1_0_Lexical_Complexity_Dataset_Classification_Functional_Test_MVP_v

April 5, 2025

[1]:

```
#@title Install Packages
```

[2]:

```
!pip install -q transformers
!pip install -q torchinfo
!pip install -q datasets
!pip install -q evaluate
!pip install -q nltk
```

```
                                    491.2/491.2 kB
13.7 MB/s eta 0:00:00
                                    116.3/116.3 kB
11.6 MB/s eta 0:00:00
                                    183.9/183.9 kB
17.1 MB/s eta 0:00:00
                                    143.5/143.5 kB
14.6 MB/s eta 0:00:00
                                    194.8/194.8 kB
18.0 MB/s eta 0:00:00
```

```
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of the
following dependency conflicts.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is
incompatible.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cublas-cu12
12.5.3.2 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cuda-cupti-cu12
12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cuda-nvrtc-cu12
12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127; platform_system
== "Linux" and platform_machine == "x86_64", but you have nvidia-cuda-runtime-
cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cudnn-cu12
9.3.0.75 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cufft-cu12
11.2.3.61 which is incompatible.
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-curand-cu12
10.3.6.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cusolver-cu12
11.6.3.83 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cusparse-cu12
12.5.1.3 which is incompatible.
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-nvjitlink-cu12
12.5.82 which is incompatible.
                          84.0/84.0 kB
```

```
3.6 MB/s eta 0:00:00
```

[3]:

```
!sudo apt-get update
! sudo apt-get install tree
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,632 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease [1,581 B]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease [6,555 B]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:7 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ Packages [70.9
kB]
Get:8 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
Packages [1,381 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Hit:10 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:11 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:13 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,808 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,092
kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
[1,241 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[1,540 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages
[4,148 kB]
Get:18 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,688 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages
[2,775 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64
Packages [3,978 kB]
Fetched 30.1 MB in 4s (6,851 kB/s)
Reading package lists… Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 46 not upgraded.
```

```
Need to get 47.9 kB of archives.
After this operation, 116 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tree amd64 2.0.2-1
[47.9 kB]
Fetched 47.9 kB in 0s (158 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78,
<> line 1.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package tree.
(Reading database … 126210 files and directories currently installed.)
Preparing to unpack …/tree_2.0.2-1_amd64.deb …
Unpacking tree (2.0.2-1) …
Setting up tree (2.0.2-1) …
Processing triggers for man-db (2.10.2-1) …
```

[4]:

```
#@title Imports

import transformers
import evaluate

import nltk

from datasets import load_dataset
from torchinfo import summary

from transformers import AutoTokenizer, AutoModel,␣
 ↪AutoModelForSequenceClassification
from transformers import TrainingArguments, Trainer

import os
import pandas as pd
import numpy as np
```

[5]:

```
# @title Mount Google Drive
```

[6]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[7]:

```
dir_root = '/content/drive/MyDrive/266-final/'
# dir_data = '/content/drive/MyDrive/266-final/data/'
```

```
dir_data = '/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/'
dir_models = '/content/drive/MyDrive/266-final/models/'
dir_results = '/content/drive/MyDrive/266-final/results/'
```

[8]:
```
!tree -L 2 /content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/
```

```
/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/
  evaluate.py
  Readme.md
  test
     lcp_multi_test.tsv
     lcp_single_test.tsv
  test-labels
     lcp_multi_test.tsv
     lcp_single_test.tsv
  train
     lcp_multi_train.tsv
     lcp_single_train.tsv
  trial
      lcp_multi_trial.tsv
      lcp_single_trial.tsv

4 directories, 10 files
```

[9]:
```
# !tree -L 4 /content/drive/MyDrive/266-final/
```

[10]:
```
!ls -R /content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/
```

```
/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/:
evaluate.py  Readme.md  test  test-labels  train  trial

/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/test:
lcp_multi_test.tsv  lcp_single_test.tsv

/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/test-labels:
lcp_multi_test.tsv  lcp_single_test.tsv

/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/train:
lcp_multi_train.tsv  lcp_single_train.tsv

/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/trial:
lcp_multi_trial.tsv  lcp_single_trial.tsv
```

[11]:
```
#@title Import Data
```

[15]:
```
# # Construct file paths
# train_multi_path = os.path.join(dir_data, "train", "lcp_multi_train.tsv")
# train_single_path = os.path.join(dir_data, "train", "lcp_single_train.tsv")
# test_multi_path = os.path.join(dir_data, "test", "lcp_multi_test.tsv")
# test_single_path = os.path.join(dir_data, "test", "lcp_single_test.tsv")

# # Load them into pandas DataFrames
# df_train_multi = pd.read_csv(train_multi_path, sep="\t")
# df_train_single = pd.read_csv(train_single_path, sep="\t")
# df_test_multi = pd.read_csv(test_multi_path, sep="\t")
# df_test_single = pd.read_csv(test_single_path, sep="\t")
```

[16]:

```
# print("Train Multi:\n", df_train_multi.head())
# print("Train Multi:\n", df_train_multi.shape)
# print("Train Multi:\n", df_train_multi.info())
# print("")
# print("\nTrain Single:\n", df_train_single.head())
# print("\nTrain Single:\n", df_train_single.shape)
# print("\nTrain Single:\n", df_train_single.info())
# print("")
# print("\nTest Multi:\n", df_test_multi.head())
# print("\nTest Multi:\n", df_test_multi.shape)
# print("\nTest Multi:\n", df_test_multi.info())
# print("")
# print("\nTest Single:\n", df_test_single.head())
# print("\nTest Single:\n", df_test_single.shape)
# print("\nTest Single:\n", df_test_single.info())
```

[17]:

```
from datasets import load_dataset

# Suppose you just want to load train + test for your "single" dataset
data_files = {
    "train": os.path.join(dir_data, "train", "lcp_single_train.tsv"),
    "test":  os.path.join(dir_data, "test",  "lcp_single_test.tsv"),
}

dataset = load_dataset(
    "csv",  # we can still pass "csv" even though it's TSV
    data_files=data_files,
    delimiter="\t"  # crucial to handle TSV
)

# dataset is now a DatasetDict with 'train' and 'test' splits
print(dataset)

# Suppose each row has columns like `sentence`, `target_word`, `complexity`
# Next, define a tokenization function:
from transformers import AutoTokenizer

model_checkpoint = "bert-base-cased"  # for example
```

```python
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)

def tokenize_function(examples):
    # Usually you'd tokenize the 'sentence' field (and maybe 'target_word'?).
    return tokenizer(examples["sentence"], truncation=True)

# Apply tokenization to each row in the dataset
tokenized_dataset = dataset.map(tokenize_function, batched=True)

# If you want to train with a Trainer:
from transformers import DataCollatorWithPadding, AutoModelForSequenceClassification

model = AutoModelForSequenceClassification.from_pretrained(model_checkpoint, num_labels=1)
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)

# etc. (TrainingArguments, Trainer, and so on).
```

```
Generating train split: 0 examples [00:00, ? examples/s]

Generating test split: 0 examples [00:00, ? examples/s]

DatasetDict({
    train: Dataset({
        features: ['id', 'corpus', 'sentence', 'token', 'complexity'],
        num_rows: 7232
    })
    test: Dataset({
        features: ['id', 'corpus', 'sentence', 'token', 'complexity'],
        num_rows: 808
    })
})

tokenizer_config.json:   0%|          | 0.00/49.0 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/570 [00:00<?, ?B/s]

vocab.txt:   0%|          | 0.00/213k [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/436k [00:00<?, ?B/s]

Map:   0%|          | 0/7232 [00:00<?, ? examples/s]

Map:   0%|          | 0/808 [00:00<?, ? examples/s]

Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
```

```
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`

model.safetensors:   0%|          | 0.00/436M [00:00<?, ?B/s]

Some weights of BertForSequenceClassification were not initialized from the
model checkpoint at bert-base-cased and are newly initialized:
['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

[21]:

```
# import os
# import pandas as pd
# from transformers import AutoTokenizer

# train_single_path = os.path.join(dir_data, "train", "lcp_single_train.tsv")
# df_train_single = pd.read_csv(train_single_path, sep="\t")

# # Inspect or preprocess in Pandas if you like
# print(df_train_single.head())

# # Suppose the file has columns: "id", "sentence", "target_word", "complexity"
# train_sentences = df_train_single["sentence"].tolist()
# train_targets = df_train_single["complexity"].tolist()  # e.g., a float label

# # Now tokenize
# model_checkpoint = "bert-base-cased"
# tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)

# encoded_train = tokenizer(
#     train_sentences,
#     truncation=True,
#     padding=True,      # or "max_length", "longest", or later w/DataCollator
#     return_tensors="pt"
# )
# # encoded_train now has input_ids, attention_mask, etc.
```

[22]:

```
import os
import pandas as pd

# Assuming you've already defined:
# dir_data = '/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/'

train_dir = os.path.join(dir_data, 'train')
test_dir = os.path.join(dir_data, 'test')
```

[23]:

```
# # -- Single
# train_single_df = pd.read_csv(
#     os.path.join(train_dir, "lcp_single_train.tsv"),
#     sep="\t"
# )
# test_single_df = pd.read_csv(
#     os.path.join(test_dir, "lcp_single_test.tsv"),
#     sep="\t"
# )

# -- Multi
train_multi_df = pd.read_csv(
    os.path.join(train_dir, "lcp_multi_train.tsv"),
    sep="\t"
)
test_multi_df = pd.read_csv(
    os.path.join(test_dir, "lcp_multi_test.tsv"),
    sep="\t"
)
```

```
---------------------------------------------------------------------------
ParserError                               Traceback (most recent call last)
<ipython-input-23-d3ddc1bc1f15> in <cell line: 0>()
     14     sep="\t"
     15 )
---> 16 test_multi_df = pd.read_csv(

     17     os.path.join(test_dir, "lcp_multi_test.tsv"),
     18     sep="\t"

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
 ↪usecols, dtype, engine, converters, true_values, false_values,␣
 ↪skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na,␣
 ↪na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format,␣
 ↪keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator,␣
 ↪chunksize, compression, thousands, decimal, lineterminator, quotechar,␣
 ↪quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect ␣
 ↪on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision,␣
 ↪storage_options, dtype_backend)
   1024     kwds.update(kwds_defaults)
   1025
-> 1026     return _read(filepath_or_buffer, kwds)
   1027
   1028

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪_read(filepath_or_buffer, kwds)
    624
    625     with parser:
--> 626         return parser.read(nrows)
    627
    628

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪read(self, nrows)
   1921                     columns,
   1922                     col_dict,
-> 1923                 ) = self._engine.read(  # type: ignore[attr-defined]
   1924                     nrows
   1925                 )

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/c_parser_wrapper.py in␣
 ↪read(self, nrows)
    232         try:
    233             if self.low_memory:
--> 234                 chunks = self._reader.read_low_memory(nrows)
    235                 # destructive to chunks
    236                 data = _concatenate_chunks(chunks)

parsers.pyx in pandas._libs.parsers.TextReader.read_low_memory()

parsers.pyx in pandas._libs.parsers.TextReader._read_rows()

parsers.pyx in pandas._libs.parsers.TextReader._tokenize_rows()

parsers.pyx in pandas._libs.parsers.TextReader._check_tokenize_status()

parsers.pyx in pandas._libs.parsers.raise_parser_error()

ParserError: Error tokenizing data. C error: EOF inside string starting at row 40
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[19]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```