WCE-LIC: a Word Confidence Estimation toolkit for Machine Translation

April 4, 2016

Abstract

This is a beta version of the documentation for WCE-LIG you will find some additional data to adapt the toolkit to your needs.

Document

2016-04-04: Creation by Christophe Servan

Contents

1 Introduction

This toolkit, written in python (python3), enables you to estimate the quality of an automatic translation at word level. It outputs a good (G) or a bad (B) label for each word in of the translation hypothesis. For instance:

Source:	give me some pills
Translation hypothesis:	me donner des pilules
WCE:	B B G G
Human post-edition:	donnes moi des pilules

1.1 Features extracted

Here is the list of all the features which are used in the toolkit. Detailed description can be founded if the paper directory.

1 Proper Name	17 Left Target POS	25 WPP Exact
2 Unknown Stemming	18 Left Target Word	26 WPP Any
3 Number of Word Occurrences	19 Left Target Stem	27 Max
4 Number of Stemming Occurrences	20 Right Target POS	28 Min
5 Source POS	21 Right Target Word	29 Nodes
6 Source Word	22 Right Target Stem	30 Constituent Label
7 Source Stem	15 Target Word	31 Distance To Root
8 Left Source POS	16 Target Stem	32 Numeric
9 Left Source Word	17 Left Target POS	33 Punctuation
10 Left Source Stem	18 Left Target Word	34 Stop Word
11 Right Source POS	19 Left Target Stem	35 Occur in Google Translate
12 Right Source Word	20 Right Target POS	36 Occur in Bing Translator
13 Right Source Stem	21 Right Target Word	37 Polysemy Count – Target
14 Target POS	22 Right Target Stem	38 Backoff Behaviour – Target
15 Target Word	23 Longest Target N-gram Length	
16 Target Stem	24 Longest Source N-gram Length	

This toolkit have been created to be released to the community and especially to reproduce experiments done in our publications. This toolkit can easily be addepted to another languages wince you can ma de some modifications in the code.

1.2 Requirements

- Set the WCE_ROOT environment variable (see Readme file)
- python3
- PyYAML-3.11
- NLTK for python 3
- tools: see tools directory
- 7zip to decompress data in the input_data directory

2 Directory organisation

Here is the directory organisation of the toolkit:

```
|
|__WGE-LIG/
   |__docs/
   |__input_data/
      |__tools/
      |  |__BabelSenseCount_v25/
      |  |__berkeley_parser/
      |  |__fastnc/
      |  |__giza-pp/
      |  |__moses/
      |  |__srilm-1.7.1/
      |  |__tercpp/
      |  |__terplus/
      |  |__treetagger/
      |  |__wapiti-1.5.0/
      |__wce_system/
         |__common_module/
         |__config/
         |__feature/
         |__lib/
         |__metrics/
         |__preprocessing/
         |__solution/
         |__var/
```

3 Basic usage

There are three stages needed to use the toolkit as it is:

1. Pre-processing
2. Feature extraction
3. Classification process

the parameters in the configuration file `input_data/config_end_user.yml` must be set correctly. As parameters are quite verbose, we do not detail them here (e.g.: *source_language* is the code of the source language used, like “fr”, “en”...).

3.1 Pre-processing

The pre-processing is launch by using the script: preprocessing/pre_processing.py

3.2 Feature Extraction

The feature extraction process is launch by using the script: feature/extract_all_features.py

3.3 Classification Process

The classification process is launch by using the script: metrics/demo_metrics.py

3.4 all-in-One

all the processes can be launch with one single script: solution/lauching_solution.py

4 Advance usage

We present in this section some possibilities.

4.1 Language changing

Of course you cold be interested in changing the language you seek for. For instance, you want to target german. you have to modify the following configuration file `input_data/config_end_user.yml`:

- tool_babel_net_de: /tools/BabelSenseCount_v25/BabelNet-2.5/calculateSenses_german.sh
- treetagger_german: /tools/treetagger/cmd/tree-tagger-german
- grammar_de_for_berkeley_parser_path: /tools/berkeley_parser/<*name of the grammar file*>

and of course set up correctly other parameters like:

- source_language: en
- target_language: de
- language_pair: en_de
- ...

Then, to use BabelNet, you have to create a script base on the following one : `/tools/BabelSenseCount_v25/BabelNet-2.5/calculateSenses_french.sh` [1]

[1] this should change in next version

4.2 Advance options

Some options are available only in `wce_system/config/configuration.yml`. For example, you can set the number of threads possible to use, or the features you want to use (binary option).