Université de Montréal

IFT6285 (TALN) — Project 2
Order your words

Contact :
Philippe Langlais
RALI/DIRO
Université de Montréal
+1 514 343 61 11 ext: 47494
felipe@iro.umontreal.ca
http://www.iro.umontreal.ca/~felipe/

last compilation : 15 novembre 2021 (21:49)

Context

In this project, you will study the problem of reordering a bag of words. For example, as a speaker of English, you have no difficulty producing the sentence *why everything does have to become such a big issue ?* for the bag of words *[? everything big why to become does have such issue a]*.

Your goal is to study automatic approaches to this problem. No particular constraints are imposed here, except that you can only use the training data that are detailed below. You can therefore not consider an approach consisting in querying an online search engine (which would not be very ecological anyway). On the other hand, it is quite simple to use the language models that you have deployed using `kenlm` in the devoir 2.

Data

To develop your approach, you should use only the 99 slices of the 1B-word corpus (see devoir 1), a copy of which is available at DIRO :

```
/u/demorali/corpora/1g-word-lm-benchmark-r13output/
   training-monolingual.tokenized.shuffled/
```

Three development corpus (projet2-dev.tar.gz) allow you to study the behavior of your solution :

news contains 1000 sentences (`news.ref`) and their associated bag of words (`news.test`). Those sentences are of the same distribution as the training material,

hans contains 1000 sentences (`hans.ref`) and their associated bag of words (`hans.test`). Those sentences are extracte from the so called Canadian Hansards,

euro contains 1000 sentences (`euro.ref`) and their associated bag of words (`euro.test`). Those sentences are from the Europarl corpus.

The development data has been prepared with the program `pre-process.py` as [1] (where `<fichier>` is a text file containing one sentence per line) :

1. Csh syntaxe.

```
cat <fichier> | python3 pre-process.py --min=5 --max=25
                                       --no='" - -- # www http'
                                       --nb=1000 --out=<fichier>.ref
                                       --lower >! <fichier>.test
```

This program uses tokenisation produced by model `en_core_web_sm` of spacy. It removes too short and too long sentences, as well as those containing words space-separated in option `--no`. Sentences (a total of 1000) are further lowercased.

Short before the deadline, test corpora will be released on which you will apply your best solution. The very same preparation will be applied.

Todo

Write a report (pdf format) of at most 6 pages which relates your experiments. In particular, you should :

1. Describe evaluation metrics used to evaluate your approaches,
2. Produce de solution which answers best the problem. You will take care of analyzing the performance of your solution, comparing it to fair baselines. You will also analyze things such as computation time, etc.,
3. Identify your best approach. Any none trivial solution involves meta-parameters [2] and you will specify yours,
4. Analyze the limitations of your approach,
5. Shortly before the deadline, test material will be distributed. You will apply your best approach on it. For each test file, fic, you must produce a file fic-names which contains the same number of lines and where `names` is the name of the persons participating in the project. [3]

Your report should look like a mini-scientific article and should before all demonstrate your curiosity. A few lines of python are enough to achieve with tools like `kenlm` a "decent" approach. You can also look at the transformer

2. Thresholds, architectural choices, data selection, algorithms. etc.

3. Links on files are just there for illustrating the expected format, and not for showing good results : here, a left-to-right greedy search has been produced thanks to a trigram language model trained with kenlm on the first million training sentences.

platform which offers ready-to-use code to query pre-trained *transformers* models. In particular, I recommend reading the `Causal Language Modeling` section of the overview. The point is that none of those solutions are perfect and your task is to improve a solution you designed (with or without the help of such tools).

Rating

The rating is not correlated to the performance of your approaches, but to the **curiosity** that you will develop and to your **analysis**. The clarity and informativeness of your report are two important criteria.

Here are some of the questions we will be asking during the grading process :

— are the *baseline* models reasonable ?
— are the efforts to exceed the performance of the baseline model justified, well described, consistent ?
— is the report clear, are the analyses informative ?

If you are using code written by others, you should mention it in your report.

Remise

The submission is to be made on Studium under the name `project2`. You have to submit your code, your report (pdf format, text in English or French), and the processed test files in an archive (gzip, tar, tar.gz) whose name is prefixed with `project2-names`, where `names` is to be replaced by the identity of the people (`firstname_lastname`) involved in the project. So if I had to submit alone, I would do it under the name `project2-philippe_langais.tar.gz`. Make sure that the names of the people involved in the project are indicated on the report. The project is due in groups of up to three people by Saturday, December 18 at 11 :59pm.