Project 2 : Quality text prediction

As seen during class, a lot of systems exists to produce automatic summarization. Lately, deep learning was involved to this task in order to improve results. This new systems generates abstract summaries, which means that new unseen sentences in source documents can be generated. In this case a problem occurs : how to evaluate the quality of generated sentences. These summaries are human readable, are they grammatically correct ?

The goal of this project is to predict the level of grammaticality of pos-tagged summaries.

The Files

The available datas are composed of 3 files (available on a corpus zip file on brightspace - project2).

Differents systems are evaluated on several topics. A topic is a set of journal articles on a specific subject. (you don't have the source documents, you only have the result summaries of different systems on several topics).

« *input_learning_gramm_ST_2006* » and « *input_learning_gramm_ST_2007* » follows the same format :
Each summary is split by sentences. One sentence by line.
A line example :
D0701 10 5 NNP VBZ RB JJ .

Column 1 : « D0701 ». It's an id of topic or subject.
Column 2 : « 10 ». It's an id of system. (An id with a letter is a human generated summary=perfect).
Column 3 : « 5 ». It'a grammatical grade for a summary (between 1 and 5 included).
The rest until end of line : « NNP VBZ RB JJ . » It's a sentence of a summary pos-tagged with punctuation symbols.

Your model don't had to exploit first and second column. Their are only given to explain how data are generated and how properly load these datas.

Since files *input_learning_gramm_ST_2006* and *input_learning_gramm_ST_2007* shares same structure, you can merge these two files. But be aware to keep a set of training examples and a set of testing examples (ideally to avoid over-tuning under testing examples, you can also keep a set of validation examples).

An another file « CNN1.txt » is given. Each line of this file is a sentence pos-tagged.
This corpus represent a dataset of perfect pos-tagged sentences. We assume that is equivalent to a 5/5 grade on grammaticality.

The Model

You must build a system capable to predict the level of grammaticality for a set of sentences.
Here, we describe a main solution to achieve this goal (but you are free to propose a diferent model, if it's better). We can decompose the problem in two steps :

first step : learning a model language only based on perfect CNN pos-tagged sentences.

You can adapt the model from Keras documentation to generate text from Nietzsche's writings : https://github.com/keras-team/keras/blob/master/examples/lstm_text_generation.py

In this example the goal is to predict the next item (a character, for Nietzsche's example, a pos-tag for us).

Second step : once you had a model capable from a sequence of pos-tags to predict (for example the next) pos-tags, now we can use it to estimate the grammaticality of summaries.

Since the language model is learnt on perfect sentences we made the following hypothesis :
If I present a pos-tagged text grammatically correct to my model, we suppose that my language model will predict fine a majority of pos-tags.
But if I present a pos-tagged text grammatically incorrect to my model, we suppose that my language model will made more mistakes because this « bad » example doesn't follow the same distribution of all positives examples that my language model as already seen.

We need to transform the amount of error of your language model in a grammatical grade prediction.
For example, if my model made 0 mistakes on a text then a predict the maximal grade 5/5.
But if my model made some mistakes on a text, what grade a predict ?
Here we can learn a second supervised model that has in input the error (or error statistics) and in output a grade. You can use the files : *input_learning_gramm_ST_2006* » and « *input_learning_gramm_ST_2007 »* for this step.

Warning : In this two files, the grade in front of a sentence is not the grade for a sentence, but the grade for the entire summary (a set of sentences).

Tips : for the language model, predicting the next item is maybe not the only way or the best way.
Predicting an item in the middle of the input may work better…

You don't have to follow the steps discribe above. You can for example try to predict directly given a set of pos-tagged sentences a grade. It's to possible to do that using only files *input_learning_gramm_ST_2006 and input_learning_gramm_ST_2007*. But the problem doing that is the lake of examples in this files. This way lead to bad results.

To evaluate the quality of your entire model, you can use the Spearman correlation between your predicted grades and the true grades.

This work can been done by pairs of students.
The deadline is fixed to 18 january 2019 at midnight (and 25 january for apprentis).
You will make a report in english explaining your choices (in pdf format). If you try differents models or solutions you must explain it. You will send also your code.
In order to make a global automatic evaluation, I will send you a file with the same format of *input_learning_gramm_ST_2006* but without grade column.
You will send me back a file with only a grade by summary by line.
The name of the file : « nameStudent1_nameStudent2.txt »
For example, if I send you a file with 3 summaries you will return a file with the following content :

1
5
2

The non graded summaries will be available on brightspace the 26 january morning. Then you will upload your file with predicted grades the same day.

Annexe : meaning of pos-tags

At https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf
You can find the meaning of different pos-tags used by the the Penn Treebank.

Good luck and Happy New Year.