Introduction

TM Cleaner is software for identifying the wrong translation units (that contain segments that are not translation of each other) in translation memories or parallel corpora. The identification of these TU is stated as a classification task: the software returns "1" if it thinks that the TU is correct and "0" otherwise.

TM Cleaner can work with any classification algorithm implemented in scikit-learn provided a connection to the algorithm is provided. TM Cleaner needs training data to create its models.

TM Cleaner works in three mutual exclusive modalities:

1. **Bing**. It uses Bing translation engine to translate the source segment and then uses a similarity measure between the target segment and the translation of the source
2. **Hunalign**. It uses the Hunalign alignment score as a feature.
3. **Fastalign**. It computes various features based on the word alignments in the source and target segments.

TM Cleaner allows you to train a model, to classify new data and to evaluate the classifier performance.

TM Cleaner has been tested on Linux and Mac OS X for Bing and Hunalign modalities. In Fastalign modality it only works on Linux because it relies MMT (Modern Machine Translation) distribution that only compiles on Linux.

Install

1. Checkout the python source code from the "git" repository.

The software you need before downloading TM Cleaner is:

2. Python 2.7.x or higher.
3. Java 8 or higher.
4. Scikit-learn 0.17.x or higher.
    a. To check your version of scikit-learn do:

```
>>>  import sklearn
>>>  print('{}.'.format(sklearn.__version__))
```

5. To work with Bing Translation engine you need an application identifier provided by Bing. For details see the tutorial on working with Bing Translation Engine.
6. If you want to work with Hunalign sentence aligner you need to download and install Hunalign. For details see the tutorial on working with Hunalign sentence aligner.
7. If you want to work with Fastalign word aligner you need to download and install MMT (Modern Machine Translation) java application. For details see the tutorial on working with Fastalign word aligner.

Additional software used

For language identification we use Cybozu language identifier. The language identifier is used through a java program that is called from the main script. We have bundled with this distribution the language profiles used by Cybozu.
In Fastalign modality we use a java program that merges the backward and forward alignments. Like Cybozu, this program is called from the main python script and is provided with this distribution.

Input files and training file.

All input files should be placed in a directory and should be in utf-8 format. Each file is composed of a number of lines separated by the end of line character ("\n"). Each line should have the following mandatory fields separated by "@#@":

1. Identifier_1 = it starts with 0 and is incremented for each line
2. Identifier_2= in the example this field is the copy of the previous one. However the user can put whatever information s/he wants (for example a database id)
3. Source Segment =The source segment
4. Target Segment= The target segment
5. The training file should contain the category (0 or 1) as the last item.

Example of a small file for English Italian language-pair:

0@#@0@#@Epistle of Jude@#@Lettera di Giuda
1@#@1@#@Tel: +351-21 000 86 00 Romania Novartis Animal Health d. o. o.@#@Tel: +351-21 000 86 00

Output.

The output will be created in a directory called "Classified" inside the input directory. WARNING: at the next run the directory Classified will be deleted and recreated. Therefore, after a run, move the classified files somewhere else.

The format of the output file is the following:

1. Identifier_1 =as in the input
2. Identifier_2=as in the input
3. For some algorithms like Logistic Regression we can output the probabilities for the classes 0 and 1
4. The rule that decided the output ("ML" stays for machine learning)
5. The inferred category (0 or 1)

Example of the previous file classified:

0@#@0 @#@0.04-0.96@#@ML@#@1
1@#@1@#@0.89-0.11@#@ML@#@0

Configuration files:

The configuration files are under the directory Parameters. For each modality (Bing, Hunalign, Fastalign) you will find a directory with the configuration files for training ("p-Training-XXX.txt") and testing ("p-Batch-XXX.txt"). The parameters are commented in each file on lines starting with the symbol "#".

To understand the parameters please read a commented configuration file.

What Next

To see how to run the software in each of the three modalities, please, read the corresponding tutorials.