WN-Toolkit

Introduction

The WN-Toolkit is a set of programs for the creation of WordNets following the expand model. The toolkit is distributed under a free software license. If you use this toolkit in your work, please cite:

Oliver, A. (2014) WN-Toolkit: Automatic generation of WordNets following the expand model Proceedings of the 7th International Global WordNet Conference. January 25-29, 2014. Tartu. Estonia. pp. 7-15. ISBN: 978-9949-32-492-7

Pre-requisites

WN-Toolkit is written in Python (version 2.7.x). To run the programs you need a Python interpreter installed on your system. Typical Linux and Mac distributions include a Python interpreter so you don't need to install anything on your system. Windows users can freely download a Python interpreter from http://www.python.org/download/ (download and install a 2.7.x version).

Components

The WN-Toolkit includes programs for WordNet creation as well as some resources.

Programs

In this section the programs of the Toolkit are presented. Some basic instructions for their use are also provided.

Miscelaneous tools

In the **0_pwn30-monosemic** directory of the distribution we can find a program called **createmonosemicwordlist.py**. This program extracts the list of monosemic variants (that is, the variants associated to only one synset) from the data.adj, data.adv, data.noun, data.verb, index.adj, index.adv, index.noun and index.verb files of the Princeton WordNet. The program creates 3 files: one for all monosemic variants, one for those written in lower-case (-min.txt) and one for those written with the first letter in upper-case (-maj.txt):

The program has the **-h** option to show the arguments:

```
python createmonosemicwordlist.py -h
```

```
usage: createmonosemicwordlist.py [-h] [-v] [-d DIRECTORY] [-p STRING]

Monosemic variants extraction

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -d DIRECTORY, --directory DIRECTORY
                        The directory where the data.adj, data.adv, data.noun,
                        data.verb, index.adj, index.adv, index.noun and
                        index.verb files of PWN are located. Default: current
                        directory
  -p STRING, --prefix STRING
                        The prefix for the name of the output files:
                        prefix.txt, prefix-maj.txr, prefix-min.txt. Default:
                        pwn
```

In the same directory we can find the three files already created for PWN 3.0:

- pwn30-monosemic.txt
- pwn30-monosemic-min.txt
- pwn30-monosemic-maj.txt

Dictionary based strategy

The main program for this strategy is **wndictionary.py**. If we use the **-h** option the arguments will be shown:

```
python wndictionary.py -h

usage: wndictionary.py [-h] [-v] -d FILE -i FILE -o FILE

Translates PWN variants with bilingual dictionaries.

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -d FILE, --dictionary FILE
                        The dictionary to be used.
  -i FILE, --input FILE
                        The input file containig the variants to be translated
                        (createmonosemicwordlistprogram.py can be used to
                        generate the input files)
  -o FILE, --output FILE
                        The output file containig the translated variants
```

For the use of this program we need a dictionary with English word, POS, and target language word separated with tabulators. There is an example:

```
climate change	n	canvi climàtic
```

We also need the a file with a list of variants to be translated (and created with the createmonosemicwordlist.py) for example:

```
13449450-n	climate_change
```

Then we will get a new variant for this synset:

```
13449450-n	canvi_climàtic
```

We offer a variation of **wndictionary.py** called **wndictionary-normalizecaps.py** that implements a simple strategy to normalize the capitalization of the entries. This is necessary for some dictionaries where all the entries start with upper case letters. This strategy only works for languages with similar capitalization rules as English (only proper names are written with upper cap letters).

This program needs an additional argument:

```
  -n FILE, --normalize FILE
                        The English wn-data-eng.tab file from OMW
```

OWN: Open Multilingua WordNet http://www.casta-net.jp/~kuribayashi/multi/
Along with **wndictionary.py** we distribute some programs for the creation of bilingual dictionaries with the required format from some freely available dictionaries:

- wiktionary2bildic.py: creation of bilingual dictionaries from Wiktionary XML dumps http://dumps.wikimedia.org/enwiktionary/
- wikipedia2bildic.py: creation of bilingual dictionaries from Wikipedia XML dumps http://dumps.wikimedia.org/enwiki/
- apertium2bildic.py: creation of bilingual dictionaries from Apertium http://apertium.org
- dacco2bildic.py: creation of bilingual dictionaries from the Dacco Catalana-English dictionary http://www.catalandictionary.org/
- TO2bildic.py: creation of bilingual dictionaries from the terminology databases from Termcat's Terminologia Obertahttp://www.termcat.cat/productes/toberta.htm

All these programs accepts the **-h** argument to get the help.

A program for combining several dictionaries is also provided (**combinedictionary.py**). The program takes a list of files to combine and the output file as the last element of the list. To avoid deleting an existing dictionary the program checks if the output dictionary exists. If so, the program stops.

```
python combinedictionary.py dict1.txt dict2.txt ... dictn.txt outputdictionary.txt
```

Babelnet based strategy

Babelnet http://babelnet.org/ is a multilingual lexicalized semantic network and ontology. BabelNet was automatically created by linking the largest multilingual Web encyclopedia (Wikipedia) to the most popular computational lexicon of the English language, WordNet. (source: Wikipedia (http://en.wikipedia.org/wiki/BabelNet)).

As Babelnet relates WordNet synsets with Wikipedia entries, and Wikipedia entries are multilingual through the interlingual link, this resource can be directly used for the creation of WordNets for several languages. The programs we provide simply modifies the format of the files, and all the interesting things are done by the Babelnet project.

BabelNet 2.0 version was recently released and we offer programs for dealing with both versions (1.1.1. and 2.0).

For the 1.1.1 version of Babelnet we can use the **babel2wordnet.py** program. This version can be downloaded fromhttp://babelnet.org/data/babelnet-1.1.1-glosses.bz2 and you can download a copy from the resources section of WN-Toolkit. You can use the **-h** option to get the help:

```
python babel2wordnet.py -h

usage: babel2wordnet.py [-h] [-v] -l LANG -o FILE [-d FILE] [-w FILE]
                        BABEL_GlOSSES

Creates a WordNet from BabelNet

positional arguments:
  BABEL_GlOSSES         The path to the babel-glosses file

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -l LANG, --lang LANG  the target language: es, fr, ca...
  -o FILE, --output FILE
                        a file to write the results
  -d FILE, --diccionari FILE
                        a file containing a dictionary created from Wikipedia.
  -w FILE, --wordnet FILE
                        The directory where the PWN data.noun file of PWN is
                        located. Default: current directory. Useful for caps
                        normalization.
```

The **-d** and **-w** arguments are optional.

The WN-Toolkit also provides a program for using the Babelnet 2.0 version, that can be downloaded fromhttp://babelnet.org/data/babelnet-2.0.0-core-dump.bz2. The program needs two parameters: the path and name of the output file and the language code.\The babelnet-2.0.0-core-dump file must be in the same directory as this program.

Parallel corpora based strategies

The main program for this strategy is **synset-word-alignment.py**. If you use the **-h** option you'll get the help of the program:

```
python synset-word-alignment.py -h

usage: synset-word-alignment.py [-h] [-v] -s FILE -t FILE -o FILE [-i VALUE]
```

```
                                 [-f VALUE]

Synset - variant alignment algorithm

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -s FILE, --sc FILE    The sense-tagged corpus
  -t FILE, --tc FILE    The target corpus, tagged with simple tags
  -o FILE, --output FILE
                        The output file
  -i VALUE, --index VALUE
                        The minimun index (freq of 1st candidate divided by
                        freq of 2n candidate). Recommended higher than 1.
                        Default: 2.5
  -f VALUE, --fr VALUE  The maximun number of times the synset frequency can
                        be higher than the variant frequency. Default: 5
```

This program requires an English sense-tagged corpora with the following format:

```
02186338-a 15123115-n , the 09917593-n had 01825237-v to 01974062-v onto 00007846-n 's 02374451-n .

They 01825237-v to 02128873-v what his 05558717-n 02730471-v like - the 00007846-n 's .

He 02133435-v 00146594-r 00476819-a 02604760-v 02310895-a .
```

It also needs a target language corpus with simple tags, aligned with the English sense-tagged one, as the following:

```
un|c dia|n ,|c el|c nen|n haver|c voler|v pujar|v a_cavall|r general|a burnside|n .|c
voler|v veure|v el|c que|c el|c seu|c esquena|n es|c sentir|v com|c -|c el|c de|c el|c general|n .|c
es|c veure|v tan|r còmode|a ser|v recte|a .|c
```

Then the program uses a simple word-alignment strategy to get the target language variants, as:

```
00048475-r  ara
05833840-n  idea
05311054-n  ull
```

As corpora in these formats are not easily available, we provide some programs to help in the creation of the corpora:

- •freeling-simpletag.py: Tags a text with simplified tagset using Freeling (eng, cat, spa, ita, por, rus, glg). The query is done using a Freeling server that can run in your computer or remotely.
- •treetagger-simpletag.py: Tags a text with simplified tagset using Tree Tagger (fre, deu, rus). To add more languages, simply edit this program.) You need to have TreeTagger installed on your system.

All these programs have the **-h** option to get the help.

We also provide the **ukbtosenses.py** program to transform the Freeling+UKB tagged corpus into a suitable format for the **synset-word-alignment.py**: 2 arguments must been provided: the path and name of the input file and the path and name of the output file.

In the **Resources** section we provide several corpus pre-processed for the use of this strategy. They are fully described in that section.

Evaluation tools

The toolkit also provides programs to allow the automatic evaluation of the results comparing them with a reference WordNet for the target languages. This reference WordNet must be in the Open Multilingual WordNet format. Several freely available WordNets in this format can be downloaded from: http://www.casta-net.jp/~kuribayashi/multi/. This automatic evaluation usually offers lower values than manual evaluation (see the paper for details).
The **wn-evaluate.py** program can perform evaluation of the results. If we call the **-h** option we get:

```
python wn-evaluate.py -h

usage: wn-evaluate.py [-h] [-v] -r FILE -e FILE

Evaluation algorithm

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -r FILE, --ref FILE   The reference wordnet for evaluation in Open
                        Multilingual WordNet format
  -e FILE, --eva FILE   The target WordNet to evaluate
```

The evaluation provides the following figures:

```
TOTAL      : 4153
EVALUATED  : 2939
PRECISION  : 79.58
PRECISION N: 79.95
PRECISION V: 75.16
PRECISION A: 78.73
PRECISION R: 63.23
```

The Toolkit also offers a variation of this program using the same arguments: the **wn-evaluate-info.py**. This program needs the wordnet30-eng.db (distributed with the WN-Toolkit) and this file should be located in the same directory. This program creates two output files:

- incorrect.txt: containing information about the variants evaluated as incorrect.
- nonevaluated.txt: containing information about the non-evaluated variants.

This files can help in the manual evaluation of the results and offer the following information:

The incorrect.txt offers:

```
00176150-a  favorable   propici auspicious  auguring favorable circumstances and good luck
```

That is: synset, evaluated target-language variant, reference target language variants, English variants, English definition

The output of the nonevaluated.txt offers similar information, but lacking the reference target language variants.

```
01981916-a  al·legòric  allegorical,allegoric   used in or characteristic of or containing allegory
```

The program also provides some figures:

```
TOTAL       : 4153
PRECISION   : 79.58
-------
NEW VARIANTS: 1213
```

Resources

Several languages resources have been adapted and released.

Dictionaries

Apertium dictionaries

A set of dictionaries created from the transfer dictionaries of the **Apertium** Machine Translation System:

```
├── dict-apertium-en-ca.txt
├── dict-apertium-en-cy.txt
├── dict-apertium-en-eo.txt
├── dict-apertium-en-es.txt
├── dict-apertium-en-eu.txt
├── dict-apertium-en-gl.txt
├── dict-apertium-en-ht.txt
├── dict-apertium-en-is.txt
└── dict-apertium-en-mk.txt
```

Wiktionary dictionaries

We also offer several dictionaries created from the **Wiktionary** (2012-08-05 dump):

```
├── wiktionary-en-ar.txt
├── wiktionary-en-bg.txt
├── wiktionary-en-ca.txt
├── wiktionary-en-cs.txt
├── wiktionary-en-da.txt
├── wiktionary-en-de.txt
├── wiktionary-en-el.txt
├── wiktionary-en-es.txt
├── wiktionary-en-et.txt
├── wiktionary-en-eu.txt
├── wiktionary-en-fi.txt
├── wiktionary-en-fr.txt
├── wiktionary-en-gl.txt
├── wiktionary-en-hi.txt
├── wiktionary-en-hr.txt
├── wiktionary-en-hu.txt
├── wiktionary-en-it.txt
├── wiktionary-en-kn.txt
├── wiktionary-en-lt.txt
├── wiktionary-en-lv.txt
├── wiktionary-en-ml.txt
├── wiktionary-en-mt.txt
├── wiktionary-en-nl.txt
├── wiktionary-en-pl.txt
├── wiktionary-en-pt.txt
├── wiktionary-en-ro.txt
├── wiktionary-en-ru.txt
├── wiktionary-en-sk.txt
├── wiktionary-en-sl.txt
├── wiktionary-en-sv.txt
├── wiktionary-en-ta.txt
├── wiktionary-en-te.txt
└── wiktionary-en-zh.txt
```

Wikipedia dictionaries

We also offer several dictionaries created from the Wikipedia (2012-08-05 dump):

```
 ├─── wikipedia-en-ar.txt
├── wikipedia-en-bg.txt
├── wikipedia-en-ca.txt
├── wikipedia-en-cs.txt
├── wikipedia-en-da.txt
├── wikipedia-en-de.txt
├── wikipedia-en-el.txt
├── wikipedia-en-es.txt
├── wikipedia-en-et.txt
├── wikipedia-en-eu.txt
├── wikipedia-en-fi.txt
├── wikipedia-en-fr.txt
├── wikipedia-en-gl.txt
├── wikipedia-en-hr.txt
├── wikipedia-en-hu.txt
├── wikipedia-en-it.txt
├── wikipedia-en-lt.txt
├── wikipedia-en-lv.txt
├── wikipedia-en-mt.txt
├── wikipedia-en-nl.txt
├── wikipedia-en-pl.txt
├── wikipedia-en-pt.txt
├── wikipedia-en-ro.txt
├── wikipedia-en-ru.txt
├── wikipedia-en-sk.txt
├── wikipedia-en-sl.txt
├── wikipedia-en-sv.txt
└── wikipedia-en-zh.txt
```

All the dictionaries are in the required format for the use with the WN-Toolkit, that is: English word tab POS tab Target Language Word:

```
accidentally    r    accidentalment
acclaim v    aclamar
acclamation n    aclamació
acclimate    v    aclimatar
```

If your language is not listed, you can create the dictionaries with the programs distributed with the WN-Toolkit.

Parallel corpora

Semcor 3.0

We offer some preprocessed files, ready for the use with WN-Toolkit:

semcor30-eng.txt: The English Semcor corpus, one sentence by line:

```
One day, the children had wanted to get up onto General Burnside's horse.
They wanted to see what his back felt like - the General's.
He looked so comfortable being straight.
```

semcor30-senses-eng.txt: the English Semcor where sense-tagged word have been replaced by their PWN synset.

```
02186338-a 15123115-n , the 09917593-n had 01825237-v to 01974062-v onto 00007846-n 's 02374451-n .
They 01825237-v to 02128873-v what his 05558717-n 02730471-v like - the 00007846-n 's .
He 02133435-v 00146594-r 00476819-a 02604760-v 02310895-a .
```

Translations for several languages performed using Google Translate:

```
Un dia, els nens havia volgut pujar a cavall general Burnside.
Volien veure el que la seva esquena es sentia com - la del General.
Es veia tan còmode ser recta.
```

Translations for the following languages are distributed:

```
├── semcor30-bul.txt
├── semcor30-cat.txt
├── semcor30-deu.txt
├── semcor30-eng.txt
├── semcor30-fra.txt
├── semcor30-hrv.txt
├── semcor30-ita.txt
├── semcor30-lat.txt
├── semcor30-por.txt
├── semcor30-rus.txt
├── semcor30-spa.txt
```

Some of these translation have been tagged with simple tags. These are the files required for the WN-Toolkit:

```
un|c dia|n ,|c el|c nen|n haver|c voler|v pujar|v a_cavall|r general|a burnside
|n .|c
voler|v veure|v el|c que|c el|c seu|c esquena|n es|c sentir|v com|c -|c el|c de
|c el|c general|n .|c
es|c veure|v tan|r còmode|a ser|v recte|a .|c
```

```
├── semcor30-tagged-cat.txt
├── semcor30-tagged-deu.txt
├── semcor30-tagged-eng.txt
├── semcor30-tagged-fra.txt
├── semcor30-tagged-ita.txt
├── semcor30-tagged-por.txt
├── semcor30-tagged-rus.txt
└── semcor30-tagged-spa.txt
```

Princeton WordNet Gloss Corpus

We offer some preprocessed files, ready for the use with WN-Toolkit:

pwgc30-eng.txt: The English Princeton WordNet Gloss Corpus, one sentence by line:

```
(usually followed by to) having the necessary means or skill or know-how or aut
hority to do something; able to swim; she was able to program her computer; we
were at last able to buy a car; able to get a grant for the project;
(usually followed by to) not having the necessary means or skill or know-how; u
nable to get to town without a car; unable to obtain funds;
```

pwgc30-senses-eng.txt: the English Princeton WordNet Gloss Corpus where sense-tagged word have been replaced by their PWN synset.

```
( usually followed by to ) having the 01580050-a 00172710-n or skill or 05616786-n or 05196582-n to do something ; 00001740- to swim ; she was 00001740- to program her computer ; we were at last 00001740- to buy a car ; 00001740- to get a grant for the project ;
( usually followed by to ) 00024073-r having the 01580050-a 00172710-n or skill or 05616786-n ; 00002098- to get to town without a car ; 00002098- to obtain funds ;
```

Translations for several languages performed using Google Translate:

```
(En general seguit per a) tenir els mitjans necessaris o habilitat o coneixement o autoritat per fer alguna cosa, saber nedar, ella era capaç de programar l'ordinador, estàvem per fi capaç de comprar un cotxe, capaç d'obtenir una beca per el projecte;
(En general seguit per a) que no tenen els mitjans necessaris o habilitat o coneixement, incapaç d'arribar a la ciutat sense un cotxe, no poden obtenir fons;
```

Translations for the following languages are distributed:

```
├─── pwgc30-cat.txt
├── pwgc30-deu.txt
├── pwgc30-eng.txt
├── pwgc30-fra.txt
├── pwgc30-ita.txt
├── pwgc30-por.txt
├── pwgc30-rus.txt
```

Some of these translation have been tagged with simple tags. These are the files required for the WN-Toolkit:

```
(|c en_general|r seguit|n per|c a|c )|c tenir|v el|c mitjà|n necessari|a o|c habilitat|n o|c coneixement|n o|c autoritat|n per|c fer|v algun|c cosa|n ,|c saber|v nedar|v ,|c ell|c ser|v capaç|a de|c programar|v el|c ordinador|n ,|c estar|v per|c fi|n capaç|a de|c comprar|v un|c cotxe|n ,|c capaç|a de|c obtenir|v un|c beca|n per|c el|c projecte|n ;|c
```

```
├─── pwgc30-senses-eng.txt
├── pwgc30-tagged-cat.txt
├── pwgc30-tagged-deu.txt
├── pwgc30-tagged-eng.txt
├── pwgc30-tagged-fra.txt
├── pwgc30-tagged-ita.txt
├── pwgc30-tagged-por.txt
└── pwgc30-tagged-spa.txt
```

DGT_TM-release2013

We offer a pre-processed version of this corpus for several languages:

```
├── DGT-TM-preprocess-en-bg.txt.gz
├── DGT-TM-preprocess-en-cs.txt.gz
├── DGT-TM-preprocess-en-da.txt.gz
├── DGT-TM-preprocess-en-de.txt.gz
├── DGT-TM-preprocess-en-el.txt.gz
├── DGT-TM-preprocess-en-es.txt.gz
├── DGT-TM-preprocess-en-et.txt.gz
├── DGT-TM-preprocess-en-fi.txt.gz
├── DGT-TM-preprocess-en-fr.txt.gz
├── DGT-TM-preprocess-en-hu.txt.gz
├── DGT-TM-preprocess-en-it.txt.gz
├── DGT-TM-preprocess-en-lt.txt.gz
├── DGT-TM-preprocess-en-lv.txt.gz
├── DGT-TM-preprocess-en-mt.txt.gz
├── DGT-TM-preprocess-en-nl.txt.gz
├── DGT-TM-preprocess-en-pl.txt.gz
├── DGT-TM-preprocess-en-pt.txt.gz
├── DGT-TM-preprocess-en-ro.txt.gz
├── DGT-TM-preprocess-en-sk.txt.gz
├── DGT-TM-preprocess-en-sl.txt.gz
├── DGT-TM-preprocess-en-sv.txt.gz
```

These files have several fields separated by tabulators:

- The English text.
- The English text where sense-tagged words are replaced by their PWN synset. The word sense disambiguation and Tagging has been made using Freeling + UKB.
- The corresponding target language text.

Here we can see an example:

```
Corrigendum to Commission Regulation (EU) No 177/2010 of 2 March 2010 amending
Regulation (EEC) No 2454/93 laying down provisions for the implementation of Co
uncil Regulation (EEC) No 2913/92 establishing the Community Customs Code
06769578-n to Commission_Regulation ( 08173515-n ) 07205104-n 177/2010 of ??] 0
0205885-v 06664051-n ( 08173515-n ) 07205104-n 2454/93 01494310-v 00096089-r 01
057200-n for the 00044150-n of Council_Regulation ( 08173515-n ) 07205104-n 291
3/92 01647229-v the Community_Customs_Code  Corrección de errores del Regla
mento (UE) no 177/2010 de la Comisión, de 2 de marzo de 2010, que modifica el R
eglamento (CEE) no 2454/93, por el que se fijan determinadas disposiciones de a
plicación del Reglamento (CEE) no 2913/92 del Consejo, por el que se establece
el código aduanero comunitario
```

Some of these corpora have been further preprocessed and the tagged version (using simple tags) of the target language text have been added. This has been done for the following languages:

```
├── DGT-TM-preprocess-simpletagged-en-de.txt.gz
├── DGT-TM-preprocess-simpletagged-en-es.txt.gz
├── DGT-TM-preprocess-simpletagged-en-fr.txt.gz
├── DGT-TM-preprocess-simpletagged-en-it.txt.gz
├── DGT-TM-preprocess-simpletagged-en-nl.txt.gz
├── DGT-TM-preprocess-simpletagged-en-pt.txt.gz
```

Here we can see an example:

```
Corrigendum to Commission Regulation (EU) No 177/2010 of 2 March 2010 amending
Regulation (EEC) No 2454/93 laying down provisions for the implementation of Co
uncil Regulation (EEC) No 2913/92 establishing the Community Customs Code
06769578-n to Commission_Regulation ( 08173515-n ) 07205104-n 177/2010 of ??] 0
0205885-v 06664051-n ( 08173515-n ) 07205104-n 2454/93 01494310-v 00096089-r 01
057200-n for the 00044150-n of Council_Regulation ( 08173515-n ) 07205104-n 291
3/92 01647229-v the Community_Customs_Code  Corrección de errores del Regla
mento (UE) no 177/2010 de la Comisión, de 2 de marzo de 2010, que modifica el R
eglamento (CEE) no 2454/93, por el que se fijan determinadas disposiciones de a
plicación del Reglamento (CEE) no 2913/92 del Consejo, por el que se establece
el código aduanero comunitario  corrección|n de|c error|n de|c el|c reglamento|
n (|c ue|n )|c no|r 177/2010|c de|c el|c comisión|n ,|c de|c [??|c ,|c que|c mo
dificar|v el|c reglamento|n (|c cee|n )|c no|r 2454/93|c ,|c por|c el|c que|c s
e|c fijar|v determinar|v disposición|n de|c aplicación|n de|c el|c reglamento|n
 (|c cee|n )|c no|r 2913/92|c de|c el|c consejo|n ,|c por|c el|c que|c se|c est
ablecer|v el|c código|n aduanero|a comunitario|a
```

These files should be splited for the use of the WN-Toolkit. The easiest way to do so is using the Linux command **cut**. As the program need the sense-tagged English corpus, we can get it doing, for example:

```
cut -f 2 DGT-TM-preprocess-simpletagged-en-es.txt > DGT-TM-senses-eng.txt
```

and we also need the target language tagged corpus using simple tags:

```
cut -f 4 DGT-TM-preprocess-simpletagged-en-es.txt > DGT-TM-tagged-cat.txt
```

If you're working with Windows, try the CoreUtils for Windows http://gnuwin32.sourceforge.net/packages/coreutils.htm.

EMEA-03 Corpus

We offer a pre-processed version of this corpus for several languages:

```
├── bg-en-preprocess.txt.gz
├── cs-en-preprocess.txt.gz
├── da-en-preprocess.txt.gz
├── de-en-preprocess.txt.gz
├── el-en-preprocess.txt.gz
├── en-es-preprocess.txt.gz
├── en-et-preprocess.txt.gz
├── en-fi-preprocess.txt.gz
├── en-fr-preprocess.txt.gz
├── en-hu-preprocess.txt.gz
├── en-it-preprocess.txt.gz
├── en-lt-preprocess.txt.gz
├── en-lv-preprocess.txt.gz
├── en-mt-preprocess.txt.gz
├── en-nl-preprocess.txt.gz
├── en-pl-preprocess.txt.gz
├── en-pt-preprocess.txt.gz
├── en-ro-preprocess.txt.gz
├── en-sk-preprocess.txt.gz
├── en-sl-preprocess.txt.gz
└── en-sv-preprocess.txt.gz
```

These files have several fields separated by tabulators:

- The English text.
- The English text where sense-tagged words are replaced by their PWN synset. The word sense disambiguation and Tagging has been made using Freeling + UKB.
- The corresponding target language text.

Here we can see an example:

```
Abilify is a medicine containing the active substance aripiprazole. Abilify
 02604760-v a 00612160-n 02701210-v the 00035465-a 05921123-n aripiprazole .
Abilify es un medicamento que contiene el principio activo aripiprazol.
```

Some of these corpora have been further preprocessed and the tagged version (using simple tags) of the target language text have been added. This has been done for the following languages:

```
├─── de-en-preprocess-simpletagged.txt.gz
├── en-es-preprocess-simpletagged.txt.gz
├── en-fr-preprocess-simpletagged.txt.gz
├── en-it-preprocess-simpletagged.txt.gz
├── en-nl-preprocess-sipletagged.txt.gz
├── en-pt-preprocess-simpletagged.txt.gz
```

Here we can see an example:

```
Abilify is a medicine containing the active substance aripiprazole. Abilify
 02604760-v a 00612160-n 02701210-v the 00035465-a 05921123-n aripiprazole .
Abilify es un medicamento que contiene el principio activo aripiprazol. abilify
|n ser|v uno|c medicamento|n que|c contener|v el|c principio|n activo|a aripipr
azol|n .|c
```

These files should be splitted for the use of the WN-Toolkit. The easiest way to do so is using the Linux command **cut**. As the program need the sense-tagged English corpus, we can get it doing, for example:

```
cut -f 2 en-es-preprocess-simpletagged-en-es.txt > EMEA-senses-eng.txt
```

and we also need the target language tagged corpus using simple tags:

```
cut -f 4 en-es-preprocess-simpletagged-en-es.txt > EMEA-tagged-cat.txt
```

If you're working with Windows, try the CoreUtils for Windows http://gnuwin32.sourceforge.net/packages/coreutils.htm.

UNCorpus

We offer a pre-processed version of this corpus for several languages:

```
├── ar-en-preprocess.txt.gz
├── en-es-preprocess.txt.gz
├── en-fr-preprocess.txt.gz
├── en-ru-preprocess.txt.gz
└── en-zh-preprocess.txt.gz
```

These files have several fields separated by tabulators:

- The English text.
- The English text where sense-tagged words are replaced by their PWN synset. The word sense disambiguation and Tagging has been made using Freeling + UKB.
- The corresponding target language text.

Here we can see an example:

```
Adopted at the 81st plenary meeting, on 4 December 2000, on the recommendation
of the Committee (A/55/602/Add.2 and Corr.1, para. 94),The draft resolution rec
ommended in the report was sponsored in the Committee by: Bolivia, Cuba, El Sal
vador, Ghana and Honduras. by a recorded vote of 106 to 1, with 67 abstentions,
 as follows:    02381726-v at the 02194255-a 00528167-a 08307589-n , on ??] , o
n the 06694540-n of the 08324514-n ( A/55/602/Add.2 and Corr.1 , 13671310-n . 9
4 ) , The 13377268-n 06511874-n 00882948-v in the 06681551-n 02445925-v 0221994
0-v in the 08324514-n by Fd 08852843-n , 08750334-n , 08738272-n , 08946187-n a
nd 08737716-n . by a 01000214-v 00183505-n of 106 to 1 , with 67 04882622-n , a
s 02346895-v Fd Aprobada en la 81a. sesión plenaria, celebrada el 4 de diciembr
e de 2000, por recomendación de la Comisión (A/55/602/Add.2, párr. 94),El proye
cto de resolución recomendado en el informe fue patrocinado en la Comisión por
los países siguientes: Bolivia, Cuba, El Salvador, Ghana y Honduras. en votació
n registrada de 106 votos contra uno y 67 abstenciones, como sigue:
```

Some of these corpora have been further preprocessed and the tagged version (using simple tags) of the target language text have been added. This has been done for the following languages:

```
├── en-es-preprocess-simpletagged.txt.gz
├── en-fr-preprocess-simpletagged.txt.gz
├── en-ru-preprocess-simpletagged.txt.gz
```

Here we can see an example:

```
Adopted at the 81st plenary meeting, on 4 December 2000, on the recommendation
of the Committee (A/55/602/Add.2 and Corr.1, para. 94),The draft resolution rec
ommended in the report was sponsored in the Committee by: Bolivia, Cuba, El Sal
vador, Ghana and Honduras. by a recorded vote of 106 to 1, with 67 abstentions,
 as follows:    02381726-v at the 02194255-a 00528167-a 08307589-n , on ??] , o
n the 06694540-n of the 08324514-n ( A/55/602/Add.2 and Corr.1 , 13671310-n . 9
4 ) , The 13377268-n 06511874-n 00882948-v in the 06681551-n 02445925-v 0221994
0-v in the 08324514-n by Fd 08852843-n , 08750334-n , 08738272-n , 08946187-n a
nd 08737716-n . by a 01000214-v 00183505-n of 106 to 1 , with 67 04882622-n , a
s 02346895-v Fd Aprobada en la 81a. sesión plenaria, celebrada el 4 de diciembr
e de 2000, por recomendación de la Comisión (A/55/602/Add.2, párr. 94),El proye
cto de resolución recomendado en el informe fue patrocinado en la Comisión por
los países siguientes: Bolivia, Cuba, El Salvador, Ghana y Honduras. en votació
n registrada de 106 votos contra uno y 67 abstenciones, como sigue: aprobar
|v en|c el|c 81a|c .|c sesión|n plenario|a ,|c celebrar|v el|c [??|c ,|c por|c
recomendación|n de|c el|c comisión|n (|c A/55/602/Add.2|c ,|c párr.|n 94|c )|c
,|c el|n proyecto|n de|c resolución|n recomendar|v en|c el|c informe|n ser|v pa
trocinar|v en|c el|c comisión|n por|c el|c país|n siguiente|a |c bolivia|n ,|c
cuba|n ,|c el_salvador|n ,|c ghana|n y|c honduras|n .|c en|c votación|n registr
ar|v de|c 106|c voto|n contra|c 1|c y|c 67|c abstención|n ,|c como|c seguir|v |
c
```

These files should be splitted for the use of the WN-Toolkit. The easiest way to do so is using the Linux command **cut**. As the program need the sense-tagged English corpus, we can get it doing, for example:

```
cut -f 2 en-es-preprocess-simpletagged-en-es.txt > UNCorpus-senses-eng.txt
```

and we also need the target language tagged corpus using simple tags:

```
cut -f 4 en-es-preprocess-simpletagged-en-es.txt > UNCorpus-tagged-cat.txt
```

If you're working with Windows, try the CoreUtils for Windows http://gnuwin32.sourceforge.net/packages/coreutils.htm.