Addicter Tutorial

Jan Berka

February 17, 2012

Introduction

Addicter is a software tool for:

- Automatic error detection and classification
- Visual analysis of machine translation
- Monolingual alignment

Operated via

- Command line (experiment preparation, error detection, alignments)
- Web browser (visual analysis)

Written in Perl

Installation

- Perl interpreter
- Checkout additional libraries: `svn --username public checkout https://svn.ms.mff.cuni.cz/svn/dzlib ~/lib`
  `export PERL5LIB=~/lib:$PERL5LIB`
- Checkout Addicter: `svn --username public checkout https://svn.ms.mff.cuni.cz/svn/statmt/trunk/addicter addicter`
- You don't need to install or configure Apache or other web server – Addicter has its own lightweight web server

Addicter in Web Browser

Main page: list of experiments
Experiment main page

- Word Explorer
- Test Data Browser
- Error Summary

Word Explorer Page

Informations about given word
Table with source and target side:

- first row – source side
- second row – target side words aligned to the source
- third and fourth row – target side and source words aligned to it

Alignment summary

Test Data Browser Page

Informations about given sentence
Table

- Five pairs of rows
- In each pair, the first row is the first side written as it goes
- The second are words from the second side aligned to the first (together with the alignment)
- Found erroneous tokens color-labeled
- Words in table are links to the Word Explorer
- You can switch between alignments

Sentence error summary under the table

Error Summary Page

Summary of errors of the test data
Table with counts of different error types
Links to sentences with given error occurence (to Test Data Browser) under the table

Preparation Steps

In order to use Addicter, you should follow these few simple steps

- Get data into defined file structure
- (Optional) Train index
- (Optional) Get different alignments
- Run error detection

Addicter File Tree

```
Addicter folder
|-- cgi
|   |-- Experiment folder
|       |-- test.src
|       |-- test.tgt
|       |-- test.system.tgt
|-- prepare
|   |-- addictindex.pl
|   |-- detecter.pl
|-- server.pl
\-- testchamber
    \-- other Perl scripts and libraries
```

Train index

In order to use the word explorer, you need to run an indexing script over your aligned index
It will create a bunch of index files that will later tell the viewer where to look for examples of a particular word
You need:

- `train.src` – source side of training corpus
- `train.tgt` – target side of training corpus
- `train.ali` – alignment of training corpus

Optional:

- `test.src` – source side of test data
- `test.tgt` – reference translation of test data
- `test.ali` – alignment of the source and reference translation of test data
- `test.system.tgt` – system output for test data
- `test.system.ali` – alignment of the source and the system output for test data

Train index

Mandatory files only:

- Go into experiment folder
- Run `../../prepare/addictindex.pl -trs train.src -trt train.tgt -tra train.ali -oprf s`
- Run `../../prepare/addictindex.pl -trs train.src -trt train.tgt -tra train.ali -oprf t -target`

Also optional files:
Like before, but also with `-s test.src -r test.tgt -h test.system.tgt -ra test.ali -ha test.system.ali`

Get different alignments

Error detection works with reference to hypothesis alignment

- Addicter currently implements LCS, HMM, injective greedy alignment
- Each ref-hyp alignment must be in its own subfolder of the experiment folder
- Each ref-hyp alignment must be named test.refhyp.ali

Addicter alignment algorithms examples

- `../../testchamber/align-hmm.pl test.tgt test.system.tgt >./HMM/test.refhyp.ali`
- `../../testchamber/align-lcs.pl test.tgt test.system.tgt >./LCS/test.refhyp.ali`

File Tree Again

```
cgi
|-- Experiment 1
|    |-- HMM
|        |-- test.refhyp.ali
|    |-- LCS
|        |-- test.refhyp.ali
|    |-- test.src
|    |-- test.tgt
|    |-- test.system.tgt
|    |-- bunch of index files
|-- Experiment 2
|    |-- same structure
\-- ...
```

Error Detection

Automatic error detection done by prepare/detecter.pl

Mandatory files:

- `test.src`
- `test.tgt`
- `test.system.tgt`

Optional:

- `test.refhyp.ali`

Usage:

```
./detecter.pl -s test.src -r test.tgt -h test.system.tgt
[-a test.refhyp.ali -w working directory]
```

Working directory should be a subfolder of experiment folder

If you don't use the -a option, detecter.pl will use Addicter's injective greedy aligner

Error Detection Example

In experiment 1 folder

```
../../prepare/detecter.pl -s test.src -r test.tgt -h
test.system.tgt -a HMM/test.refhyp.ali -w HMM
```

Resulting file tree

```
Experiment 1
|-- HMM
|   |-- test.refhyp.ali
|   |-- tcali.txt
|   |-- tcerr.txt
|-- LCS
|   |-- test.refhyp.ali
|-- test.src
|-- test.tgt
|-- test.system.tgt
\-- bunch of index files
```

- `./server.pl port_number` – starts the Addicter server at given port
- Addicter main page – list of experiments
- Experiment main page – first few sentences, links to
  - Test Data Browser – browsing through test sentences with view on src, ref, hyp, alignments and detected errors
  - Word Explorer (if you ran addictindex.pl) – exploring training (and test) data
  - Error Summary – global view on error types occurences and links to sentences with given errors

File Tree

Create experiment folder, copy training and test data

```
Addicter
|-- cgi
    |-- Tamil-English
        |-- test.src
        |-- test.tgt
        |-- test.system.tgt
        |-- train.src
        |-- train.tgt
        \-- train.ali
```

Run indexer

From experiment folder run

```
../../prepare/addictindex -trs train.src -trt train.tgt
-tra train.ali -oprs s
../../prepare/addictindex -trs train.src -trt train.tgt
-tra train.ali -oprs t -target
```

Index files now created, located in experiment folder

Get different alignments

From experiment folder run

```
../../testchamber/align-lcs.pl test.tgt test.system.tgt
>./LCS/test.refhyp.ali
../../testchamber/align-hmm.pl test.tgt test.system.tgt
>./HMM/test.refhyp.ali
../../testchamber/align-greedy.pl test.tgt test.system.tgt
>./Greedy/test.refhyp.ali
```

Run error detection with created alignments

From experiment folder run

```
../../prepare/detecter.pl -s test.src -r test.tgt -h
test.system.tgt -a LCS/test.refhyp.ali -w LCS
../../prepare/detecter.pl -s test.src -r test.tgt -h
test.system.tgt -a HMM/test.refhyp.ali -w HMM
../../prepare/detecter.pl -s test.src -r test.tgt -h
test.system.tgt -a Greedy/test.refhyp.ali -w Greedy
```

Resulting file tree (without indexes and training data)

```
Tamil-English
|-- LCS
|   |-- test.refhyp.ali
|   |-- tcali.txt
|   |-- tcerr.txt
|-- HMM
|   |-- test.refhyp.ali
|   |-- tcali.txt
|   |-- tcerr.txt
|-- Greedy
|   |-- test.refhyp.ali
|   |-- tcali.txt
|   |-- tcerr.txt
|-- test.src
|-- test.tgt
\-- test.system.tgt
```

Run server

From cgi folder run `./server.pl port_number`:
Please contact me at:
<URL:http://computer_name:port_number/cgi/index.pl>

References

Daniel Zeman, Mark Fishel, Jan Berka, Ondřej Bojar: Addicter: What Is Wrong with My Translations?, PBML 96
(http://ufal.mff.cuni.cz/pbml/96/art-zeman-fishel-berka-bojar.pdf)
Addicter wiki pages
https://wiki.ufal.ms.mff.cuni.cz/user:zeman:addicter