WSDDE 0.64 – readme

26 czerwca 2012

Mateusz Kopeć

Institute of Computer Science Polish Academy of Sciences

About

The current development version of the environment facilitates the construction and evaluation of WSD methods in the supervised Machine Learning (ML) paradigm.

Homepage: http://zil.ipipan.waw.pl/WSDDE/

Contact person: Mateusz Kopeć [mateusz.kopec@ipipan.waw.pl]

Authors: Rafał Młodzki, Adam Przepiórkowski

1 Requirements

Java Runtime Environement (JRE) 1.6 or newer.

2 Input and output format

The input format for performing WSD experiments is .wsdc (WSDCorpus). Each corpus contains a number of examples of one polysemous word in context. A file in that format should look like following example:

```
SENSES Jezyk1 Jezyk2

CONTEXT 1
SENSE Jezyk1
PREDICTED_SENSE null

TOK -3    czas              czas              subst:sg:nom:m3
TOK -2    chodził           chodzić           praet:sg:m3:imperf
TOK -1    z                 z                 prep:gen:nwok
TOK 0     językiem          język             subst:sg:inst:m3
TOK 1     skręconym         skręcić           ppas:sg:inst:m3:perf:aff
TOK 2     w                 w                 prep:acc:nwok
TOK 3     trąbkę            trąbka            subst:sg:acc:f

END_OF_CONTEXT 1

CONTEXT 2
[...]
```

First line of the file contains possible senses of the polysemous word in that corpus (in example senses are: "Jezyk1" and "Jezyk2"). Then comes an empty line, and after that, a line with word "CONTEXT" and its number. Context has proper sense marked in the next line ("Jezyk1") and automatically predicted sense ("null"), if this corpus was already annotated by WSDDE.

After these information and an empty line the context is listed. Each line beginning with "TOK" represents a single token. A number after "TOK" indicates the position of the token relative to the polysemous word in context. After it comes the orthographic form of token ("chodził"), followed by base form("chodzić"), and finally the morphosyntactic information ("praet:sg:m3:imperf").

Context is ended by an empty line and line with "END_OF_CONTEXT" and number of this context. Next contexts are represented further, till the end of file.

3 Installation

One only needs to unpack the zipped file to use most of the environment. The structure of directories after unzipping looks as follows:

```
/wsdde.jar
/config.ini
/readme_en.txt
/readme_pl.txt
/wsdde.pdf (an article from LTC'09)
/wsdde_lib (external libraries used by the WSDDE)
/doc/gpl.txt (the gpl license)
/source
/resources
    /corpora_raw (a small, hand-annotated corpus and pseudowords corpora)
    /corpora_enriched (corpora enriched with extra information (e.g. POS))
    /experiment_meta_descriptions (some experiment's meta descriptions)
    /experiment_descriptions (some experiment's meta descriptions)
    /results
    /wsdmethods (some WSD methods ready to load and use)
```

To achieve the full functionality, the WSDDE uses external libraries. All the necessary libraries are in `wsdde_lib` directory:

- commons-cli.jar (1.2) – command line
- weka.jar (3.6.1) – algorithms of selection and machine learning
- mysql.jar (5.1.10) – jdbc connector to the MYSQL; it is required only if you use the MYSQL. The WSDDE user should have access to the MYSQL server and provide (config.ini) server address, port, DB name, DB user and password.
- poliqarp.jar (1.3.7) – client library for poliqarpd. It is used only if you want to generate pseudowords corpora.

The numbers in brackets are the versions which are currently used and work properly. In the case of MySQL and Poliqarp only the client-part is bundled with the WSDDE. One should give the servers' settings in config.ini file. The newest version of poliqarp is accessible at sourceforge.net. Corpora which are used by poliqarp server are at nkjp.pl or korpus.pl.

The environment can also use the TAKIPI tagger (http://nlp.ipipan.waw.pl/TaKIPI/). If you want to use it, you should install TAKIPI and edit config.ini (you should put the path to TaKIPI's executable file).

4 User guide

The use of the WSDDE mainly from the command line is recommended. Command:

```
java -jar wsdde.jar
```

shows available options, which should be passed as the program's parameter(s):

```
java -jar wsdde.jar [options]
```

In case of a no-memory error, one should use the -XmxMEMm (MEM = memory in megabytes) switch for JVM, e.g.

```
java -Xmx1000m -jar wsdde.jar [options]
```

All input files (corpora, experiments' descriptions, etc.) should be UTF-8 encoded. If you give the output path, all the given directories must exist.

4.1 Example use scenario

Here is an example scenario on how to use WSDDE:

1. create a pseudowords corpus (check poliqarp properites in config.ini, run poliqarp server)
   `java -jar wsdde.jar -pws -wc parlament=500 dom=500 -o pws.xml`
   This generates file named pws.xml with 1000 context of pseudoword parlement-dom (izba).
2. convert corpus to the enriched format (check if takipi propeties is configured in config.ini or use -simple option)
   `java -jar wsdde.jar -enr -i pws.xml -takipi -o pws.wsdc`
3. split the corpus into train and test corpora
   `java -jar wsdde.jar -spl -i pws.wsdc -ss 500 500 -p pws`
4. create the experiment's description from the metadescription (edit meta.desc; to obtain syntax, just run "`java -jar wsdde.jar -xsd`")
   `java -jar wsdde.jar -gmd -i metadesc.xml -o desc.xml`
5. conduct the experiment
   `java -Xmx1000m -jar wsdde.jar -coe -i desc.xml -o result.xml`

5 For developers

If you want to create your own feature generator, you should extend the abstract class wsdde.generator.FeatureGenerator. Moreover, the extending class should be in the package wsdde.genearator. You should also put XML schemas (xsd) describing generator's parameters into the wsdde.generator package. It is a good idea to look at built-in feature generators.

In the current version, using extra knowledge sources (e.g. wordnet, shallow parsing) is not supported by the architecture. Of course, you can do it on your own. In the future, when the environment accepts NKJP annotation standards, supporting of extra knowledge sources will be part of the architecture.