

Demo 3: Part-of-Speech Tagging part 3: Deep Learning for NLU

Sanjeet Maisnam (sanjeet.maisnam@epita.fr)

I. Creating data

Exercise 1:

1. Each of the train and the test datasets has one word per line. They are preprocessed to create the corresponding bitexts which have one complete sentence per line. The bitext file pairs have the property that every sentence pairs (one each from .src and .tgt) have the same number of words and each word in a sentence corresponds to the word of the same position in the other sentence pair.

The train and the test datasets are pre-processed using the following commands:

```
[ ] %%!
cat /content/train.words.txt | perl -pe 's/^$/<s>/' | perl -pe 's/\n/ /g' |
perl -pe 's/<s>/\n/g' | perl -pe 's/^[\ ]+//g' | perl -pe 's/[ ]+$//g' | head

⇒ ["Confidence in the pound is widely expected to take another sharp dive if trade figures for September , due for release
"Chancellor of the Exchequer Nigel Lawson 's restated commitment to a firm monetary policy has helped to prevent a freefall
"But analysts reckon underlying support for sterling has been eroded by the chancellor 's failure to announce any new policy
'This has increased the risk of the government being forced to increase base rates to 16 % from their current 15 % level
"`` The risks for sterling of a bad trade figure are very heavily on the down side , '' said Chris Dillow , senior U.K.
"`` If there is another bad trade number , there could be an awful lot of pressure , '' noted Simon Briscoe , U.K. economist
'Forecasts for the trade figures range widely , but few economists expect the data to show a very marked improvement from
'The August deficit and the # 2.2 billion gap registered in July are topped only by the # 2.3 billion deficit of October
"Sanjay Joshi , European economist at Baring Brothers & Co. , said there is no sign that Britain 's manufacturing industry
'At the same time , he remains fairly pessimistic about the outlook for imports , given continued high consumer and capital

[ ] %%!
cat /content/test.words.txt | perl -pe 's/^$/<s>/' | perl -pe 's/\n/ /g' |
perl -pe 's/<s>/\n/g' | perl -pe 's/^[\ ]+//g' | perl -pe 's/[ ]+$//g' | head

⇒ ["Rockwell International Corp. 's Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co.
'Rockwell said the agreement calls for it to supply 200 additional so-called shipsets for the planes .',
"These include , among other parts , each jetliner 's two major bulkheads , a pressure floor , torque box , fixed leadin
'Under the existing contract , Rockwell said , it has already delivered 793 of the shipsets to Boeing .",
'Rockwell , based in El Segundo , Calif. , is an aerospace , electronics , automotive and graphics concern .",
"Frank Carlucci III was named to this telecommunications company 's board , filling the vacancy created by the death of
'Mr. Carlucci , 59 years old , served as defense secretary in the Reagan administration .",
'In January , he accepted the position of vice chairman of Carlyle Group , a merchant banking concern .",
'SHEARSON LEHMAN HUTTON Inc .',
'Thomas E. Meador , 42 years old , was named president and chief operating officer of Balcor Co. , a Skokie , Ill. , sub
```

The preprocessed files are stored in files of corresponding formats which would be used later for training and evaluation.

```
⇒

[68] %%!
cat /content/train.words.txt | perl -pe 's/^$/<s>/' | perl -pe 's/\n/ /g' |
perl -pe 's/<s>/\n/g' | perl -pe 's/^[\ ]+//g' | perl -pe 's/[ ]+$//g' > /content/train.src

⇒ []

[69] %%!
cat /content/train.tags.txt | perl -pe 's/^$/<s>/' | perl -pe 's/\n/ /g' |
perl -pe 's/<s>/\n/g' | perl -pe 's/^[\ ]+//g' | perl -pe 's/[ ]+$//g' > /content/train.tgt

⇒ []

[70] %%!
cat /content/test.words.txt | perl -pe 's/^$/<s>/' | perl -pe 's/\n/ /g' |
perl -pe 's/<s>/\n/g' | perl -pe 's/^[\ ]+//g' | perl -pe 's/[ ]+$//g' > /content/test.src

⇒ []

[71] %%!
cat /content/test.tags.txt | perl -pe 's/^$/<s>/' | perl -pe 's/\n/ /g' |
perl -pe 's/<s>/\n/g' | perl -pe 's/^[\ ]+//g' | perl -pe 's/[ ]+$//g' > /content/test.tgt

⇒ []
```

Thus, the bitexts files are created for each of the training and test sets.

```
[73] !head /content/test.src /content/test.tgt

[<-- ==> /content/test.src <=]
Rockwell International Corp. 's Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co. to Rockwell said the agreement calls for it to supply 200 additional so-called shipsets for the planes . These include , among other parts , each jetliner 's two major bulkheads , a pressure floor , torque box , fixed leading Under the existing contract , Rockwell said , it has already delivered 793 of the shipsets to Boeing . Rockwell , based in El Segundo , Calif. , is an aerospace , electronics , automotive and graphics concern . Frank Carlucci III was named to this telecommunications company 's board , filling the vacancy created by the death of Mr. Carlucci , 59 years old , served as defense secretary in the Reagan administration . In January , he accepted the position of vice chairman of Carlyle Group , a merchant banking concern . SHEARSON LEHMAN HUTTON Inc . Thomas E. Meador , 42 years old , was named president and chief operating officer of Balcor Co. , a Skokie , Ill. , subsi

==> /content/test.tgt <=
NNP NNP NNP POS NNP NN VBD PRP VBD DT JJ NN VBG PRP$ NN IN NNP NNP TO VB JJ NNS IN NNP POS CD NNS PUNCT
NNP VBD DT NN VBZ IN PRP TO VB CD JJ JJ NNS IN DT NNS PUNCT
DT VBP PUNCT IN JJ NNS PUNCT DT NN POS CD JJ NNS PUNCT DT NN NN PUNCT NN NN PUNCT VBN VBG NNS IN DT NNS CC DT JJ NN NN PU
IN DT VBG NN NNP PUNCT NNP VBD PUNCT PRP VBZ RB VBN CD IN DT NNS TO NNP PUNCT
NNP PUNCT VBN IN NNP NNP PUNCT NNP PUNCT VBZ DT NN PUNCT NNS PUNCT JJ CC NNS VBP PUNCT
NNP NNP VBD VBN TO DT NNS NN POS NN PUNCT VBG DT NN VBN IN DT NN IN NNP NNP JJ NNP PUNCT
NNP NNP PUNCT CD NNS JJ PUNCT VBN IN NN NN IN DT NNP NN PUNCT
IN NNP PUNCT PRP VBD DT NN IN NN NN IN NNP NNP PUNCT DT NN NN NN PUNCT
NNP NNP NNP NNP PUNCT
NNP NNP NNP PUNCT CD NNS JJ PUNCT VBD VBN NN CC JJ VBG NN IN NNP NNP PUNCT DT NNP PUNCT NNP PUNCT NN IN DT NNP NNP NN NN
```

```
[72] !head /content/train.src /content/train.tgt

[<-- ==> /content/train.src <=]
Confidence in the pound is widely expected to take another sharp dive if trade figures for September , due for release to Chancellor of the Exchequer Nigel Lawson 's restated commitment to a firm monetary policy has helped to prevent a freefall But analysts reckon underlying support for sterling has been eroded by the chancellor 's failure to announce any new poli This has increased the risk of the government being forced to increase base rates to 16 % from their current 15 % level t `` The risks for sterling of a bad trade figure are very heavily on the down side , '' said Chris Dillow , senior U.K. ec `` If there is another bad trade number , there could be an awful lot of pressure , '' noted Simon Briscoe , U.K. economi Forecasts for the trade figures range widely , but few economists expect the data to show a very marked improvement from The August deficit and the # 2.2 billion gap registered in July are topped only by the # 2.3 billion deficit of October 1 Sanjay Joshi , European economist at Baring Brothers & Co. , said there is no sign that Britain 's manufacturing industry At the same time , he remains fairly pessimistic about the outlook for imports , given continued high consumer and capita

==> /content/train.tgt <=
NN IN DT NN VBZ RB VBN TO VB DT JJ NN IN NN NNS IN NNP PUNCT JJ IN NN NN PUNCT VB TO VB DT JJ NN IN NNP CC NNP POS JJ NNS
NNP IN DT NNP NNP POS VBN NN TO DT NN JJ NN VBZ VBN TO VB DT NN IN NN IN DT JJ NN PUNCT
CC NNS VBP VBG NN IN NN VBZ VBN VBN IN DT NN POS NN TO VB DT JJ NN NNS IN PRP$ NNP NNP JJ JJ NNP PUNCT
DT VBZ VBN DT NN IN DT NN VBG VBN TO VB NN NNS TO CD NN IN PRP$ JJ CD NN NN TO VB DT NN PUNCT NNS CC JJ NN NN NNS VBP PUN
PUNCT DT NNS IN NN IN DT JJ NN NN VBP RB RB IN DT JJ NN PUNCT PUNCT VBD NNP NNP PUNCT JJ NNP NN IN NNP NNP NNP PUNCT
PUNCT IN EX VBZ DT JJ NN NN PUNCT EX MD VB DT JJ NN IN NN PUNCT PUNCT VBD NNP NNP PUNCT NNP NN IN NNP NNP PUNCT DT NN IN
NNS IN DT NN NNS VBP RB PUNCT CC JJ NNS VBP DT NNS TO VB DT RB VBN NN IN DT # CD CD ( $ CD CD ) NN IN DT JJ NN VBD IN NNP
DT NNP NN CC DT # CD CD NN VBN IN NNP VBP VBN RE IN DT # CD CD NN IN NNP CD PUNCT
NNP NNP PUNCT JJ NN IN NNP NNPS CC NNP PUNCT VBD EX VBZ DT NN IN NNP POS NN NN VBZ VBG PRP TO VB NNS PUNCT
IN DT JJ NN PUNCT PRP VBZ RB JJ IN DT NN IN NNS PUNCT VBN VBD JJ NN CC NN NNS NNS PUNCT
```

The valid set can be extracted from the train set using the head and tail bash commands. We first checked the number of lines for the train set.

```
[75] !wc /content/train.src /content/train.tgt /content/test.src /content/test.tgt

[<-- ==> /content/train.src <=]
8936 211727 1156502 /content/train.src
8936 211727 773102 /content/train.tgt
2012 47377 261818 /content/test.src
2012 47377 173169 /content/test.tgt
21896 518208 2364591 total
```

Then, the last 1000 lines is extracted and the valid set is created.

```
[ ] %%!
tail -n 1000 /content/train.src > /content/val.src
tail -n 1000 /content/train.tgt > /content/val.tgt
```

The train set is also structured to remove the last 1000 lines.

```
[ ] %%!
tail -n 1000 /content/train.src > /content/val.src
tail -n 1000 /content/train.tgt > /content/val.tgt

▷ []

[82] %%!
mv /content/train.src /content/train_copy.src
mv /content/train.tgt /content/train_copy.tgt

head -n 7936 /content/train_copy.src > /content/train.src
head -n 7936 /content/train_copy.tgt > /content/train.tgt

▷ []

[85] %%!
wc /content/train.src
wc /content/train.tgt

▷ [' 7936 188059 1027529 /content/train.src',
    ' 7936 188059 686779 /content/train.tgt']
```

2. The three vocabularies are extracted from the training data as follows:

```
[92] !onmt-build-vocab --save_vocab /content/vocab.src /content/train.src
[93] !onmt-build-vocab --save_vocab /content/vocab.tgt /content/train.tgt
▶ %%!
onmt-build-vocab --tokenizer_config /content/char_tokenization.yml --save_vocab /content/vocab_chars.src /content/train.
```

The vocab files can be seen as follows:

```
[97] !head -n 5 /content/vocab.src /content/vocab.tgt /content/vocab_chars.src

▷ ==> /content/vocab.src <==
<blank>
<s>
</s>
'
the

==> /content/vocab.tgt <==
<blank>
<s>
</s>
NN
PUNCT

==> /content/vocab_chars.src <==
<blank>
<s>
</s>
-
e
```

3. Details of the train and test data can be observed through the following commands:

```
[75] !wc /content/train.src /content/train.tgt /content/test.src /content/test.tgt

▷      8936 211727 1156502 /content/train.src
      8936 211727 773102 /content/train.tgt
      2012 47377 261818 /content/test.src
      2012 47377 173169 /content/test.tgt
     21896 518208 2364591 total
```

II. Training the DL model

Exercise 1:

1. Before training, we need to set up the configuration for the model as follows:

```
s# WARNING:  
# Do not use this file as is! It only lists and documents available options  
# without caring about their consistency. Prefer using "opennmt-defaults.yml" as a  
# base configuration and specialize it with additional configuration files.  
# The directory where models and summaries will be saved. It is created if it does not exist.  
model_dir: /content #workspace/model_to_train/  
  
data:  
  train_features_file: /content/train.src  
  train_labels_file: /content/train.tgt  
  eval_features_file: /content/test.src  
  eval_labels_file: /content/test.tgt  
  
  # (optional) Models may require additional resource files (e.g. vocabularies).  
  source_1_vocabulary: /content/vocab.src  
  source_2_vocabulary: /content/vocab.chars.src  
  target_vocabulary: /content/vocab.tgt  
  # word_embeddings: /workspace/model_to_train/data/embeddings #optional you have to modify the the model file to take account of the word embeddings
```

As displayed in the config file, the model is stored in the “/content” directory. The train, test and the corresponding vocabs datasets are stored inside this “/content” directory.

The training can be performed as follows using the following command line:

```
!onmt-main --model /content/seq_tagger_updated.py --config /content/config_sample.yml train --with_eval  
...  
2020-06-14 17:47:38.236083: E tensorflow/stream_executor/cuda/cuda_driver.cc:351] failed call to cuInit: CUDA_ERROR_NO_DE  
2020-06-14 17:47:38.236122: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to b  
2020-06-14 17:47:38.236521: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this  
2020-06-14 17:47:38.243503: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 2200000000 Hz  
2020-06-14 17:47:38.243750: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x25a2a00 initialized for platf  
2020-06-14 17:47:38.243782: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default  
INFO:tensorflow:Using parameters:  
data:  
  eval_features_file: /content/test.src  
  eval_labels_file: /content/test.tgt  
  source_1_vocabulary: /content/vocab.src  
  source_2_vocabulary: /content/vocab.chars.src  
  target_vocabulary: /content/vocab.tgt  
  train_features_file: /content/train.src  
  train_labels_file: /content/train.tgt  
eval:  
  batch_size: 32  
  steps: 1000  
infer:  
  batch_size: 10  
  buffer_size: 100  
  length_bucket_width: null  
  n_best: 1  
  num_parallel_process_calls: 4  
model_dir: /content  
params:  
  average_loss_in_time: false  
  beam_width: 5  
  clip_gradients: 5.0  
  decay_params:  
    model_dim: 512  
    warmup_steps: 4000  
  decay_rate: 0.7  
  decay_steps: 500  
  decay_type: NoamDecay  
  eval_delay: 120
```

The details of the model including the parameters, optimization and batch size can be obtained as a byproduct of the above command.

```
model_dir: /content
params:
...
    average_loss_in_time: false
    beam_width: 5
    clip_gradients: 5.0
    decay_params:
        model_dim: 512
        warmup_steps: 4000
    decay_rate: 0.7
    decay_steps: 500
    decay_type: NoamDecay
    eval_delay: 120
    learning_rate: 0.01
    length_penalty: 0.2
    maximum_iterations: 200
    minimum_learning_rate: 1.0e-05
    num_hypotheses: 1
    optimizer: Adam
    scheduled_sampling_k: 0
    scheduled_sampling_read_probability: 1
    scheduled_sampling_type: constant
    staircase: true
    start_decay_steps: 1000
score:
    batch_size: 64
train:
    batch_size: 512
    batch_type: examples
    buffer_size: 10000
    eval_delay: 100
    keep_checkpoint_max: 10
    length_bucket_width: 1
    max_steps: 10000
    maximum_features_length: 250
    maximum_labels_length: 250
    num_buckets: 5
    num_parallel_process_calls: 4
    sample_buffer_size: 500000
    save_checkpoints_steps: 1000
    save_eval_predictions: false
    save_summary_steps: 100
```

The different steps of training are recorded as follows:

First few 100 steps:

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/ops/resource_variable_ops.py:1786:
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
INFO:tensorflow:Number of model parameters: 2489172
INFO:tensorflow:Number of model weights: 13 (trainable = 13, non trainable = 0)
INFO:tensorflow:Saved checkpoint /content/ckpt-1
INFO:tensorflow:Step = 100 ; steps/s = 0.82, source words/s = 2292, target words/s = 2292 ; Learning rate = 0.000010 ; Lo
INFO:tensorflow:Step = 200 ; steps/s = 1.00, source words/s = 2802, target words/s = 2802 ; Learning rate = 0.000010 ; Lo
INFO:tensorflow:Step = 300 ; steps/s = 1.26, source words/s = 2458, target words/s = 2458 ; Learning rate = 0.000010 ; Lo
```

After 13000 steps:

```
INFO:tensorflow:Running evaluation for step 13000
INFO:tensorflow:Evaluation result for step 13000: loss = 33.013252 ; perplexity = 217507025649664.000000 ; accuracy = 0.6
INFO:tensorflow:Step = 13100 ; steps/s = 0.89, source words/s = 2814, target words/s = 2814 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13200 ; steps/s = 1.16, source words/s = 2856, target words/s = 2856 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13300 ; steps/s = 1.33, source words/s = 2757, target words/s = 2757 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13400 ; steps/s = 0.95, source words/s = 3162, target words/s = 3162 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13500 ; steps/s = 1.24, source words/s = 2642, target words/s = 2642 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13600 ; steps/s = 1.20, source words/s = 2965, target words/s = 2965 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13700 ; steps/s = 1.00, source words/s = 3085, target words/s = 3085 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13800 ; steps/s = 1.29, source words/s = 2567, target words/s = 2567 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 13900 ; steps/s = 1.05, source words/s = 3131, target words/s = 3131 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 14000 ; steps/s = 1.11, source words/s = 2918, target words/s = 2918 ; Learning rate = 0.000010 ;
INFO:tensorflow:Saved checkpoint /content/ckpt-14000
INFO:tensorflow:Running evaluation for step 14000
```

After 29000 steps:

```
INFO:tensorflow:Step = 28200 ; steps/s = 1.13, source words/s = 2784, target words/s = 2784 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28300 ; steps/s = 0.91, source words/s = 2797, target words/s = 2797 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28400 ; steps/s = 1.25, source words/s = 2488, target words/s = 2488 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28500 ; steps/s = 0.97, source words/s = 2898, target words/s = 2898 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28600 ; steps/s = 1.01, source words/s = 2657, target words/s = 2657 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28700 ; steps/s = 1.28, source words/s = 2549, target words/s = 2549 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28800 ; steps/s = 0.90, source words/s = 2970, target words/s = 2970 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 28900 ; steps/s = 1.11, source words/s = 2499, target words/s = 2499 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 29000 ; steps/s = 1.19, source words/s = 2722, target words/s = 2722 ; Learning rate = 0.000010 ;
INFO:tensorflow:Saved checkpoint /content/ckpt-29000
INFO:tensorflow:Running evaluation for step 29000
INFO:tensorflow:Evaluation result for step 29000: loss = 11.079386 ; perplexity = 64821.054688 ; accuracy = 0.880702
INFO:tensorflow:Step = 29100 ; steps/s = 0.80, source words/s = 2569, target words/s = 2569 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 29200 ; steps/s = 1.21, source words/s = 2450, target words/s = 2450 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 29300 ; steps/s = 1.04, source words/s = 2907, target words/s = 2907 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 29400 ; steps/s = 0.98, source words/s = 2760, target words/s = 2760 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 29500 ; steps/s = 1.29, source words/s = 2519, target words/s = 2519 ; Learning rate = 0.000010 ;
INFO:tensorflow:Step = 29600 ; steps/s = 0.92, source words/s = 2972, target words/s = 2972 ; Learning rate = 0.000010 ;
```

It is observed that as the number of iterations increases, the loss decreases, and the accuracy increases gradually. However, the perplexity of the model varies.

2. Training the model was an arduous task and took almost a day.

III. Evaluation

1. While training the model through the OPEN-NMT command provided in the model training section, the description of the model is generated automatically and stored in a python file titled “model_description.py.” The content of the python file can be seen as follows:

```
"""Defines a bidirectional LSTM-CNNs-CRF as described in https://arxiv.org/abs/1603.01354."""
import tensorflow as tf
import opennmt as onmt

def model():
    return onmt.models.SequenceTagger(
        inputter=onmt.inputters.MixedInputter([
            onmt.inputters.WordEmbedder(
                embedding_size=50,
                trainable=True),
            onmt.inputters.CharConvEmbedder(
                embedding_size=50,
                num_outputs=30,
                kernel_size=3,
                stride=1,
                dropout=0.5)],
        dropout=0.5),
        encoder=onmt.encoders.RNNEncoder(
            num_layers=1,
            num_units=400,
            bidirectional=True,
            dropout=0.5,
            residual_connections=False,
            cell_class=tf.keras.layers.LSTMCell),
        crf_decoding=True)
```

Using this model description and evaluation python file “eval.py,” the model is evaluated on the test-tags file through the following command.

```
In [51]: !python tp3_files/eval.py colab_files/test.tgt colab_files/model_description.py
```

TAG	Precision	Recall	F1
(:	100	100	100.0
):	100	100	100.0
\$:	100	100	100.0
#:	0.0	100	0.0
CC:	0.0	100	0.0
CD:	100	100	100.0
DT:	0.0	100	0.0
EX:	0.0	100	0.0
FW:	100	100	100.0
IN:	0.0	100	0.0
JJ:	0.0	100	0.0
JJR:	100	100	100.0
JJS:	0.0	100	0.0
MD:	0.0	100	0.0
NN:	100	100	100.0
NNP:	0.0	100	0.0
NNPS:	0.0	100	0.0
NNS:	100	100	100.0
PDT:	0.0	100	0.0
POS:	0.0	100	0.0
PRP:	100	100	100.0
PRP\$:	100	100	100.0
PUNCT:	100	100	100.0
RB:	100	100	100.0
RBR:	100	100	100.0
RBS:	100	100	100.0
RP:	0.0	100	0.0
TO:	100	100	100.0
UH:	100	100	100.0
VB:	0.0	100	0.0
VBD:	100	100	100.0
VBG:	100	100	100.0
VBN:	100	100	100.0
VBP:	100	100	100.0
VBZ:	100	100	100.0
WDT:	100	100	100.0
WP:	100	100	100.0
WP\$:	100	100	100.0
WRB:	100	100	100.0
All:	64.1	100.0	78.13

The evaluation on the valid set could also be observed as follows:

```
In [53]: !python tp3_files/eval.py colab_files/val.tgt colab_files/model_description.py
```

TAG	Precision	Recall	F1
(:	100	100	100.0
):	100	100	100.0
\$:	0.0	100	0.0
#:	0.0	100	0.0
CC:	0.0	100	0.0
CD:	100	100	100.0
DT:	0.0	100	0.0
EX:	100	100	100.0
FW:	100	100	100.0
IN:	0.0	100	0.0
JJ:	100	100	100.0
JJR:	100	100	100.0
JJS:	0.0	100	0.0
MD:	100	100	100.0
NN:	100	100	100.0
NNP:	0.0	100	0.0
NNPS:	0.0	100	0.0
NNS:	100	100	100.0
PDT:	100	100	100.0
POS:	0.0	100	0.0
PRP:	0.0	100	0.0
PRP\$:	100	100	100.0
PUNCT:	100	100	100.0
RB:	0.0	100	0.0
RBR:	100	100	100.0
RBS:	100	100	100.0
RP:	100	100	100.0
TO:	100	100	100.0
UH:	100	0.0	0.0
VB:	100	100	100.0
VBD:	0.0	100	0.0
VBG:	100	100	100.0
VBN:	100	100	100.0
VBP:	100	100	100.0
VBZ:	33.33	0.18	0.36
WDT:	100	100	100.0
WP:	0.0	100	0.0
WP\$:	100	100	100.0
WRB:	100	100	100.0
All:	64.96	94.88	77.12

2. The eval.py generates evaluation of the model through three performance measures: Precision, Recall and F1 score.

Let us first note the four parameters:

True Positive (TP): The actual correctly labelled tags predicted truly by the model
 True Negative (TN): The actual wrongly labelled tags predicted truly by the model

False Positive (TP): The actual correctly labelled tags predicted wrongly by the model. (Type-II error)

False Negative (TN): The actual wrongly labelled tags predicted truly by the model. (Type-I error)

Precision = $\frac{TP}{TP+FP}$. It is the ratio of the correctly predicted positive observations (TP) to the total predicted positive observations (TP+FP).

Recall = $\frac{TP}{TP+FN}$. It is the ratio of correctly predicted positive observations to the all observations in actual class (TP+FN), not the predicted class.

F1 score = $\frac{2*(Recall*Precision)}{(Recall+Precision)}$. It is the weighted average of recall and precision. Therefore, this score takes both false positives and false negatives into account.

Now, from the corresponding commands for the evaluation of the model, we observe that the model has

1. precision = 64.1, recall = 100 and F1 score = 78.13 on the test set.
2. precision = 64.96, recall = 94.88 and F1 score = 77.12 on the valid set.

Note that higher the F1 score, better is the prediction of the model. A recall value of higher than 50 is also a good measure. Meanwhile, high precision relates to the low false positive rate.