Fumi Package Documentation

NYSOL Version: Ver. 1.2, 2.0

revise history:
October 6, 2014 : first release

August 12, 2015

Contents

Chapter 1

Introduction

1.1 Abstract

The Fumi package is comprised of several commands for the purpose of text mining for Japanese text.

Data mining is mostly deal with quantitative information in structured format. In recent years, advancements in computer performance and data analysis technology have increased the capacity to handle unstructured information. This type of data contains numbers, time and facts. An example of unstructured information in essays written by humans (natural language).

This package contains morphological analyser JUMAN for Japanese, and case structure analyser KNP. The algorithms are developed by Kyoto University Department of Intelligence Science and Technology, Kurohashi and Kawahara lab. Since the results generated are saved in CSV file format, various analytical processes with mcommand can be carried out to create analytical models. Please refer to the official website of JUMAN and KNP for more details.

- Morphological Analyzer System JUMAN http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN
- Case Structure Analyzer System KNP http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP

1.2 Installation

This package is embedded within the nysol package. Please refer to installation of nysol package (http://www.nysol.jp/en/home/install) for more details.

However, please note that the commands contain in this package requires installation of JUMAN and KNP separately.

Please download JUMAN and KNP from the official homepage, and refer to the installation instructions in the `README` and `INSTALL` file.

This fumi package is compatible with JUMAN version 7.0 or higher, and KNP version 4.0 or higher.

1.3 Copyright Information on JUMAN and KNP

The copyright of the Morphological Analyzer JUMAN and Case Structure Analyzer KNP belongs to Kyoto University Graduate School of Informatics, Department of Intelligence Science and Technology, Intelligence media language media Kurohashi and Kawahara lab (http://nlp.ist.i.kyoto-u.ac.jp/). Please refer to the terms and conditions as follows.

1.3.1 JUMAN

1.3.2 KNP

Chapter 2

Text Mining Commands

2.1 mjuman.rb Morphological analysis using JUMAN

Given a document in text file, JUMAN analyzes the morphological grammar and saves the output in CSV format. This command can process multiple text files, and can be executed in multi-processing mode based on compatibility of OS.

2.1.1 Format

```
mjuman.rb I= O= [P=] [mp=] [log=] [-mcmdenv] [--help]
```

`I=`	: Path name of the text document input.
`O=`	: Path name of the analysis results saved as CSV file.
`P=`	: Path name to save the direct output of JUMAN (default do not return output)
`mp=`	:Number of parallel processes. Default is 2.
`log=`	: Log file of processing error from JUMAN.
`-mcmdenv`	: MCMD message containing environment variables.
	: Default returns warning and error message (KG_VerboseLevel=2).
`--help`	: Display help

Example of input file

The structure of text is preferred to include one sentence per line. The character code must be in UTF8 format.

```
子どもはリンゴがすきです。
望遠鏡で泳ぐ少女を見た。
```

Example of output file

Results of morphological analysis in CSV format.

```
aid,sid,tid,word,orgWord,daiWord,yomi,class1,class2,class3,class4,annotation
test.txt,0,0, 子ども, 子ども, 子供, こども, 名詞, 普通名詞,,, 代表表記:子供/こども カテゴリ:人
test.txt,0,1, は, は,, は, 助詞, 副助詞,,,
test.txt,0,2, リンゴ, リンゴ, 林檎, りんご, 名詞, 普通名詞,,, 代表表記:林檎/りんご カテゴリ:植物
:
```

The contents of CSV is based on the output of JUMAN, the description of each item is as follows.

`aid`	: Input file name
`sid`	: Row number （Sentence ID）
`tid`	: Morpheme number （Token ID）
`word`	: Word (original form)
`orgWord`	: Notation in text
`daiWord`	: Representative notation
`yomi`	: Read
`class1`	: Part of speech (level 1)
`class2`	: Part of speech (level 2)
`class3`	: Part of speech (level 3)
`class4`	: Part of speech (level 4)
`annotation`	: Semantic information

2.1.2 Examples

Example 1: Basic Example

File `test.txt` is contained in `text` directory for morphological analyisis. The results are saved in `csv` directory.

```
$ more text/test.txt
子どもはリンゴがすきです。
望遠鏡で泳ぐ少女を見た。
$ mjuman.rb I=text O=csv
```

```
#MSG# KNP: reading text/test.txt
#MSG# JUMAN: MP-2 aid=test.txt sid=0 (sentences:1/2, articles:1/1)
#MSG# JUMAN: MP-2 aid=test.txt sid=1 (sentences:2/2, articles:1/1)
#MSG# JUM2CSV 1/1
#MSG# Elapse: 0.048sec, # of sentences=2, # of articles=1
#MSG#   0.024sec/sentence, 0.048sec/article
#MSG#   mpCount=2, poolSize=1000
#MSG#   maxLen=512Byte, maxSec=30sec, sizeLimit=2000MB
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mjuman.rb I=text O=csv
$ more csv/test.txt
aid,sid,tid,word,orgWord,daiWord,yomi,class1,class2,class3,class4,annotation
test.txt,0,0,子ども,子ども,子供,こども,名詞,普通名詞,,,代表表記:子供/こども カテゴリ:人
test.txt,0,1,は,は,,は,助詞,副助詞,,,
test.txt,0,2,リンゴ,リンゴ,林檎,りんご,名詞,普通名詞,,,代表表記:林檎/りんご カテゴリ:植物
test.txt,0,3,が,が,,が,助詞,格助詞,,,
test.txt,0,4,すきだ,すきです,好きだ,すきです,形容詞,,ナ形容詞,デス列基本形,代表表記:好きだ/
test.txt,0,5,。,。,。,。,特殊,句点,,,
test.txt,1,0,望遠,望遠,望遠,ぼうえん,名詞,普通名詞,,,代表表記:望遠/ぼうえん カテゴリ:抽象物
test.txt,1,1,鏡,鏡,鏡,かがみ,名詞,普通名詞,,,代表表記:鏡/かがみ 漢字読み:訓 カテゴリ:人工物-
test.txt,1,2,で,で,,で,助詞,格助詞,,,
test.txt,1,3,泳ぐ,泳ぐ,泳ぐ,およぐ,動詞,,子音動詞ガ行,基本形,代表表記:泳ぐ/およぐ
test.txt,1,4,少女,少女,少女,しょうじょ,名詞,普通名詞,,,代表表記:少女/しょうじょ カテゴリ:人
test.txt,1,5,を,を,,を,助詞,格助詞,,,
test.txt,1,6,見る,見た,見る,みた,動詞,,母音動詞,タ形,代表表記:見る/みる 補文ト 自他動詞:自:
test.txt,1,7,。,。,。,。,特殊,句点,,,
```

Example 2: Example of results of JUMAN (original)

JUMAN results (original) is saved in `juman` directory.

```
$ more text/test.txt
子どもはリンゴがすきです。
望遠鏡で泳ぐ少女を見た。
$ mjuman.rb I=text O=csv P=juman
#MSG# KNP: reading text/test.txt
#MSG# JUMAN: MP-2 aid=test.txt sid=0 (sentences:1/2, articles:1/1)
#MSG# JUMAN: MP-2 aid=test.txt sid=1 (sentences:2/2, articles:1/1)
#MSG# JUM2CSV 1/1
#MSG# Elapse: 0.054sec, # of sentences=2, # of articles=1
#MSG#   0.027sec/sentence, 0.054sec/article
#MSG#   mpCount=2, poolSize=1000
#MSG#   maxLen=512Byte, maxSec=30sec, sizeLimit=2000MB
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mjuman.rb I=text O=csv P=juman
$ more juman/test.txt
子ども こども 子ども 名詞 6 普通名詞 1 * 0 * 0 "代表表記:子供/こども カテゴリ:人"
は は は 助詞 9 副助詞 2 * 0 * 0 NIL
リンゴ りんご リンゴ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:林檎/りんご カテゴリ:植物
が が が 助詞 9 格助詞 1 * 0 * 0 NIL
すきです すきです すきだ 形容詞 3 * 0 ナ形容詞 21 デス列基本形 29 "代表表記:好きだ/すきだ 反
。 。 。 特殊 1 句点 1 * 0 * 0 NIL
EOS
望遠 ぼうえん 望遠 名詞 6 普通名詞 1 * 0 * 0 "代表表記:望遠/ぼうえん カテゴリ:抽象物"
鏡 かがみ 鏡 名詞 6 普通名詞 1 * 0 * 0 "代表表記:鏡/かがみ 漢字読み:訓 カテゴリ:人工物-その
で で で 助詞 9 格助詞 1 * 0 * 0 NIL
泳ぐ およぐ 泳ぐ 動詞 2 * 0 子音動詞ガ行 4 基本形 2 "代表表記:泳ぐ/およぐ"
少女 しょうじょ 少女 名詞 6 普通名詞 1 * 0 * 0 "代表表記:少女/しょうじょ カテゴリ:人"
を を を 助詞 9 格助詞 1 * 0 * 0 NIL
見た みた 見る 動詞 2 * 0 母音動詞 1 タ形 10 "代表表記:見る/みる 補文ト 自他動詞:自:見える/
。 。 。 特殊 1 句点 1 * 0 * 0 NIL
EOS
```

2.2 mknp.rb Parsing case structure using KNP

Given a document in text file, KNP parses the case structure and saves the output in XML format. This command can process multiple text files, and can be executed in multi-processing mode based on compatibility of OS.

2.2.1 Format

```
mknp.rb I= O= [P=] [mp=] [log=] [-mcmdenv] [--help]
```

`I=`	: Path name of the text document input.
`O=`	: Path name of the analysis results saved as XML file.
`P=`	: Path name to save the direct output of KNP (default do not return output).
`mp=`	: Number of parallel processes. Default is 2.
`log=`	: Log file of processing error from KNP.
`-mcmdenv`	: Display MCMD message containing environment variables.
	: Default returns warning and error message (KG_VerboseLevel=2).
`--help`	: Display help

Example of input file

The structure of text is preferred to include one sentence per line. The character code must be in UTF8 format.

```
子どもはリンゴがすきです。
望遠鏡で泳ぐ少女を見た。
```

Example of output file

Results of case structure analysis is stored in XML file format.

```
<?xml version='1.0' encoding='UTF-8'?>
<article id='test.txt'>
 <sentence id='0' text='子どもはリンゴがすきです。'>
  <chunk id='0' link='2' phraseType='格助詞句' caseType='ガ２格' phrase='子供' phraseTok='子
   <token id='0' class1='名詞' class2='普通名詞' word='子ども' orgWord='子ども' daiWord='子供
   <token id='1' class1='助詞' class2='副助詞' word='は' orgWord='は'/>
  </chunk>
  <chunk id='1' link='2' phraseType='格助詞句' caseType='ガ格' phrase='林檎' phraseTok='リン
   <token id='2' class1='名詞' class2='普通名詞' word='リンゴ' orgWord='リンゴ' daiWord='林檎
   <token id='3' class1='助詞' class2='格助詞' word='が' orgWord='が'/>
  </chunk>
  <chunk id='2' link='-1' phraseType='用言句' phraseTok='すきだ' rawPhrase='すきです。' phrase
   <token id='4' class1='形容詞' class3='ナ形容詞' class4='デス列基本形' word='すきだ' orgWord
   <token id='5' class1='特殊' class2='句点' word='。' orgWord='。'/>
  </chunk>
 </sentence>
:
```

The mcaseframe.rb command can be used to extract dependency relationship from this XML.

2.2.2 Examples

Example 1: Basic Example

File `test.txt` is contained in `text` directory for parsing. The resuls is saved in `xml` directory.

```
$ more text/test.txt
子どもはリンゴがすきです。
望遠鏡で泳ぐ少女を見た。
$ mknp.rb I=text O=xml
#MSG# KNP: reading text/test.txt
#MSG# KNP: MP-2 aid=test.txt sid=0 (sentences:1/2, articles:1/1)
#MSG# KNP: MP-2 aid=test.txt sid=1 (sentences:2/2, articles:1/1)
#MSG# KNP2XML 1/1
```

```
#MSG# Elapse: 0.149sec, # of sentences=2, # of articles=1
#MSG#   0.075sec/sentence, 0.149sec/article
#MSG#   mpCount=2, poolSize=1000
#MSG#   maxLen=512Byte, maxSec=30sec, sizeLimit=2000MB
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mknp.rb I=text O=xml
$ head -n20 xml/test.txt
<?xml version='1.0' encoding='UTF-8'?>
<article id='test.txt'>
  <sentence id='0' text=' 子どもはリンゴがすきです。'>
    <chunk id='0' link='2' phraseType=' 格助詞句' caseType=' ガ２格' phrase=' 子供' phraseTok='
      <token id='0' class1=' 名詞' class2=' 普通名詞' word=' 子ども' orgWord=' 子ども' daiWord='
      <token id='1' class1=' 助詞' class2=' 副助詞' word=' は' orgWord=' は'/>
    </chunk>
    <chunk id='1' link='2' phraseType=' 格助詞句' caseType=' ガ格' phrase=' 林檎' phraseTok=' リ
      <token id='2' class1=' 名詞' class2=' 普通名詞' word=' リンゴ' orgWord=' リンゴ' daiWord='
      <token id='3' class1=' 助詞' class2=' 格助詞' word=' が' orgWord=' が'/>
    </chunk>
    <chunk id='2' link='-1' phraseType=' 用言句' phraseTok=' すきだ' rawPhrase=' すきです。' ph
      <token id='4' class1=' 形容詞' class3=' ナ形容詞' class4=' デス列基本形' word=' すきだ' or
      <token id='5' class1=' 特殊' class2=' 句点' word='。' orgWord='。'/>
    </chunk>
  </sentence>
  <sentence id='1' text=' 望遠鏡で泳ぐ少女を見た。'>
    <chunk id='0' link='3' phraseType=' 格助詞句' caseType=' デ格' phrase=' 望遠鏡' phraseTok='
      <token id='0' class1=' 名詞' class2=' 普通名詞' word=' 望遠' orgWord=' 望遠' daiWord=' 望遠
      <token id='1' class1=' 名詞' class2=' 普通名詞' word=' 鏡' orgWord=' 鏡' daiWord=' 鏡' cate
```

Example 2: Example of output results of KNP (original).

Resuls of KNP (original) is saved in `knp` directory.

```
$ more text/test.txt
子どもはリンゴがすきです。
望遠鏡で泳ぐ少女を見た。
$ mknp.rb I=text O=xml P=knp
#MSG# KNP: reading text/test.txt
#MSG# KNP: MP-2 aid=test.txt sid=0 (sentences:1/2, articles:1/1)
#MSG# KNP: MP-2 aid=test.txt sid=1 (sentences:2/2, articles:1/1)
#MSG# KNP2XML 1/1
#MSG# Elapse: 0.147sec, # of sentences=2, # of articles=1
#MSG#   0.074sec/sentence, 0.147sec/article
#MSG#   mpCount=2, poolSize=1000
#MSG#   maxLen=512Byte, maxSec=30sec, sizeLimit=2000MB
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mknp.rb I=text O=xml P=knp
$ head knp/test.txt
# S-ID:1 KNP:4.11-CF1.1 DATE:2014/07/28 SCORE:-21.86138
* 2D <文頭><SM-主体><SM-人><ハ><助詞><体言><係:未格><提題><区切:3-5><主題表現><格要素><連用
+ 2D <文頭><SM-主体><SM-人><ハ><助詞><体言><係:未格><提題><区切:3-5><主題表現><格要素><連用
子ども こども 子ども 名詞 6 普通名詞 1 * 0 * 0 "代表表記:子供/こども カテゴリ:人" <代表表記:
は は は 助詞 9 副助詞 2 * 0 * 0 NIL <かな漢字><ひらがな><付属>
* 2D <ガ><助詞><体言><係:ガ格><区切:0-0><格要素><連用要素><正規化代表表記:林檎/りんご><主辞
+ 2D <ガ><助詞><体言><係:ガ格><区切:0-0><格要素><連用要素><名詞項候補><先行詞候補><正規化代
リンゴ りんご リンゴ 名詞 6 普通名詞 1 * 0 * 0 "代表表記:林檎/りんご カテゴリ:植物
が が が 助詞 9 格助詞 1 * 0 * 0 NIL <かな漢字><ひらがな><付属>
* -1D <文末><句点><用言:形><レベル:C><区切:5-5><ID:（文末）><係:文末><提題受:30><主節><格要
```

2.3 mcaseframe.rb Extract Case Frame

This command extracts case frame from the output of KNP analysis.

Case frame refers to the clause consisting of verb and case particle in Japanese, such as 「リンゴ（が)」+「好き」、「望遠鏡（で)」+「見る」. This command reads the parsing results (in XML format) from `mknp.rb` command, extracts the case frame, and saves the output in CSV format.

2.3.1 Format

```
mcaseframe.rb I= o= [-key] [-mcmdenv] [--help]
```

`I=`	: Path name of the parsing results in XML file from mknp.rb.
`o=`	: Output of case frame file name.
`-key`	: Output in key type format.
`-mcmdenv`	: Display MCMD message containing environment variables. Default returns warning and error message (KG_VerboseLevel=2).
`--help`	: Display help

Extract case frame

The XML output of `mknp.rb` command is shown below (excerpt).

```
<sentence id='0' text='子どもはリンゴがすきです。'>
 <chunk id='0' link='2' phraseType='格助詞句' caseType='ガ２格' phrase='子供' phraseTok='子
  <token id='0' class1='名詞' class2='普通名詞' word='子ども' orgWord='子ども' daiWord='子供
  <token id='1' class1='助詞' class2='副助詞' word='は' orgWord='は'/>
 </chunk>
 <chunk id='1' link='2' phraseType='格助詞句' caseType='ガ格' phrase='林檎' phraseTok='リン
  <token id='2' class1='名詞' class2='普通名詞' word='リンゴ' orgWord='リンゴ' daiWord='林檎
  <token id='3' class1='助詞' class2='格助詞' word='が' orgWord='が'/>
 </chunk>
 <chunk id='2' link='-1' phraseType='用言句' phraseTok='すきだ' rawPhrase='すきです。' phrase
  <token id='4' class1='形容詞' class3='ナ形容詞' class4='デス列基本形' word='すきだ' orgWord
  <token id='5' class1='特殊' class2='句点' word='。' orgWord='。'/>
 </chunk>
</sentence>
```

In the examples, chunk id='0'「子どもは」 in link='2', chunk id='1'「リンゴが」also has link='2', therefore, chunk id='2'「すきです」is related . The dependency relationship is illustrated in the figure.

When using this command, the dependency relationships are extracted and saved as CSV shown as follows.

```
aid,sid,cid,contrastConj,denial,declinableWord,lid,caseWord,case
test.txt,0,2,,,すきだ,0,子ども,ガ２
test.txt,0,2,,,すきだ,1,リンゴ,ガ
```

The meaning of each item of CSV is shown as follows.

`aid`	: Name of input file
`sid`	: Line number (sentence ID)
`cid`	: Chunk ID
`contrastConj`	: Reverse connection conjunctions
`denial`	: Set as 1 when chunk contains negative word
`declinableWord`	: Verb clause
`lid`	: Chunk ID of case particle clause
`caseWord`	: Case particle clause
`case`	: Kind of case particle clause

2.3.2 Examples

Example 1: Basic example

Example used in the previous section. One line has become one case frame.

```
$ more xml/test.txt
<?xml version='1.0' encoding='UTF-8'?>
<article id='test.txt'>
  <sentence id='0' text='子どもはリンゴがすきです。'>
    <chunk id='0' link='2' phraseType='格助詞句' caseType='ガ２格' phrase='子供' phraseTok='
      <token id='0' class1='名詞' class2='普通名詞' word='子ども' orgWord='子ども' daiWord='
      <token id='1' class1='助詞' class2='副助詞' word='は' orgWord='は'/>
    </chunk>
    <chunk id='1' link='2' phraseType='格助詞句' caseType='ガ格' phrase='林檎' phraseTok='リ
      <token id='2' class1='名詞' class2='普通名詞' word='リンゴ' orgWord='リンゴ' daiWord='
      <token id='3' class1='助詞' class2='格助詞' word='が' orgWord='が'/>
    </chunk>
    <chunk id='2' link='-1' phraseType='用言句' phraseTok='すきだ' rawPhrase='すきです。' ph
      <token id='4' class1='形容詞' class3='ナ形容詞' class4='デス列基本形' word='すきだ' or
      <token id='5' class1='特殊' class2='句点' word='。' orgWord='。'/>
    </chunk>
  </sentence>
  <sentence id='1' text='望遠鏡で泳ぐ少女を見た。'>
    <chunk id='0' link='3' phraseType='格助詞句' caseType='デ格' phrase='望遠鏡' phraseTok='
      <token id='0' class1='名詞' class2='普通名詞' word='望遠' orgWord='望遠' daiWord='望遠
      <token id='1' class1='名詞' class2='普通名詞' word='鏡' orgWord='鏡' daiWord='鏡' cate
      <token id='2' class1='助詞' class2='格助詞' word='で' orgWord='で'/>
    </chunk>
    <chunk id='1' link='2' phraseType='用言句' phrase='泳ぐ' phraseTok='泳ぐ' rawPhrase='泳
      <token id='3' class1='動詞' class3='子音動詞ガ行' class4='基本形' word='泳ぐ' orgWord=
    </chunk>
    <chunk id='2' link='3' phraseType='格助詞句' caseType='ヲ格' phrase='少女' phraseTok='少
      <token id='4' class1='名詞' class2='普通名詞' word='少女' orgWord='少女' daiWord='少女
      <token id='5' class1='助詞' class2='格助詞' word='を' orgWord='を'/>
    </chunk>
    <chunk id='3' link='-1' phraseType='用言句' phraseTok='見る' rawPhrase='見た。' phrase='
      <token id='6' class1='動詞' class3='母音動詞' class4='タ形' word='見る' orgWord='見た'
      <token id='7' class1='特殊' class2='句点' word='。' orgWord='。'/>
    </chunk>
  </sentence>
</article>mcaseframe.rb I=xml o=caseframe.csv
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mcaseframe.rb I=xml o=caseframe.csv
more caseframe.csv
aid,sid,cid,contrastConj,denial,declinableWord,lid,caseWord,case
test.txt,0,2,,,すきだ,0,子ども,ガ２
test.txt,0,2,,,すきだ,1,リンゴ,ガ
test.txt,1,3,,,見る,0,望遠鏡,デ
test.txt,1,3,,,見る,2,少女,ヲ
```

Example 2: Output of key type format

When executing by adding the option `-key`, case particle influencing inflectable word from the line is saved in output.

```
$ mcaseframe.rb -key I=xml o=caseframe2.csv
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mcaseframe.rb -key I=xml o=caseframe2.c
$ more caseframe2.csv
aid,sid,cid,contrastConj,denial,lid,word,type
test.txt,0,2,,,2,すきだ,用言
test.txt,0,2,,,0,子ども,ガ２
test.txt,0,2,,,1,リンゴ,ガ
test.txt,1,1,,,1,泳ぐ,用言
test.txt,1,3,,,3,見る,用言
test.txt,1,3,,,0,望遠鏡,デ
test.txt,1,3,,,2,少女,ヲ
```

2.4 mnewdic.rb Output adjacent word pair candidate from corpus

From text corpus (large set of text files), returns the candidate information of adjacent word pair to be registered in the dictionary.

JUMAN contains a standard dictionary with common words, for text mining in specific fields, interpretation of 1 phrase may be divided into multiple words.

`mnewdic.rb` command analyze the given corpus based on the frequency of phrases with words pairs that appears at the same time, and output the list in CSV file.

`mjumandic.rb` command simply dictionary registration in the process of text mining.

2.4.1 Format

```
mnewdic.rb [i=] [O=] [S=] [n=] [seed=] [-dai] [-mcmdenv] [--help]
```

`i=`	: Corpus file name
`O=`	: Output directory name
`S=`	: Minimum value of word pair appearance
`n=`	: Output sentence number for word pair
`seed=`	: Seed of random number
`-dai`	: Use a symbol to represent headwords
`-mcmdenv`	: Display MCMD message containing environment variables. Default returns warning and error message (KG_VerboseLevel=2).
`--help`	: Display help

Example of input file

Given the corpus file specified at `i=` parameter, one row corresponds to one sentence in text file.

```
3年ぶりにウォークマンを買ったけど、育休中はあまり活躍の余地がないですね。
待機児童解消の方がいい気がするけど。
:
```

Example of Output File

In the directory specified at `O=` parameter, the results are saved in `words.csv` file and `corpus.csv` file （when `nkf` command is installed, the character code converted from Shift JIS is also saved in the output in both files.

```
見出し語, 品詞, 読み, カテゴリ, ドメイン,pid,word1,word2,freq
職場復帰,,,,,0, 職場, 復帰,31
授業参観,,,,,1, 授業, 参観,28
会議参加,,,,,2, 会議, 参加,26
:
```

The description of each item in `words.csv` file is shown below.

見出し語	: 見出し語
品詞	: 品詞
読み	: 読み
カテゴリ	: カテゴリ
ドメイン	: ドメイン
`pid`	: pid
`word1`	: 語 1
`word2`	: 語 2
`freq`	: 出現頻度

Users can refer to `corpus.csv` to check whether the word of registered candidates appeared in the text.

```
pid,id,text
0,52,"神戸で始めての、育休後職場復帰セミナーを開催しました。"
0,317,"僕の知り合いは 2 人子どもを産んで、立て続けに産休＋育休を取って、職場復帰した。"
:
```

2.4.2 Examples

Example 1: Basic Example

```
$ head tweets.txt
3 年ぶりにウォークマンを買ったけど、育休中はあまり活躍の余地がないですね。
待機児童解消の方がいい気がするけど。
読売テレビ（日本テレビ系列）ウェークアップ！ぷらすに蓮舫ネクスト規制改革担当大臣が生出演！
この学生さんは、仕事に不利じゃなかったら、3 年育休取れるのも良いな、って思ってるよね。
今の人事制度のまま育休 3 年とか、前以上に女性が締め出されるだけでは。
女子大生でも分かる、3 年間の育児休暇が最悪な結果をもたらす理由。(中嶋よしふみ)
保育園を中心に期間とか決めるの、おかしいよな\UTF{FF5E}。
女性が必ず子育てしなきゃならない社会なら結婚絶対したくない…
育休とかの前に、母親に育児に専念させるなら女性の雇用よりもまず、男性の雇用、給料なんだよね。
安倍総理きた! 育児休暇三年は…女としては嬉しいけど、会社に申し訳ないよねえ
$ mnewdic.rb i=tweets.txt O=newdic
#MSG# start to parse each line...
#MSG# working at line 0
#MSG# working at line 100
#MSG# working at line 200
#MSG# working at line 300
#MSG# working at line 400
#MSG# working at line 500
#MSG# working at line 600
#MSG# working at line 700
#MSG# working at line 800
#MSG# working at line 900
#MSG# working at line 1000
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mnewdic.rb i=tweets.txt O=newdic
$ ls newdic
corpus.csv
corpus_sjis.csv
words.csv
words_sjis.csv
$ head newdic/words.csv
見出し語, 品詞, 読み, カテゴリ, ドメイン,pid,word1,word2,freq
職場復帰,,,,,0, 職場, 復帰,31
授業参観,,,,,1, 授業, 参観,28
会議参加,,,,,2, 会議, 参加,26
育休延長,,,,,3, 育休, 延長,19
子育て支援,,,,,4, 子育て, 支援,18
育児休暇３,,,,,5, 育児休暇, ３,18
待機児童ゼロ,,,,,6, 待機児童, ゼロ,17
規制緩和,,,,,7, 規制, 緩和,16
給付金,,,,,8, 給付, 金,15
```

2.5 mjumandic.rb Conversion from CSV to JUMAN Dictionary

Convert CSV dictionary data to JUMAN dictionary format.

The output from `mnewdic.rb` command is also in CSV format.

2.5.1 Format

```
mjumandic.rb [i=] [O=] [exe=] [-mcmdenv] [--help]
```

`i=`	: Dictionary file name of CSV file.
`O=`	: Directory name that contains the dictionary of JUMAN.
`exe=`	: Command path such as makein (Default: /usr/local/bin) This is not required when JUMAN is installed with default setting.
`-mcmdenv`	: Display MCMD message containing environment variables. Default returns warning and error message (KG_VerboseLevel=2).
`--help`	: Display help

Example of input file

An example of dictionary file defined at `i=` parameter is shown below. The 5 items includes `headword`, `read`, `part of speech`, `category`, `domain`.

```
id, 見出し語, 読み, 品詞, カテゴリ, ドメイン
1, 連結営業利益, れんけつえいぎょうりえき, 普通名詞, 抽象物, ビジネス
2, 米国債, べいこくさい,, 抽象物, ビジネス
3, 上方修正, じょうほうしゅうせい, サ変名詞, 抽象物, ビジネス
4, 日本航空, にほんこうくう, 組織名,,
5, 夏目漱石, なつめそうせき, 人名, 日本, 姓
6, 安倍首相 安倍晋太郎 安倍晋太郎首相, あべしゅしょう, 人名, 日本, 姓名
```

The meaning of each item is as follows.

見出し語 見出し語には、表記ゆれなどの複数の見出し語を半角空白で区切って列挙できる。見出し語がないとエラーとなる。

品詞 品詞は名詞のみ対応しており、以下に示す名詞の下位の品詞を「品詞」項目に登録する。
普通名詞, サ変名詞, 時相名詞, 数詞, 副詞的名詞, 固有名詞, 人名, 組織名, 地名
品詞が省略されると、「普通名詞」が指定されたものとする。品詞の体系は以下の URL を参照のこと。http://www.unixuser.org/~euske/doc/postag/

読み 読みがないとエラーとなる。

カテゴリ カテゴリは以下の 22 種（省略可能）
人, 組織・団体, 動物, 植物, 動物-部位, 植物-部位, 人工物-食べ物, 人工物-衣類, 人工物-乗り物
人工物-金銭, 人工物-その他, 自然物, 場所-施設, 場所-施設部位, 場所-自然, 場所-機能
場所-その他, 抽象物, 形・模様, 色, 数量, 時間

ドメイン ドメインは以下の 12 種（省略可能）
文化・芸術, 交通, レクリエーション, 教育・学習, スポーツ, 科学・技術, 健康・医学
ビジネス, 家庭・暮らし, メディア, 料理・食事, 政治

Category and domain is a valid item to common noun and word formed by adding suru to noun.

Please refer to the following URL for references on category and domain.

http://nlp.ist.i.kyoto-u.ac.jp/DLcounter/lime.cgi?down=http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/knp/20090930-juman-knp.ppt&name=20090930-juman-knp.ppt

2.5.2 Examples

Example 1: Basic example

```
$ more dic1.csv
id, 見出し語, 読み, 品詞, カテゴリ, ドメイン
1, 連結営業利益, れんけつえいぎょうりえき, 普通名詞, 抽象物, ビジネス
2, 米国債, べいこくさい, , 抽象物, ビジネス
3, 上方修正, じょうほうしゅうせい, サ変名詞, 抽象物, ビジネス
4, 日本航空, にほんこうくう, 組織名, ,
5, 夏目漱石, なつめそうせき, 人名, 日本, 姓
6, 安倍首相 安倍晋太郎 安倍晋太郎首相, あべしゅしょう, 人名, 日本, 姓名
7, ２ちゃんねる にちゃんねる, にちゃんねる, , ,
$ mjumandic.rb i=dic1.csv O=jumandic
#END# kgcut f=品詞, 見出し語, 読み i=dic1.csv
#END# kgdelnull f=見出し語, 読み
#END# kgsortf f=見出し語
#END# kguniq k=見出し語 o=/tmp/__MTEMP_68157_70357348549040_0
#END# mcsvin i=/tmp/__MTEMP_68157_70357348549040_0
Mon Jul 28 01:38:37 2014
/usr/local/share/juman/dic/JUMAN.grammar parsing... done.

Mon Jul 28 01:38:37 2014
/usr/local/share/juman/dic/JUMAN.katuyou parsing... done.

Mon Jul 28 01:38:37 2014
/usr/local/share/juman/dic/jumandic.tab parsing... done.

jumandic.dic parsing... done.

execution time:    0.000s
processor time:    0.000s
File Name "/Users/maegawa/git/nysol/nysol/doc/fumi/jp/examples/jumandic/jumandic.dat"

## 10 entry  814 th char
Saving pat-tree "/Users/maegawa/git/nysol/nysol/doc/fumi/jp/examples/jumandic/jumandic.pat"
QUIT
#MSG# jumandic 内の jumandic.dat,jumandic.pat の２つのファイルがユーザ辞書として必要となる。
#MSG# ~/.jumanrc ファイルを編集し、これらのファイルが格納されたパス名を以下のように追加登録す
#MSG# (辞書ファイル
#MSG#         /usr/local/share/juman/dic
#MSG#         /usr/local/share/juman/autodic
#MSG#         /usr/local/share/juman/wikipediadic
#MSG#         /Users/maegawa/git/nysol/nysol/doc/fumi/jp/examples/jumandic
#MSG# )
#END# /Users/maegawa/.rvm/rubies/ruby-2.0.0-p247/bin/mjumandic.rb i=dic1.csv O=jumandic
$ ls jumandic
jumandic.dat
jumandic.dic
jumandic.int
jumandic.pat
$ more jumandic/jumandic.dic
(名詞 (サ変名詞 ((読み じょうほうしゅうせい) (見出し語 上方修正) (意味情報 "代表表記:上方修
(名詞 (人名 ((読み なつめそうせき) (見出し語 夏目漱石) (意味情報 ""))))
(名詞 (人名 ((読み あべしゅしょう) (見出し語 安倍首相 安倍晋太郎 安倍晋太郎首相) (意味情報 "
(名詞 (組織名 ((読み にほんこうくう) (見出し語 日本航空) (意味情報 "代表表記:日本航空/にほん
(名詞 (普通名詞 ((読み べいこくさい) (見出し語 米国債) (意味情報 "代表表記:米国債/べいこくさ
(名詞 (普通名詞 ((読み れんけつえいぎょうりえき) (見出し語 連結営業利益) (意味情報 "代表表記
(名詞 (普通名詞 ((読み にちゃんねる) (見出し語 ２ちゃんねる にちゃんねる) (意味情報 "代表表
```