onlineMgiza++

User Manual

N. Bertoldi, A. M. Farajian
FBK-irst, Trento, Italy

November 30, 2015

(Version 1.0.5)

onlineMgiza++ is an extension of **Mgiza++** well-suited for the real-time applications in which the sentence pairs are required to be word-aligned, one at a time.

Users of this toolkit might cite in their publications:

> A. M. Farajian, N. Bertoldi, M. Federico, *Online Word Alignment for Online Adaptive Machine Translation*, Proceedings of EACL 2014 Workshop on Humans and Computer-assisted Translation, Gothenburg, Sweden, 2014, pp. 84-92.

onlineMgiza++ has been partially funded by the EU project MateCat (ICT-2011.4.2-287688).

The official website of **onlineMgiza++**, containing this manual, source code, and examples is:

`www.mt4cat.org/software/onlinemgiza`

[1] `http://www.gnu.org/licenses/gpl-3.0.html`

1 Description

With respect to its parent, **onlineMgiza++** permits to align at word level any new sentence pairs on-the-fly without the need to reload the models, and not only in a batch modality.

Nevertheless, **onlineMgiza++** inherits all functions of **Mgiza++**, in particular those to train word-alignment models.

Mgiza++ permits to align at word level a new set of sentence pairs, but these pairs are read from file, and the generated word alignments are saved into file as well. **Mgiza++** permits to load pre-trained models, and to restart their training procedure on a parallel corpora. The final models obtained after the conclusion of the re-started training are dumped into files.

onlineMgiza++ extends these functions, permitting word alignment in an online modality; it reads a sentence pairs input from stdin, and outputs its computed word alignment to stdout, by loading pre-trained models just once.

The current version of the **onlineMgiza++** does not support incremental training. In the other words, it does not adapt to the new sentence pairs, hence the models do not change over time. In the current version of **onlineMgiza++**, in order to handle the new words appearing in the new test sentence pair for which no information is available in the models, we simply map them to the UNK token. But to be able to retrieve the correct words to be printed in the final output, we need to store the index of the new words in the local vocabularies.

1.1 Architecture

onlineMgiza++ consists of two main modules: `mgizaServer` and `mgizaClient`. `mgizaServer` is responsible for computing the alignment of the given sentence pairs. To avoid unnecessary I/O operations, `mgizaServer` loads all the required models once at the beginning of the alignment session, and releases them at the end. `mgizaClient` is in charge of reading the input sentence pair from stdin, communicating with `mgizaServer`, and writing the computed word alignment to stdout.

`mgizaServer` and `mgizaClient` run in two different threads.

2 Installation

onlineMgiza++ is compiled by means of GNU make.
Either AutoTools[2] or CMake[3] can be chosen to configure the compilation scripts.

2.1 Install with AutoTools

To install **onlineMgiza++**, the following packages are required:

- GNU compiler
- Boost library (system and thread) (version 1.49 or higher)
- AutoTools

Go to the root directory of **onlineMgiza++** source code:

```
cd onlineMgiza++
```

Set BOOST variables

```
export BOOST_ROOT=/path/to/boost/root
export CPLUS_INCLUDE_PATH=${BOOST_ROOT}/include:${CPLUS_INCLUDE_PATH}
export LIBRARY_PATH=${BOOST_ROOT}/lib:${LIBRARY_PATH}
```

Prepare Makefiles using AutoTools:

```
./regenerate_makefiles.sh --force
./configure [--prefix=/Path/to/the/installation/directory]
```

To install the documentation please add ”`--enable-doc`” on the configuration command line. Note that pdf manual is compiled and installed under ”PREFIX/doc/” only if `pdflatex` exists.

Compile and install:

```
make
make install
```

2.2 Install with CMake

To install onlineMGIZA++, the following packages are required:

- GNU compiler
- Boost library (system and thread) (version 1.49 or higher)
- CMake

Go to the root directory of **onlineMgiza++** source code:

[2] www.gnu.org/software/automake/manual/html_node/Autotools-Introduction.html

[3] www.cmake.org

```
cd onlineMgiza++
```

Set BOOST variables

```
export BOOST_ROOT=/path/to/boost/root
export BOOST_LIBRARYDIR=$BOOST_ROOT/lib
```

Prepare Makefiles using CMake:

```
cmake [-DCMAKE_INSTALL_PREFIX:PATH=/Path/to/the/installation/directory] .
```

Compile and install:

```
make
make install
```

Note that pdf manual is compiled and installed under ”PREFIX/doc/” only if pdflatex exists.

3 Usage

To run **onlineMgiza++** in online modality, execute the following command:

```
mgiza configfile -onlineMode 1 [options]
```

The configuration file (configfile) must be set as explained in Section 4. The parameter ”-onlineMode 1” enables the online modality. The parameters specified in ”options” overwrites those specified in ”configfile”; The parameter ”-o” is mandatory, and refers to the directory where temporary files are created.
The sentence pair is input from stdin in this format:

```
<src> source sentence </src><trg> target sentence </trg>
```

like in the following example

```
<src>this is the english sentence</src><trg>questa  la frase inglese</trg>
```

The word alignment computed is output in the standard **Mgiza++** format:

```
# Sentence pair (1) source length 5 target length 5 alignment score : 7.84911e-59
questa  la frase inglese
 ({ 4 5 }) this ({ 1 }) is ({ 2 }) the ({ 3 }) english ({ }) sentence ({ })
```

4 Configuration

The parameter `onlineMode` is the switch (`0|1`) between the two modalities. By default, the offline modality is set.
All parameters by **Mgiza++** are inherited by **onlineMgiza++**.
To use **onlineMgiza++** in the online mode, the following parameters are required:

```
previoust /path/to/*.t-table*.[1-9|final]
previousa /path/to/*.a-table*.[1-9|final]
previousn /path/to/*.n-table*.[1-9|final]
previousd /path/to/*.d-table*.[1-9|final]
previousd4 /path/to/*.d-table4.[1-9|final]
previousd42 /path/to/*.d-table42.[1-9|final]
previousd5 /path/to/*.d-table5.[1-9|final]
previoushmm /path/to/*.h-tablehmm.[1-9]
coocurrencefile /path/to/training.cooc
```

These files are dumped in the training modality if **onlineMgiza++** is run with the following parameters:

```
nodumps 0
onlyaldumps 1
hmmdumpfrequency N
```

where `N` is the number of iterations set for the HMM (i.e. ”`mh N`”).
Furthermore, the parameter `restart` is required to set the models to load; the restart level must agree with the loaded models, as stated in the **Mgiza++** documentation and here summarized.

```
restart 1:  restart from Model1
restart 3:  restart from Model2
restart 6:  restart from HMM
restart 9:  restart from Model3
restart 11: restart from Model4
```

4.1 Notes

- It is highly convenient using the configuration file produced during the training, and modifying it to enable the online modality and set the right models.
- The default filenames of the dumped trained models of **Mgiza++** can be ambiguous on case-insensitive file systems, like in Mac OSx;[4] To have unambiguous default filenames on all file systems, **onlineMgiza++** redefines the filenames of the dumped trained models according to the following table. Moreover, ”`D4`” is modified into ”`d-table42`”.

[4] By default, Mac OSx file system is case-insensitive, although it can be formatted in case-sensitive modality.

table type	New suffix	Old suffix
lexical probs	t-table	t
	t-table2to3	t2to3
	t-table3	t3
	t-tablehmm	thmm
fertility probs	n-table	n
	n-table2to3	n2to3
	n-table3	n3
alignment probs	a-table	a
	a-table2to3	a2to3
	a-table3	a3
	a-tablehmm	ahmm
distortion probs	d-table2to3	d2to3
	d-table3	d3
	d-table3	d4
	d-table42	D4
	d-table5	d5
null insertions probs	p0-table	p0
	p0-table2to3	p0_2to3
	p0-table3	p0_3
zero fertility probs	fe0-table3	fe0_3
jump prob	h-tablehmm	hhmm

5 Release Notes