Tutorial Hunalign (v.1.0)

August 19, 2016

Using TM Cleaner with Hunalign

In this modality the software uses an extra-feature equal to the normalized alignment score of source and target segments.

If you use Hunalign with small translation memories please take care to provide a very good bilingual dictionary otherwise the alignment scores will be meaningless. Tm-Cleaner provides three bilingual dictionaries (see the Resource section) to help you dealing with data in English-Italian, English-French and English-German.

To install Hunalign follow the instructions for installation:

http://mokk.bme.hu/resources/hunalign/

TM Cleaner Configuration File

1. *Parameters/p-HunAlign.txt*. The file contains two documented parameters:
   a. The full path to the Hunalign executable.
   b. The path to the bilingual dictionary to be used by Hunalign

To run the tutorial you should provide the full path to the Hunalign executable.

Resources

We provide three bilingual dictionaries under *Resources/Dictionaries*:

- An English-Italian dictionary
- An English-French dictionary
- An English-German dictionary

The resources that can be used with this tutorial are following:

1. An *English-Italian* scikit-learn model trained on a sample of English-Italian positive and negative bi-segments extracted from MyMemory: Training/full-English-Italian-Features.csv
2. An *English-Italian* test file obtained from automatically aligning a web site containing English and Italian parallel documents. The file contains positive and negative segments.

   a. The file to be classified: *Resources/Examples/Hunalign/about-small-en-it.txt*
   b. The file annotated with correct labels for evaluation: *Resources/Examples/Hunalign/about-small-en-it-annotated.txt*

3. An *English-French* test file obtained from automatically aligning a web site containing English and French parallel documents. The file contains positive and negative segments.
   a. The file to be classified: *Resources/Examples/Hunalign/sample-en-fr.txt*
   b. The file annotated with correct labels for evaluation: *Resources/Examples/Hunalign/sample-en-fr-annotated.txt*

Training:

The configuration parameters for training are in "*Parameters/Hunalign/p-Training-XXX.txt*" files. You should copy and edit the corresponding file to fit your purposes.

```
python generateFeaturesAndClassify.py --features --config
Parameters/Hunalign/p-Training-Italian.txt
```

For this tutorial you do not need to train: we did the training for you and obtained the model presented in the previous section.

Classification:

English-Italian example.

1. Create an empty directory (*mkdir TestFiles*) and copy the English Italian test file inside the "TestFiles" directory taking care that the TestFiles directory is empty. Also check that p-Hunalign.txt points to English-Italian dictionary.
   a. cp Resources/Examples/Hunalign/about-small-en-it.txt TestFiles/
2. Classification using the default algorithm "SVM with linear kernel"
   a. python generateFeaturesAndClassify.py --classify --config Parameters/Hunalign/p-Batch-Italian.txt

3. Classification using the algorithm “Logistic regression” with the default class 0 and the threshold 0.7 (To see what this means read the configuration file)
   a.
   ```
   python generateFeaturesAndClassify.py --classify --config Parameters/Hunalign/p-Batch-Italian.txt --mlalgorithm LogisticRegression
   ```

English-French example:

1. Copy the English French test file inside the “TestFiles” directory taking care that the TestFiles directory is empty. Also check that p-Hunalign.txt points to English-French dictionary.
   a.
   ```
   cp Resources/Examples/Hunalign/sample-en-fr.txt TestFiles
   ```
2. Classification using the default algorithm “SVM with linear kernel”. In classification we use the model train on English-Italian and the English-French dictionary. This might be a good idea given that Italian and French have many similarities.
   a.
   ```
   python generateFeaturesAndClassify.py --classify --config Parameters/Hunalign/p-Batch-French.txt
   ```
3. Classification using the algorithm “Logistic regression” with the default class 0 and the threshold 0.7 (To see what this means read the configuration file)
   a.
   ```
   python generateFeaturesAndClassify.py --classify --config Parameters/Hunalign/p-Batch-French.txt --mlalgorithm LogisticRegression
   ```

Evaluation:

To see how well the algorithms performed look inside the directory: “Resources/Examples/Hunalign/Evaluation”

To perform the evaluation and know about each file returned by the evaluation script read the Evaluation tutorial.