」,「ぇ」,「ぉ」が挿入された語はこれらを除いた表記で辞書の検索を行うことにより認識する．認識した形態素には意味情報 “非標準表記” を付加する．

長音記号の挿入:「報告しま〜〜す」の解析例

報告 ほうこく 報告 名詞 6 サ変名詞 2 * 0 * 0 "代表表記:報告/ほうこく 補文ト カテゴリ:抽象物"
し し する 動詞 2 * 0 サ変動詞 16 基本連用形 8 "代表表記:する/する 付属動詞候補（基本） 自他動詞:自:成る/なる"
ま〜〜す ます ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基本形 2 "代表表記:ます/ます 非標準表記"
EOS

小書き文字の挿入:「行きたぁあぁい」の解析例

行き いき 行く 動詞 2 * 0 子音動詞カ行促音便形 3 基本連用形 8 "代表表記:行く/いく 付属動詞候補（タ系） ドメイン:交通 反義:動詞:帰る/かえる"
たぁあぁい たい たい 接尾辞 14 形容詞性述語接尾辞 5 イ形容詞アウオ段 18 基本形 2 "代表表記:たい/たい 非標準表記"
EOS

長音記号，小書き文字への置換による異表記

長音記号「ー」,「〜」により本来の仮名が置換された語は元の表記で辞書引きを行うことにより認識する．ここで認識した形態素にも同様に意味情報 “非標準表記” を付加する．

長音記号による置換:「おはよーございます」の解析例

おはよー おはよう おはよう 感動詞 12 * 0 * 0 * 0 "代表表記:おはよう/おはよう 非標準表記"
ございます ございます ございます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基本形 2 "代表表記:御座います/ございます"
EOS

小書き文字による置換:「ぉはよぅございます」の解析例

ぉはよぅ おはよう おはよう 感動詞 12 * 0 * 0 * 0 "代表表記:おはよう/おはよう 非標準表記"
ございます ございます ございます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基本形 2 "代表表記:御座います/ございます"
EOS

4.2.2 数詞の連結処理

入力文中に数詞 (辞書に数詞として登録されている語) が連続して現れる場合，それらを連結して一語の数詞ノードを生成する．なお，数詞と数詞の間に出現する中黒（・），ピリオド（.）およびカンマ（,）は数詞の一部として扱う．ただし，カンマはアラビア数字を３桁ごとに区切っている場合のみ数詞の一部として扱う．以下に例を示す．

「１０，０００」の解析例

１０，０００ １０，０００ １０，０００ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
EOS

「１０．０」の解析例

１０．０ １０．０ １０．０ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
EOS

「１０，２０，３０」の解析例

１０ １０ １０ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
， ， ， 特殊 1 読点 2 * 0 * 0 NIL
２０ ２０ ２０ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
， ， ， 特殊 1 読点 2 * 0 * 0 NIL
３０ ３０ ３０ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
EOS

4.2.3 オノマトペ

非反復形のオノマトペはオノマトペ辞書に登録しているが，反復形オノマトペについては辞書にその表記を登録するのではなく，解析時に自動的に認識し，形態素のノードを生成する．自動的に認識するオノマトペは，２～４文字のひらがな・カタカナを繰り返しているものである（例： ぱくぱく，ビュンビュン，チャリンチャリン）．

反復形オノマトペ:「ばくばく食べる」の解析例

ばくばく ばくばく ばくばく 副詞 8 * 0 * 0 * 0 "自動認識"
食べる たべる 食べる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:食べる/たべる ドメイン:料理・食事"
EOS

4.2.4 未定義語

本システムでは，入力文字列中のあらゆる位置で未定義語（辞書に存在しない語）が存在する可能性を考慮している． 平仮名および記号・漢字については連続する最長２文字を一語の未定義語のノードとして生成する．それ以外の文字については，同種の文字 (カタカナ，アルファベット等) の終わりまでをまとまった一語の未定義語のノードとする．

解析結果に未定義語が含まれる場合には，品詞が “未定義語” で，細分類が “ その他” の形態素として出力される．なお，意味情報には “品詞推定:名詞” が付加される [8]．

未定義語：「キョウダイに行く」の解析例

```
キョウダイ キョウダイ キョウダイ 未定義語 15 その他 1 * 0 * 0 "品詞推定:名詞"
に に に 助詞 9 格助詞 1 * 0 * 0 NIL
行く いく 行く 動詞 2 * 0 子音動詞カ行促音便形 3 基本形 2 "代表表記:行く/いく 付
属動詞候補（タ系） ドメイン:交通 反義:動詞:帰る/かえる"
EOS
```

4.3 JUMAN との違い

JUMAN と JUMAN++ では，それぞれが同じ文を正しく解析した場合にも，解析結果が異なる場合がある．この節では，JUMAN を JUMAN++で置き換えて使用する場合に，注意が必要な事項について説明する．

動詞連用形の名詞化

動詞の連用形は，名詞として用いられることがある．例えば，“音の**響き**を大切にする” という文の “響き” という形態素は，動詞の “ 響く” の連用形が名詞化して用いられている．

JUMAN++ では**動詞の基本連用形**（“**響き**”，など）や，**動詞性接尾辞の基本連用形**（走り [9]“**過ぎ**”，等）が名詞的に用いられている場合に，それぞれを名詞や名詞性名詞接尾辞として出力する．動詞を名詞化する際は，代表表記を動詞の基本形から基本連用形に置き換え，末尾に動詞から品詞を変更したことを示す v を付加する (例:“響き/ひびき v”)．また，意味情報に “連用形名詞化:形態素解析” を追加する．

一方，JUMAN では名詞化に関する処理を扱わず，後段の構文・格解析システム KNP が関連する処理を行っていた．

出力例: JUMAN

```
% echo "音の響きを大切にする" | juman
音 おと 音 名詞 6 普通名詞 1 * 0 * 0 "代表表記:音/おと 漢字読み:訓 カテゴリ:抽象物"
@ 音 おん 音 名詞 6 普通名詞 1 * 0 * 0 "代表表記:音/おん 漢字読み:音 カテゴリ:抽象物"
の の の 助詞 9 格助詞 1 * 0 * 0 NIL
響き ひびき 響く 動詞 2 * 0 子音動詞カ行 2 基本連用形 8 "代表表記:響く/ひびく"
を を を 助詞 9 格助詞 1 * 0 * 0 NIL
```

[8] 言語モデルでスコアを計算する際には，未定義語ノードは名詞として扱われる．言語モデルの学習についての詳細は 5.4 で述べる．

[9] この “走り” も動詞の基本連用形であり，名詞化される．

```
大切に たいせつに 大切だ 形容詞 3 * 0 ナ形容詞 21 ダ列基本連用形 7 "代表表記:大切だ/たいせつだ 反義:形容詞:粗末だ/そまつだ"
する する する 接尾辞 14 動詞性接尾辞 7 サ変動詞 16 基本形 2 "代表表記:する/する"
EOS
```

出力例: **JUMAN++**
“響き” が名詞として出力されている.

```
% echo "音の響きを大切にする" | jumanpp
音 おん 音 名詞 6 普通名詞 1 * 0 * 0 "代表表記:音/おん 漢字読み:音 カテゴリ:抽象物"
@ 音 おと 音 名詞 6 普通名詞 1 * 0 * 0 "代表表記:音/おと 漢字読み:訓 カテゴリ:抽象物"
の の の 助詞 9 格助詞 1 * 0 * 0 NIL
響き ひびき 響き 名詞 6 普通名詞 1 * 0 * 0 "代表表記:響き/ひびきv 連用形名詞化:形態素解析"
を を を 助詞 9 格助詞 1 * 0 * 0 NIL
大切に たいせつに 大切だ 形容詞 3 * 0 ナ形容詞 21 ダ列基本連用形 7 "代表表記:大切だ/たいせつだ 反義:形容 詞:粗末だ/そまつだ"
する する する 接尾辞 14 動詞性接尾辞 7 サ変動詞 16 基本形 2 "代表表記:する/する"
EOS
```

連語の廃止
JUMAN では特殊な単語の並びや，解析を誤りやすい単語の並びを連語として登録し，解析に用いていた．JUMAN で連語を用いて解析された箇所には意味情報に “連語” が付加されていたが，JUMAN++では解析に連語を用いないため，該当箇所では JUMAN とは出力結果が異なる．以下に例を示す.

出力例: **JUMAN**

```
% echo "比べるべくもない" | juman
比べる くらべる 比べる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:比べる/くらべる"
べく べく べし 助動詞 5 * 0 助動詞く型 30 基本連用形 3 "連語"
も も も 助詞 9 副助詞 2 * 0 * 0 "連語"
ない ない ない 接尾辞 14 形容詞性述語接尾辞 5 イ形容詞アウオ段 18 基本形 2 "連語"
EOS
```

出力例: **JUMAN++**

```
% echo "比べるべくもない" | jumanpp
比べる くらべる 比べる 動詞 2 * 0 母音動詞 1 基本形 2 "代表表記:比べる/くらべる"
べく べく べし 助動詞 5 * 0 助動詞く型 30 基本連用形 3 NIL
も も も 助詞 9 副助詞 2 * 0 * 0 NIL
ない ない ない 接尾辞 14 形容詞性述語接尾辞 5 イ形容詞アウオ段 18 基本形 2 "代表表記:ない/ない"
EOS
```

数詞の読み表記の変更

JUMAN では数詞に対して読みを付与していたが，JUMAN++ではこれをやめ，読みの欄には表層形が出力されるようになった．以下に例を示す．

出力例: **JUMAN**

```
% echo "１２３４" | juman
１２３４ いちにさんよん １２３４ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
EOS
```

出力例: **JUMAN++**

```
% echo "１２３４" | jumanpp
１２３４ １２３４ １２３４ 名詞 6 数詞 7 * 0 * 0 "カテゴリ:数量"
EOS
```

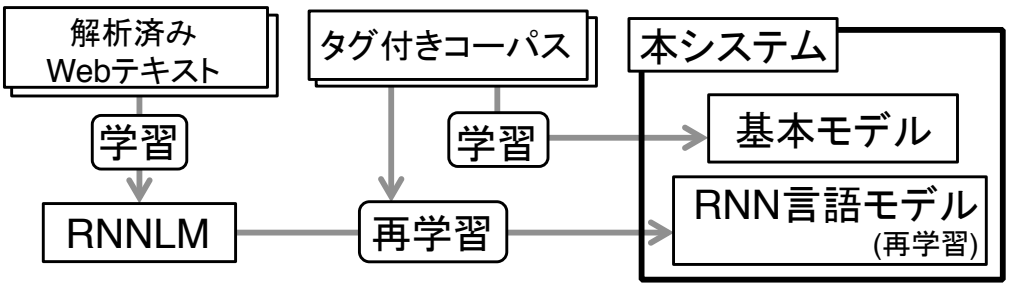

図 2: モデルの訓練フロー

5 JUMAN++で利用するモデルの訓練

JUMAN++ では，品詞間の連接に対するスコア等をコーパスから学習する．通常はシステム標準モデルの利用で十分であるが，短いひらがな語等，解析に対する副作用が大きい形態素を辞書に追加した際や，解析する対象のドメインが新聞や Web と大きく異なる場合には再度訓練を行う必要が生じる．ここではモデルの訓練を行う手順について解説を行う．

本システムでは，特徴量スコアを与える基本モデルと，言語モデルのスコアを与える言語モデルをそれぞれコーパスから訓練する．基本モデルの訓練には人手でアノテーションされたコーパス（タグ付きコーパス），言語モデルの訓練には自動的に解析したコーパスとタグ付きコーパスの両方を用いる（図 2）．これらのコーパスの詳細は，3.3 節を参照のこと．

基本モデルの訓練には Exact Soft Confidence-Weighted Learning [5] を用いる．学習アルゴリズム，および素性等の詳細は文献 [5, 6, 7] を参照のこと．

5.1 訓練コーパスの準備

京都大学テキストコーパス，京都大学ウェブ文書リードコーパスに含まれる，knp 形式のファイルを変換し，本システムの訓練に用いる．この変換は以下のコマンドで行う. ここで用いているスクリプトは配布アーカイブを展開した先の `script` ディレクトリにある．

```
$ cat xxxx.knp ... yyyy.knp | ruby script/corpus2train.rb > train.fmrp
```

以下の手順では，この変換済み訓練データ（`train.fmrp`）を用いてモデルの訓練を行う．

5.2 基本モデルの訓練

JUMAN++の訓練は実行ファイル `jumanpp` に “`--train`” オプションを付けて訓練用モードとして起動することにより行う．その他の訓練に関するオプションを以下に示す．

訓練に関するオプション

`-t, --train` :	訓練に使用する訓練データを指定する
`-i` :	訓練のイタレーション回数を指定する (default: 10)
`-o, --outputmodel` :	モデルの出力先を指定する (default: output.mdl)
`-C` :	Exact Soft Confidence-Weighted のパラメタ C を指定する (default: 1.0)
`-P` :	Exact Soft Confidence-Weighted のパラメタ ϕ を指定する (default: 1.65)
`-B, --beam` :	訓練時のビーム幅を指定する (default: 5)
`--output-intermediate-model` :	各イタレーションの終了時にモデルファイルを出力する

訓練したモデルで，リソースファイルディレクトリにある weight.mdl を置き換えることで，訓練したモデルを利用し，解析を行うことができる．以下に，モデルの訓練および訓練したモデルのインストールの手順を示す．

```
% jumanpp --train train.fmrp --outputmodel trained.mdl
% sudo rm /usr/local/share/jumanpp/weight.mdl.map
% sudo cp trained.mdl /usr/local/share/jumanpp/weight.mdl
```

（※ /usr/local/ 以下に JUMAN++をインストールした場合）

訓練時の出力

訓練中はイタレーションごとに次のように途中経過が出力される．

```
ITERATION:0
50475/50476 avg:0.0202897 loss:0
```

それぞれ，イタレーション番号，訓練に利用した文数/コーパス中の文数， イタレーション内での平均ロス, 最後に解析した文に対するロスを意味する．“--output-intermediate-model” オプションを指定した場合，各イタレーションの終了時に，その時点でのモデルファイルを作成する．

5.3 部分アノテーションを用いた訓練

実際に形態素解析を利用するうえで，解析誤りが生じることは避けられないが，解析誤りは発見されしだい，随時修正されることが望ましい．その時，解析を修正するもっとも素朴な方法は，解析誤りのあった文に対して正しい形態素列をアノテーションし，訓練データに追加する方法である．しかし，適切に文をアノテーションする作業は専門的な知識を必要とするため高いコストがかかる．

本システムでは，明らかな解析の誤りについては専門的な知識がなくとも修正できるようにするため，一部の形態素の境界のみを人手で与えて訓練しなおし，誤った解析を修正することができる．ここでは，テキストの一部に対して形態素境界を明示的に表すことを部分アノテーションと呼ぶ．

部分アノテーションを利用した訓練は２つのステップで行う．（１）部分アノテーションを用いて解析を行う．（２）解析した結果を訓練データに追加し，モデルを再度訓練する．以下で，それぞれのステップについて説明する．

部分アノテーションを用いた解析

“--partial” オプションをつけて JUMAN++を起動した場合，部分アノテーションを利用した解析を行うモードになる．このモードでは，明示的に形態素境界として指定したい箇所にタブ（\t）を挿入することで，挿入した箇所が必ず形態素境界として解析される．また，タブに囲まれた範囲は必ず一形態素として解析される．

部分アノテーションを用いた解析の例

例えば，JUMAN++ で “この実は発芽しない” という文の解析を次のように誤るとする．

```
echo "この実は発芽しない" | jumanpp
この この この 指示詞 7 連体詞形態指示詞 2 * 0 * 0 NIL
実は じつは 実は 副詞 8 * 0 * 0 * 0 "代表表記:実は/じつは"
発芽 はつが 発芽 名詞 6 サ変名詞 2 * 0 * 0 "代表表記:発芽/はつが カテゴリ:抽象物"
し し する 動詞 2 * 0 サ変動詞 16 基本連用形 8 "代表表記:する/する 付属動詞候補（基本） 自他動詞:自:成る/なる"
ない ない ない 接尾辞 14 形容詞性述語接尾辞 5 イ形容詞アウオ段 18 基本形 2 "代表表記:ない/ない"
EOS
```

部分アノテーションを用いて解析を行うことで，このような誤りを修正することができる．

```
$ echo "この実\t は発芽しない" | jumanpp --partial
この この この 指示詞 7 連体詞形態指示詞 2 * 0 * 0 NIL
実 み 実 名詞 6 普通名詞 1 * 0 * 0 "代表表記:実/み 漢字読み:訓 カテゴリ:植物-部位"
は は は 助詞 9 副助詞 2 * 0 * 0 NIL
発芽 はつが 発芽 名詞 6 サ変名詞 2 * 0 * 0 "代表表記:発芽/はつが カテゴリ:抽象物"
し し する 動詞 2 * 0 サ変動詞 16 基本連用形 8 "代表表記:する/する 付属動詞候補（基本） 自他動詞:自:成る/なる"
ない ない ない 接尾辞 14 形容詞性述語接尾辞 5 イ形容詞アウオ段 18 基本形 2 "代表表記:ない/ない"
EOS
```

配布アーカイブ中に上記の事例を含む部分アノテーションのサンプルデータ `sample/part-sample.txt` を同梱している．

部分アノテーションを用いた再訓練

部分アノテーションを用いて解析した結果を訓練データ用の形式に変換し，タグ付きコーパスとマージして，再度訓練する．以下にその手順を示す．訓練したモデルは 5.2 節と同様にインストールし，解析に利用できる．

```
% cat sample/part-sample.txt | jumanpp --partial | ruby script/corpus2train.rb
  > partial.fmrp
% cat train.fmrp partial.fmrp > part_train.fmrp
% jumanpp --train part_train.fmrp --outputmodel part_trained.mdl
```

5.4 言語モデルの訓練

本システムで用いる言語モデルでは，形態素の原形と品詞のみに注目し，形態素の並びに対してスコアを与える．この言語モデルの訓練には Faster RNNLM (HS/NCE) toolkit [10] を利用してい

[10] https://github.com/yandex/faster-rnnlm

る．Faster RNNLM (HS/NCE) toolkit は大規模なテキスト，および大規模な語彙のもとで RNN 言語モデル を訓練することを目的とした実装である．

Faster RNNLM toolkit の訓練には解析済みのテキストを用いる．訓練に利用するフォーマットは，一行が一文を表し，各行に半角スペースで区切った形態素列を記述する．形態素は原形と品詞を ‘_’ で連結して表記する．以下に例を示す．

参考_名詞 に_助詞 する_動詞 くださる_接尾辞 な_助詞 。_特殊
プレゼント_名詞 と_助詞 する_動詞 購入_名詞 する_動詞 ます_接尾辞 。_特殊

JUMAN 形式の解析済みテキストから，言語モデル訓練用の形式へ変換するには以下のスクリプトを用いる．このスクリプトにより，未定義語として解析された形態素は，推定された品詞に置き換えられて出力される．どちらのスクリプトも配布アーカイブ中の `script` ディレクトリ内にある．

```
% cat corpus.txt | jumanpp | ruby script/corpus2train.rb |
  ruby script/fullmrp2basep.rb > data_for_LM.txt
```

以下に，Faster RNNLM toolkit を用いて言語モデルを訓練するコマンドの例を示す．本システムで利用する言語モデルは`--nce` オプションと`--direct` オプションを付けて訓練したモデルである必要がある．また，あらかじめ `data_for_LM.txt` を訓練データ `data_for_LM.train`，Validation 用データ `data_for_LM.valid` に分割しておく．下記の例では，`lang.mdl`，`lang.mdl.nnet` が訓練済みの言語モデルとして出力される．これらのモデルはリソースディレクトリ内の同名のファイルに上書きすることで利用できる．

```
% faster-rnnlm --rnnlm lang.mdl --train data_for_LM.train
  --valid data_for_LM.valid --nce 22 --hidden 100 --direct 100
  --direct-order 3 -bptt 4 --use-cuda 1 -independent
```

オプションの詳細等，Faster RNNLM toolkit の詳しい利用方法については，同 toolkit に付属の readme を参照して欲しい．

6　今後の開発予定

今後は以下の項目について開発を行う．

- 動作の高速化
- Windows(Cygwin) への対応
- 辞書・モデルの改良 (随時公開予定)
- 中国語対応

参考文献

[1] 益岡隆志，田窪行則：『基礎日本語文法』くろしお出版，1989.

[2] 妙木裕，松本裕治，長尾眞「汎用日本語辞書および形態素解析システム」情報処理学会第 42 回全国大会予稿集，1991.

[3] 中村俊久，黒橋禎夫，長尾真「部分文字列情報の利用による日本語単語の高速検索」情報処理学会自然言語処理研究会NL-101, 1994.

[4] 山地治，黒橋禎夫，長尾真「連語登録による形態素解析システム JUMAN の精度向上」言語処理学会 第 2 回年次大会，1996.

[5] Jialei Wang, Peilin Zhao and Steven C.H. Hoi Exact Soft Confidence-Weighted Learning Proceedings of 29th International Conference on Machine Learning，2012.

[6] Hajime Morita, Daisuke Kawahara and Sadao Kurohashi Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model Proceedings of EMNLP 2015: Conference on Empirical Methods in Natural Language Processing，2015.

[7] 森田一，黒橋禎夫「RNN 言語モデルを用いた日本語形態素解析の実用化」情報処理学会 第 78 回全国大会，2016.

付録

A　文法定義ファイルの記法の定義と例

文法定義ファイルの記法を BNF によって示す．以下,〈···〉は非終端記号名を表わす.〈#品詞名〉のように#が付けられた非終端記号はユーザによって定義される具体的な名前を表わすと仮定する．NIL は空リストではなく，空系列を表わす.

A.1　品詞分類定義ファイル (JUMAN.grammar) の記述

A.1.1　品詞分類定義ファイルの個々の項目の記述

〈品詞分類定義ファイル項目〉　::=　（〈品詞〉）|（〈品詞〉（〈品詞細分類の並び〉））

〈品詞〉　::=　（〈#品詞名〉）|（〈#品詞名〉 %）
〈品詞細分類の並び〉　::=　〈細分類〉〈品詞細分類の並び〉
〈品詞細分類〉　::=　（〈#品詞細分類名〉）|（〈#品詞細分類名〉 %）

A.1.2　品詞分類定義ファイルの定義例

```
((動詞 %))                          ; 動詞は品詞名であり，活用する.

((名詞)                             ; 名詞は品詞名であり,
        ((普通名詞)                 ; 普通名詞，サ変名詞，固有名詞 ... に
         (サ変名詞)                 ; 細分類される.
         (固有名詞)
         (数詞)
         (形式名詞)
         (副詞的名詞)))

((接尾辞)
        ((名詞性述語接尾辞)
         (名詞性名詞接尾辞)
         (名詞性名詞助数辞)
         (形容詞性述語接尾辞 %)     ; 形容詞性述語接尾辞は活用する.
         (形容詞性名詞接尾辞 %)
         (動詞性接尾辞 %)))
```

A.2 活用関係定義ファイル (JUMAN.kankei) の記述

A.2.1 活用関係定義ファイルの個々の項目の記述

〈活用関係定義ファイル項目〉 ::= （〈品詞〉（〈活用型の並び〉））

〈品詞〉 ::= （〈#品詞名〉）| （〈#品詞名〉〈#品詞細分類名〉）
〈活用型の並び〉 ::= 〈#活用型名〉 | 〈#活用型名〉〈活用型の並び〉

A.2.2 活用関係定義ファイルの定義例

```
((形容詞)                    ; 形容詞はイ形容詞とナ形容詞という活用型をもつ
        (イ形容詞
         ナ形容詞))

((助動詞)
        (イ形容詞
         ナ形容詞
         判定詞
         無活用型
         助動詞ぬ型
         助動詞だろう型
         助動詞そうだ型))

((接尾辞 形容詞性述語接尾辞)  ; 接尾辞の中で形容詞性述語接尾辞に細分類される
        (ナ形容詞              ; 形態素はナ形容詞またはイ形容詞という活用型を
         イ形容詞))            ; もつ
```

A.3 活用定義ファイル (JUMAN.katuyou) の記述

A.3.1 活用定義ファイルの個々の項目の記述

〈活用定義ファイル項目〉 ::= (〈#活用型名〉 (〈活用形対の並び〉))

〈活用形対の並び〉 ::= 〈活用形対〉| 〈活用形対〉〈活用形対の並び〉
〈活用形対〉 ::= (〈#活用形名〉〈語尾表示〉)
〈語尾表示〉 ::= 〈#語尾〉 | *

A.3.2 活用定義ファイルの定義例

```
(母音動詞
    ((語幹              *    )
     (基本形            る   )
     (未然形            *    )
     (意志形            よう)
     (命令形            ろ   )
     (命令形            よ   )
     (基本条件形        れば)
     (基本連用形        *    )
     (タ形              た   )
     (タ系条件形        たら)
     (タ系連用テ形      て   )
     (タ系連用タリ形    たり))
)

(サ変動詞
    ((基本形            する  )
     (未然形            さ    )
     (意志形            しよう)
     (命令形            しろ  )
     (命令形            せよ  )
     (基本条件形        すれば)
     (基本連用形        し    )
     (タ形              した  )
     (タ系条件形        したら)
     (タ系連用テ形      して  )
     (タ系連用タリ形    したり))
)
```

B　意味情報の具体例

B.1　代表表記

この枠組みの中では，次のような現象を扱っている．

漢字と平仮名，送り仮名

漢字とするか平仮名とするか，また，送り仮名のバリエーション．その組み合わせ．

例）拳銃 けん銃 拳じゅう けんじゅう
表す 表わす あらわす
落とす 落す おとす

漢字別表記

例）狩人／猟人 朝日／旭
色取る／彩る 哀れむ／憐れむ
綺麗だ／奇麗だ 気掛りだ／気懸かりだ

動詞でどこまでまとめるかは微妙な問題であり，一般の国語辞典などでは意味の一致の観点からはまとめすぎである．例えば「下りる/降りる」「下ろす/降ろす」はまとめないこととした．

カタカナ表記のバリエーション

ソフトウェアとソフトウエアのようなカタカナ表記のバリエーションに対しても代表表記を設けた．これは，カタカナ語のコーパス中の出現頻度をもとに，表記バリエーションの自動認識と複合語の自動分割を行い，その結果を人手でチェックして整理した．

日本語固有語のカタカナ表記

動物，植物，食べ物などで，日本語固有語であってもカタカナ表記されるものについて，代表表記のもとに整理した．

例）大根 だいこん ダイコン
餃子 ぎょうざ ギョウザ ギョーザ (最後はカタカナ表記バリエーション)

上記のカテゴリに関わらず，カタカナ表記があるものをまとめた

例）溝 ミゾ みぞ
眼鏡 メガネ めがね
奴 ヤツ やつ

代表表記は，原則として，新聞記事での高頻度表記としたが，この選択には強いこだわりはない（妥当性を主張するものではない）．重要なことは，同じ語の表記集合がまとめられ，これを通してテキストマッチングなどが適切に行われることであり，代表表記はこの集合の ID の役割を果たすものである．しかし，ID を数字などにすることは人間にとって管理しやすいものではないので高頻度の表記を採用した．

B.2　意味カテゴリ

人　「幽霊」「神」「河童」「人魚」「半魚人」などの人間に近い生物（架空の生物も含む）を表す単語も含む.

学生 先生 歌手 父 兄 大人 子供 赤ちゃん 私 僕 あいつ 我々 誰 何者 個人 主人公 跡取り 逸材 語り手 手先 神 幽霊 故人 河童 人魚 半魚人 妖精

組織・団体　「コンビ」や「トリオ」などの複数人数を表す単語も含む.「我々」などの複数人数を表す人称代名詞は＜人＞に含める.

政府 軍 国家 党 委員会 組合 企業 マスコミ 警察 家族 チーム クラス コンビ カップル 一座 一同

動物　「怪獣」などの架空の動物を表す単語も含む.

犬 猫 鳥 ふな めだか 金魚 かえる ほ乳類 虫 さなぎ 恐竜 三葉虫 細菌 微生物 竜 しゃちほこ 怪獣 ペット 愛犬 害虫 天敵 類人猿

植物

木 草 桜 紫陽花 バラ ひまわり 朝顔 チューリップ 稲 盆栽 牧草 一年草 大木 果樹 針葉樹 こけ カビ 切株 国花 まりも

動物-部位　人間の部位も含む.「垢」や「かさぶた」など体に付着している物質はこれに含めるが,「涙」や「血」などの分泌物は＜自然物＞に含める.

手 皮膚 傷 毛 指紋 肉 アラ 尾 羽 くちばし 内臓 筋肉 ほくろ ひげ 爪 たてがみ 甲羅 うろこ 角 牙 骨 関節 かさぶた

植物-部位　＜動物-部位＞と同様に「樹液」などは＜自然物＞に含める.

葉 茎 枝 根 実 種 花片 年輪 落葉 花粉 わら 樹皮 おしべ 菌糸 葉緑素 胞子 葉脈 落葉 つた

人工物-食べ物　「りんご」や「さんま」などの人工物でない食べ物も含む.

料理 アイス パン 菓子 焼き肉 ラーメン 豆腐 コーヒー ワイン 調味料 ソース ごちそう お弁当 主食 大好物 冷凍食品 風邪薬 エサ

人工物-衣類　「眼鏡」や「コンタクト」などは＜人工物-その他＞に含める.

セーター ワイシャツ ズボン スカート レインコート ネクタイ マフラー 靴 アクセサリー 手袋 ハチマキ ベルト

人工物-乗り物

自動車 飛行機 船 自転車 三輪車 ヘリコプター 御輿 エレベーター エスカレーター ソリ 車椅子 観覧車 いかだ 宇宙船 ロケット

人工物-金銭

給料 ボーナス 借金 運賃 謝礼 切手 馬券 つけ お宝 金券 金貨 小判 食券 カード 埋蔵金 遺産 保険金 賄賂 チップ 祝儀 香典 お年玉

人工物-その他　上の４つの人工物のカテゴリに属さない単語．ただし建築物は＜場所-施設＞に含める．

鉛筆 消しゴム 箸 椅子 テーブル コップ おもちゃ カメラ 時計 鏡 傘 農薬 石鹸 布団 洗濯機 テレビ カーテン 電球 ランドセル 眼鏡

自然物　「山」などは＜場所-自然＞,「津波」などは＜現象-自然＞に含める．(「宇宙」は＜場所-自然＞,「星」は＜自然物＞とする)「アルコール」や「アミノ酸」などの物質名を表す単語も含む．

石 岩 砂 泥 空気 雲 湯気 水滴 炭素 石油 太陽 月 隕石 灰 ちり ほこり けむり 鉱物 さび 宝石 元素 原子 電子 イオン 地下水

場所-施設　＜「門」「天安門」などはこのカテゴリに含むが,「窓」「塀」などは＜場所-施設部位＞とする

ビル マンション 駅 港 遊園地 プール 橋 道路 公園 部屋 台所 風呂 トイレ 庭 門 屋台

場所-施設部位　建築物の部位である単語．ただし,「トイレ」「台所」などの「部屋」を表す単語は＜場所-施設＞とする．

天井 床 壁 屋根 窓 塀 廊下 階段 縁側 席 下座 ベランダ

場所-自然

山 海 池 沼 空 島 森 林 ジャングル がけ 地層 海底 水脈 水溜まり 平野 半島 岬 岸 草むら 砂漠 野原 山頂 火口 山脈 日向 日陰 茂み

場所-機能

上 下 左 右 中 外 前 うしろ 奥 表面 ふち 境界 方向 範囲 頂点 辺り そば あいだ 端 隅角 隣 先 東 西 向こう 起点 終点

場所-その他

都市 村 里 首都 先進国 海外 全国 世界 天国 生産地 農地 畑 牧場 領土 空き地 隣国 選挙区 いなか 北極 戦場 人込み 行き止まり 現場 吹き溜まり 踏み場 ゴール 本塁

抽象物　抽象物の中でもう一段階細かいカテゴリの検討を行ったが，意味情報としては＜抽象物＞だけを与えている．下記の具体例は細かいカテゴリごと．

《現象-自然》雨 風 霧 霞 地震 津波 噴火 高潮 つらら 高気圧 音 光 開花 紅葉 発芽 刺激 反射 蒸発 火 酸化 凝固 昇華 乾燥 夕焼け 発光 日差し スペクトル

《現象-生命》あくび しゃっくり いびき 出産 病気 風邪 癌 死 頭痛 睡眠 けが 貧血 遺伝 呼吸 排泄 老化 孵化 回復 羽化 声 命

《動作》(物理的な動作を伴った単語) 運動 仕事 電話 コピー 拍手 転倒 移動 到着 通過 貫通 落下 閉鎖 包囲 炊事 洗濯 使用 採取 運転 飲食

《出来事》(出来事を表す単語) 戦争 大会 試合 事故 事件

《様子》(「混乱」や「緊張」などの様子そのものを表す単語,「味」などの述語が付属して様子を表す（「味が良い」など）単語,「長所」などの様子を表す文字（「長」）が付属して様子を表す単語) 混乱 緊張 身勝手 表情 実情 寝相 味 長所 短所 欠点 汚点

《気持ち》(明らかに感情が伴っている単語，また「食欲」なども含む) 愛情 勇気 敵意 敬愛 あこがれ ためらい 嫌悪 遠慮 良心 自信 執着 我慢 同情 感謝 おごり 誠意 忠節 信用 心服 蔑視 哀悼 恥 決心 尊敬 賞賛

《制度・規則》法律 掟 保険 条約 校則 約束 方式 法則 文法 書式 流儀 しきたり マナー 権利 義務 戒律 摂理 ルール 鉄則 規定 政令 契約 公約

《知的生産物》(「物語」「演劇」「学問」「伝統」などの知的生産物を表す単語，ただし，＜制度・規則＞に属する単語は除く) 言語 ニュース 噂 文章 音楽 映像 演劇 学問 芸術 文化 伝承 宗教 名前 故事ことわざ

《力》(「能力」や「五感」なども含む) 能力 魅力 五感 引力 圧力 火力 エネルギー 威力 底力 動力 念力

《抽象-機能》(抽象概念の中でも特に機能的な単語) 理由 原因 結果 目的 関係 対象 条件 基本 基礎 例 内容 相互 概念 対応 由来 基準

《抽象-その他》

思考 評価 誇張 協力 想像 予測 注意 成立 証明 安定 優遇 奨励 遵守 放任 要求 推薦 誘惑 交渉 保証 承諾 許可 妥協 逃げ場 作戦 戦略 思想 思い出 本音 意見 仮説 民意 容疑 主義 主張 魂胆 異心 謀略 案 解答 方針 意志 人心 所存 予想 打算 要約 知識 広告 指示 連絡 証言 告示 評判 証拠 勝利 敗北 捌け口 とっかかり

形・模様

円 球 線 正方形 直角 縦じま ぶち まだら シルエット 凹凸 粒 列 大型 小型 フォーム 十字 いびつ スパイラル ジグザグ 流線形

色

赤 青 黄 緑 ピンク 白 黒 ベージュ

数量　「和」や「差」などの数量の関係や，単位もこれに含む．

複数 多数 和 差 比 速度 番号 ボリューム 面積 余分 速度 勾配 頭数 沢山 無数 数多 以上 以下 回 倍 些細 半分 最大 最小

時間

年 月 朝 晩 時刻 今日 休日 未来 過去 期間 季節 時期 瞬間 チャンス 途端 間髪 順序 先後 永遠 締切 一生 王朝 世代 史上 将来

以下に，複数の意味カテゴリに属する，または判断が難しいと考えられる単語の具体例とその見解を示す．

- 「魚」「野菜」などの＜動物＞や＜植物＞とも＜人工物-食べ物＞とも考えられる単語
  例えば「金魚」という単語には＜動物＞のカテゴリだけを付与するが,「さんま」という単語には＜動物＞と＜人工物-食べ物＞の両方のカテゴリを付与する．
- 「学校」「会社」「市役所」などの＜組織・団体＞とも＜場所-組織＞とも考えられる単語
  両方のカテゴリを付与する．
- 「間」などの＜場所-機能＞とも＜時間＞とも考えられる単語
  両方のカテゴリを付与する．
- 「青二才」「意気地無し」などの＜人＞とも＜様子＞とも考えられる単語
  ＜様子＞のカテゴリだけを付与する．
- 「罠」「右腕」「圧力」「像」などの実体を表す単語が比喩的に抽象概念も表す単語
  例えば「罠」という単語には＜人工物-その他＞のカテゴリだけを付与し,「右腕」という単語には＜人＞と＜動物-部位＞両方のカテゴリーを付与する．

B.3　ドメイン

以下，各ドメインの基準等を具体例とともに説明する．

文化・芸術　文化，芸術，芸能に関わる単語.「文学」や「美術」などの抽象物を表す語も,「書籍」や「ギター」,「女優」などの具体物を表す語もこのドメインに含める．また,「教会」や「仏壇」などの宗教関係の語もこのドメインに含める．ただし，葬儀関係の語は，＜文化・芸術＞と＜家庭・暮らし＞の両方に含めることとする．

写真 映画 音楽 文学 アニメ 曲 映画 デザイン 展示 映像 美術 芸術 コンサート ピアノ 演出 劇場 楽器 レ コード 芸能 劇 アーティスト 教会 仏壇

レクリエーション　遊びや趣味，娯楽に関わる語．ただし，趣味や娯楽の対象となりうるものであっても，＜レクリエーション＞以外のドメインのいずれかと強く関連するものは除く．例えば，「音楽」は趣味や娯楽となりうるが，＜文化・芸術＞と強く関連するので除外する．同様に，「ゴルフ」は＜スポーツ＞と強く関連するので除外する．一方，囲碁将棋関係は，＜レクリエーション＞と＜スポーツ＞の両方に含める．

遊園地 ゲーム 遊ぶ 旅行 温泉 観光 旅 趣味 パーティー おもちゃ 花火 カラオケ 競馬

スポーツ　スポーツや格闘技に関する語．「サッカー」などのスポーツ名はもちろん，「ドリブル」「ホームラン」などの動作に関わる語や，「ラケット」「土俵」などの道具に関わる語も含める．囲碁将棋関係は，＜レクリエーション＞と＜スポーツ＞の両方に含める．

選手 試合 スポーツ 野球 サッカー レース ボール スキー ゴルフ 競技 対戦 決勝 投手 トレーニング 予選

健康・医学　健康，医療，衛生に関わる単語．このドメインでも，「診察」「予防」などの抽象的なものから，「包帯」「医師」などの具体的なものまで，健康，医療，衛生に関わる単語を広く含める．

医療 病院 患者 感染 癌 医師 ウイルス 診断 症状 看護手術 痛む 予防 薬 風邪 医学 栄養 診療 医者 療法 傷 疾患 ダイエット 歯科 臨床 移植 外科 体重 治る 病気

家庭・暮らし　日常生活に関わる単語．朝起きて，歯を磨き，外出し，帰宅して，風呂に入って寝る．これらの合間に掃除や洗濯，買いもの等の家事を済ませ，子供の面倒を見る．こういった活動に関わる語が全てこのドメインに含まれる．また，「父」「母」「兄弟」「親戚」などの人間関係を現す語もこのドメインに含める．これら以外にも，多くの人がその人生で直面する，結婚や出産，引っ越しなども含まれる．ただし，他のドメインと強く関連する語は除外する．例えば「朝食」は日常生活の一部だが＜料理・食事＞ドメインに含め，このドメインには含めない．「出勤」や「通学」はそれぞれ＜ビジネス＞，＜教育・学習＞に含める．葬儀関係の語は，＜家庭・暮らし＞と＜文化・芸術＞の両方に含めることとする．「インターネット」「パソコン」は＜科学・技術＞とするが，「メール」「ホームページ」は＜家庭・暮らし＞とする．

結婚 出産 引越し 家 家族 住宅 家庭 風呂 暮らす ゴミ トイレ 買い物 保育 洗う 水道 掃除 帰宅 散歩 実家 玄関 世帯 家具

料理・食事　料理，食事，食べ物に関する語．料理名や料理法はもちろん，「箸」「フォーク」などの道具，「レストラン」「料亭」などの店，「炊く」「煮る」などの動作も含める．「たばこ」などの嗜好品もこのドメインに含める．

菓子 食品 食事 料理 箸 味噌 夕食 皿 カフェ 醤油 食べ物 昼食 朝食 炊く 煮る レストラン 冷蔵

交通　陸海空を問わず，「車」「船」「飛行機」などの乗り物，「信号」「標識」などの交通に関わる設備機器，「駐車」「離陸」などの動作，「運転手」「乗客」などの人物，「歩道」「駅」などの場所，その他関連する単語．

駅 道路 交通 運転 空港 航空 鉄道 信号 路線 国道 地下鉄

教育・学習　教育，勉強，学校に関わる語.「先生」「生徒」などの人物や「算数」「国語」などの科目,「成績」「留学」などの抽象的な語をこのドメインに含める．ただし，以下の＜科学・技術＞により強く関連すると考えられる語，例えば「論文」「研究」「学会」などは除外する．

たし算 教育 先生 授業 生徒 学ぶ 勉強 卒業 小学校 教室 テスト 教授 レポート 受験 教科 入学 留学 成績 学科 数学 学年 国語 塾 教材 演習 校長 大学

科学・技術　科学，技術，研究，開発，その他各種理工系専門分野に関わる語.「博士」「学者」などの人物,「解析」「実験」などの活動,「原子」「アンペア」「変数」などの専門用語的なものなどが含まれる.「メール」「ホームページ」は＜家庭・暮らし＞とするが,「インターネット」「パソコン」は＜科学・技術＞とする．

論文 研究 開発 データ 通信 科学 ネットワーク 電子 実験 コンピューター 分析 エネルギー 機械 学会 解析 理論 工学 博士 ロボット 原子 発電 回路 電波 学者

ビジネス　ビジネスあるいは仕事，経済に関する語.「販売」「経営」「契約」などの活動,「社長」「スタッフ」などの人物,「資本」「ニーズ」などの経済用語的なものなどがこのドメインに属する．ただし，個別的な仕事に関する語，例えば「農業」や「水産」などの語は含めない．あくまでビジネスや経済一般に関わる語のみを対象とする．

仕事 企業 販売 商品 経営 価格 産業 株式 市場 働く 営業 注文 メーカー 職員 組合 投資 広告 社長 資金 コスト 就職 株 職業 顧客 資本 需要 証券 退職 貿易

メディア　メディアあるいは報道機関，ジャーナリズムに関する語.「報道」「社説」「論評」などの抽象物以外にも,「キャスター」「アナウンサー」などの人物や「テレビ」「新聞」などの具体物も含める．

ニュース ラジオ テレビ 記事 放送 新聞 番組 メディア 報道 記者 マスコミ

政治　政治あるいは行政，司法，警察あるいは犯罪，福祉，人権，戦争などに関する語.「役所」「兵器」などの具体物,「民主」「法律」などの抽象物,「大統領」「判事」などの人物,「デモ」「投票」などの活動等が含まれる．一方,「合議」「対案」「代案」などは＜政治＞的なニュアンスもあるが，社会生活全般に関わるので＜ドメイン無し＞とする．

司法 行政 政府 軍 税 法律 議員 金融 国家 選挙 警察 裁判 財政 大臣 議会 党 人権 厚生 国会 交渉 テロ 役所 憲法 首相 民主 政権 都道府県 訴訟 逮捕 デモ 国連

名詞であっても，以下のように特定のドメインとの関係がない・薄いものはドメインを与えない．なお，下記の自然，天候などの分類は見通しをよくするために与えたもので，これらに分類される語に常にドメインを与えないという意味ではない.「社長」は人間だが＜ビジネス＞に,「駅」は建造物だが＜交通＞に含める．

自然： 岩 河川 宇宙 津波
天候： 雨 台風 寒気 日照り
人間： 彼女 私 太郎 善人

身体： 目 腕 筋肉 声
感情： 愛 憎悪 不安 歓喜
建造物：ビル 小屋 倉庫
その他：白 明後日 物品 北西 諸々 少量 弱点 合議 対案 代案

一方，複数のドメインに関連しうる語は多いが，いずれかがドミナントである場合はそのドメインだけを与え，複数に同程度に関連すると考えられる語であれば複数ドメインを与える．複数のドメインを与えている例は以下のようなものである．

大学院 → ＜教育・学習＞ ＜科学・技術＞
円高 → ＜ビジネス＞ ＜政治＞
薬膳 → ＜健康・医学＞ ＜料理・食事＞
登山 → ＜スポーツ＞ ＜レクリエーション＞

B.4 固有名詞

人名 (約 4,000 語) Web の自動解析結果（もちろん解析誤りも含まれる）をもとに，日本人の姓の頻度上位 1,500 位まで，名の 2,000 位までを抽出・登録した．さらに，順位，相対頻度を意味情報として付与した．

山田 → 人名:日本:姓:7:0.00607
太郎 → 人名:日本:名:45:0.00106

英語名は，姓名の区別なく，Web の頻度上位 150 位までを登録した．

ジョン → 人名:外国

日本の地名 (約 50,000 語) 日本の地名については，都道府県，郡，市区町村，町名などの市区町村以下の行政区分を意味情報とともに登録した．

京都 → 地名:日本:府
→ 地名:日本:京都府:市
上京 → 地名:日本:京都府:区
長岡京 → 地名:日本:京都府:市

さらに，地方名，主要な地名（銀座など，意味情報は地区）を登録した．

東北 → 地名:日本:地方
銀座 → 地名:日本:地区

また，主要な山，湖，島について，地名＋山などで解析できないものを若干登録した．

比叡山 → 地名:日本:山
琵琶湖 → 地名:日本:湖

例外的なものとして，東名，関越，名神を施設として登録した．

東名 → 地名:日本:施設

世界の地名 (約 700 語)　世界の地名については，国，首都，米国の州，中国の省，各国の主要都市を登録した．また主要な国の別称，略称等も登録した．

英国 → 地名:国
イギリス → 地名:国:別称:英国
ロンドン → 地名:国:英国:首都
英 → 地名:国:略称:英国
ケンブリッジ → 地名:国:英国:市
カリフォルニア → 地名:国:米国:州
四川 → 地名:国:中国:省

主要な地域，海，山などを登録した．意味情報では複数の国にまたがるものは国を与えず，一国内のものは国を与えた．なお，地域は国をこえるもの，地方は国内で市よりも大きなもの，地区は市よりも小さなものとした．

アジア → 地名:地域
太平洋 → 地名:海
アルプス → 地名:山脈
シベリア → 地名:国:ロシア:地方
マンハッタン → 地名:国:米国:地区
グアム → 地名:国:米国:島
キリマンジャロ → 地名:国:タンザニア:山

例外的なものとして，ホワイトハウス，天安門などを施設として登録した．

ホワイトハウス → 地名:国:米国:施設
天安門 → 地名:国:中国:施設

B.5　見出し語間の意味関係

以下に見出し語間につけた意味関係の具体例を示す．

尊敬動詞・謙譲動詞
仰る → 尊敬動詞:言う
申し上げる → 謙譲動詞:言う

自動詞・他動詞
壊れる → 自他動詞:他:壊す
壊す → 自他動詞:自:壊れる

授受動詞
貸す → 授受動詞:受:借りる

借りる → 授受動詞:授:貸す

反義

増える → 反義:動詞:減る
大きい → 反義:形容詞:小さい

種々の派生

大人びる → 名詞派生:大人
愚痴る → 名詞派生:愚痴
白い → 名詞派生:白
高める → 形容詞派生:高い
小さな (連体詞) → 形容詞派生:小さい