TextConverter documentation

Expected input:

Truecased text with one sentence per line.

Output:

Input text, which is tokenized and converted to colloquial form. Casing is the same as the input text.

Requirements:

MorphoDiTa C# binding.

A tagger model and a morphological dictionary for MorphoDiTa from “Czech models (MorfFlex CZ 161115 + PDT 3.0) for MorphoDiTa 161115”.

To get MorphoDiTa C# binding, do the following:

1) git clone https://github.com/ufal/morphodita.git
2) cd morphodita/bindings/csharp
3) make
4) copy the file morphodita/bindings/csharp/libmorphodita_csharp.so to the folder with the TextConverter.exe

The text converter should contain the model and dictionary files needed. If needed though, the files can be downloaded using these instructions:

1) curl --remote-name-all https://lindat.mff.cuni.cz/repository/xmlui/bitstream/handle/11234/1-1836{/czech-morfflex-pdt-161115.zip}
2) unzip czech-morfflex-pdt-161115.zip
3) The tagger model is czech-morfflex-pdt-161115/czech-morfflex-pdt-161115.tagger
4) The morphological dictionary is czech-morfflex-pdt-161115/ czech-morfflex-161115.dict

The converter expects truecased text (which implies tokenized text). Both the tokenizer and the truecaser can be found here: https://github.com/moses-smt/mosesdecoder/tree/master/scripts

To convert text, first create an instance of the class TextConverter. This instance must be initialized by calling the method Initialize(). If the initialization was successful, the method will return true. After a successful initialization, you can convert text by calling the method Convert(), which takes a TextReader for reading the input and a TextWriter for writing the output as it's arguments.

The conversion algorithm:

Read one line from the input (equivalent to one sentence because of the expected input format).

Use the MorphoDiTa tagger to tag the sentence.

Look for a possible replacement for the current word. If found, decide randomly whether or not to use the replacement (60% chance that it will be used).

If yes, get the correct form of the replacement word by calling the method GetReplacementWord().

Decide whether or not to put a filler word into this sentence (30% chance by default).

If yes, randomly pick a position for the filler word.

Decide whether or not to repeat one word in this sentence (10% chance by default).

If yes, randomly pick a word to be repeated.

Go through each word in the tagged sentence and change it to the colloquial variant if needed.

Put all the words together to form the output sentence while adding the filler word or repeating a word.

Repeat until the whole input has been processed.

The tags outputted by MorphoDiTa:

The tagger tags it's input with a so called "positional tag". A positional tag is a string of 15 characters. Every position encodes one morphological category using one character (mostly upper case letters or numbers).

Take the following example:

Dokumentuji VB-S---1P-AA—1

The word "dokumentuji" is has a tag that says it's a verb, is in singular form, has case number 1 etc.

For more info about positional tags, visit:
https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/html/ch02s02s01.html

Dictionary files:

Filler words

All filler words are stored in a file named filler-words.xml, where they're divided into 3 categories by sort of xml tags.

<beginning>

<beginning/>

<middle>

<middle/>

<end>

<end/>

One line can contain only one word. The category "end" is empty for now.

Replacement words

Replacement words are stored in a file named replacement-words.dict. Each line contains one pair of words, the word and it's replacement, separated by a tab character. For example:

word1 replacement1

word2 replacement2

Verbs should be stored as a lemma as represented in MorphoDiTa. So for example, if I want to store the verb "nastavit" and it's replacement "naštelovat", the line containing these will look like this:

nastavit_:W naštelovat_:W_,l

These are the lemma that MorphoDiTa gives when tagging these verbs.

More details on classes and methods:

Class Constants:

- Constants for default locations of various files needed such as the tagger model, the morphological dictionary, etc.
- Constants for default chances of repeating a word in a sentence or adding a filler word to a sentence.
- Constants for positions of morphological categories in the positional tag used by MorphoDiTa.

Class TextConverter:

- The main logic of the program is in this class.
- To Convert text, you have to create a TextConverter instance with the appropriate paths to the tagger model, morphological dictionary, replacement words file and filler words file. If none are given, the default path will be used which is "./files-to-load/". Once created, you must call the method Initialize() to load the given files. Then you can start converting by calling the method Convert().
- The chances of adding a filler word or repeating a word can be changed by calling the methods SetFillerChance() or SetRepetitionChance(). If not changed, the default is 30% for filler words and 10% for repeating a random word.
- The filler words are divided into 3 categories: beginning, middle and end. Some filler words are used only at the beginning of a sentence, some only at the end, some only in the middle of a sentence and some could be anywhere. Words only used at the beginning are read from the filler words file and stored in the list called fillerWordsBeginning. The same is true for fillerWordsMiddle and fillerWordsEnd.

Method ChangeToColloquialVariant()

- This method will return the given word in it's colloquial variant if there is one. Otherwise, it will return the original given word.
- The method uses the MorphoDiTa's morphological generator to find the colloquial form.
- The last position in the positional tag is used for variant/style of a word. This method modifies the last position of the tag of the input word, then tries to generate the word in colloquial form.

Method ChangeNounToColloquial()

- This method works in the same way as ChangeToColloquialVariant(). It will change the given noun to a colloquial variant if possible. For now it will only work for nouns in plural form and with case number 7. Maybe it could be expanded to work for more cases.
- For example the word “dokumentacemi” will be changed to “dokumentacema”.

Method GetReplacementWord()

- If the given word has a replacement in the replacement dictionary, this method will return that replacement.
- If the said replacement is a verb, this method will first try to inflect that verb using MorphoDiTa. If it succeeds, the correct form will be returned. Otherwise, it will return the unmodified original word.
- If the replacement is not a verb, this method will return the replacement word as listed in the dictionary.