TTagger - Documentation

July 12, 2017

1 Installation

This section describes how to setup the tagger on your computer. The following software prerequisites have to be fulfilled prior to installation:

- gcc (Unix-based system like Linux is assumed)
- log4cpp
- SRILM
- libsvm
- CRFSuite
- crf++
- boost
- Python

To install the tagger, do the following:

- unpack the tagger archive
- switch to the tagger directory
- aclocal
- autoconf
- configure TEMP_DIR=<temporary_directory> SRILM_INCLUDE=<directory> SRILM_LIB=<directory> LIBSVM_INCLUDE=<directory> LIBSVM_LIB=<directory>
- make
- make install
- make install_resources

- make install_resources copies the resources in the associated directories and creates a file called .ttagger_la.conf_gen in your home directory. You should rename this file to .ttagger_la.conf and potentially adjust it to your personal needs.

SRILM_INCLUDE denotes the directory where your srilm include files reside (analogously for SRILM_LIB, LIBSVM_INCLUDE AND LIBSVM_LIB)

2 Lexicon

The tagger supports the use of a lexicon. We implemented a parsing routines for the Collex.LA (language code=la) and Collex.EN lexicon (language code=la).

We describe in the following the format of our Latin lexicon. To use this lexicon representation, just specify -l la as argument on the command line. All attributes are tab separated and preceded by the '#' character.

The following columns are used (must show up in this sequence):

- SL-ID: id of the superlemma, ignored by the Tagger
- SL-Name: name of the superlemma, ignored by the Tagger
- SPos: superlemma pos, ignored by the Tagger
- L-ID: id of the lemma, ignored by the Tagger
- L-Name: name of the lemma
- LemmaSource: origin of the lemma, for instance Wiktionary, ignored by the Tagger
- LPos: Lemma part of speech, ignored by the Tagger
- WF-ID: id of the word form
- WF-Name: actual word form
- Case: grammatical case
- ComparisonDegree: comparison degree for adjectives and adverbs (positive, comparative or superlative)
- ConjugationType: for verbs you can specify the conjugation type here (ignored by the tagger)
- DeclensionType: for nouns you can specify the declension type here (ignored by the tagger)
- Gender, i.e., masculine, feminine, neuter or common
- Mood, e.g., infinitive, imperative, subjunctive, supine, ...

- Number, singular or plural
- Person, 1,2, or 3
- WPos part of speech of the word form
- Pronoun type, e.g., interrogative pronoun or relative pronoun
- Tense, e.g., present or past
- Source WF, origin of the source, e.g. Wiktionary, ignored by the tagger
- Source Lemma, origin of the lemma, e.g. Wiktionary, ignored by the tagger
- Source Pos, origin of the pos, e.g. Wiktionary, ignored by the tagger

Note that the field indicated with 'ignored by the tagger' can have arbitrary content but must still show up.

3 Training the part of speech tagger

The tagger is trained with files in CONLL format. To do this, switch to the directory training and type:
CrfTrainingSetCreator -l la (for the Collex.LA lexicon format which is recommended to use) -c input_file -L lexicon -o model_input_file Note that the input file must be given in conll format. Afterwards you have to create the model via CrfSuite, which is accomplished by entering: crfsuite learn -m model_file model_input_file.

4 Training the morphological tagger

Training the morphological tagger is accomplished analogously to training the part of speech tagger. The only difference is that you need to create model data from each morphological training file (as well as the partial joint model input file) as well and put all the model files in the same directory if you apply the tagger to new text. The tagger will automatically take into account such files

5 Applying the tagger

To apply the tagger enter the following: ./TTagger -R -l language_code (e.g., la) -c -i input_filename -o output_filename -L lexicon_filename. The parameter *-R* is used to indicate that the input is specified in conll format. Otherwise, it is assumed that you want to tag plain text.

6 Specifying tagger features

For specifying tagger features just create a .ttager_language_code.cfg file in your home directory. A minimal configuration file looks like this: # example logging file

version = "1.0";
ttagger:

base_directory="/home/heinz/prog/TTagger/resources";
lexicon_filename="/home/heinz/lexicons/wiktionary_najock_collexla8.csv";
crf_model_filename="/home/heinz/experiments_lrec/crf_model_allpos11.model";
rules_filename="/home/heinz/prog/TTagger/resources/rules/rules_la.xml";
selectors="rest,suffix,word,lemmatizer_suffix";
features="ngram,number,capitalization,skipgram_multiple,pos_multiple";
;

The tagger features are specified with the keys selectors and features.