afrolid Documentation

Release latest

Jul 13, 2023

COMMAND LINE INTERFACES

1 Requirements and Installation 3

2 Interactive Command Line 5

3 Integrate AfroLID with python code 7

AfroLID, a neural LID toolkit for 517 African languages and varieties. AfroLID exploits a multi-domain web dataset manually curated from across 14 language families utilizing five orthographic systems.

github
https://github.com/UBC-NLP/afrolid

demo
https://demos.dlnlp.ai/afrolid

paper
https://arxiv.org/abs/2210.11744

CHAPTER

ONE

REQUIREMENTS AND INSTALLATION

1.1 Install using pip

To install AfroLID and develop directly using pip:

```
pip install afrolid
```

or

```
pip install -U git+https://github.com/UBC-NLP/afrolid.git
```

1.2 Install Locally

To install AfroLID and develop locally:

```
git clone https://github.com/UBC-NLP/afrolid.git
cd afrolid
pip install .
```

1.3 Download AfroLID model

CHAPTER

TWO

INTERACTIVE COMMAND LINE

- AfroLID interactive cli `afrolid_cli` support only beam search with the following default setting:
  - `-m` or `--model_path`: Rath of the AfroLID model directory, (`Required`)
  - `-o` or `--max_outputs`: The maximum of the output tanslations (`default value is 3`)
  - `-l` or `--logging_file`: Number of beams (`default value is 1`)
  - `-n` or `--no_repeat_ngram_size`: Number of n-gram that doesn't appears twice (`default value is 2`)
- `afrolid_cli` command asks you you to input your input text. Moreover, you can write q to exsit as shown in the following image.

2.1 Usage and Arguments

```
afrolid_cli -h
```

2.2 AfrlioLID Interactive

```
!afrolid_cli --model_path /path/to/model
```

```
2022-12-06 18:01:24 | INFO | afroli.afrolid_cli | AfroLID Command Line Interface
2022-12-06 18:01:24 | INFO | afroli.afrolid_cli | Initalizing AfroLID's task and model.
| [input] dictionary: 64001 types
| [label] dictionary: 528 types
Type your input text or (q) to STOP:          5
2022-12-06 18:01:41 | INFO | afroli.afrolid_cli | Input text:          5
Predicted languages:
      |-- ISO: tir  Name: Tigrinya  Script: Ethiopic        Score: 100.0%
Type your input text or (q) to STOP:             50
2022-12-06 18:01:57 | INFO | afroli.afrolid_cli | Input text:             50
Predicted languages:
      |-- ISO: amh  Name: Amharic   Script: Ethiopic        Score: 49.74%
      |-- ISO: tir  Name: Tigrinya  Script: Ethiopic        Score: 49.34%
      |-- ISO: gof  Name: Goofa     Script: Latin   Score: 0.82%
Type your input text or (q) to STOP:       -   :
2022-12-06 18:02:09 | INFO | afroli.afrolid_cli | Input text:       -   :
```

(continues on next page)

(continued from previous page)

```
Predicted languages:
      |-- ISO: rif  Name: Tarifit   Script: Arabic  Score: 100.0%
Type your input text or (q) to STOP: Vamteta vakulu na vagogo va vandu vamkotili
2022-12-06 18:02:18 | INFO | afroli.afrolid_cli | Input text: Vamteta vakulu na vagogo␣
↪va vandu vamkotili
Predicted languages:
      |-- ISO: ngo  Name: Ngoni     Script: Latin   Score: 99.95%
      |-- ISO: rwk  Name: Rwa       Script: Latin   Score: 0.01%
      |-- ISO: asa  Name: Asu       Script: Latin   Score: 0.01%
Type your input text or (q) to STOP: q
```

2.3 Google Colab Link

You can find the full examples on the Google Colab on the following link https://colab.research.google.com/github/UBC-NLP/afrolid/blob/main/examples/afrolid_interactive_cli.ipynb

CHAPTER

THREE

INTEGRATE AFROLID WITH PYTHON CODE

(1) Install AfroLID

```
pip install git+https://github.com/UBC-NLP/afrolid.git --q
```

3.1 Initial AfroLID object

Import related packges

```
import os, sys
import logging
from afrolid.main import classifier
```

```
logging.basicConfig(
      format="%(asctime)s | %(levelname)s | %(name)s | %(message)s",
      datefmt="%Y-%m-%d %H:%M:%S",
      level=os.environ.get("LOGLEVEL", "INFO").upper(),
      force=True, # Resets any previous configuration
)
logger = logging.getLogger("afrolid")
```

Create turjuman object

```
cl = classifier(logger, model_path=/path/to/model)
```

3.2 Get language prediction(s)

```
## Gold label = dip
 text="6Acï looi aya në wuöt dït kk yiic ku l wuöt tu tëmec piny de Manatha ku Eparaim ku Thimion , ku ään mec tu të l rut cï Naptali"
 predicted_langs = cl.classify(text) # default max_outputs=3
 print("Predicted languages:")
 for lang in predicted_langs:
 print("     |-- ISO: {}\tName: {}\tScript: {}\tScore: {}%".format(
                lang,
                predicted_langs[lang]['name'],
```

(continues on next page)

(continued from previous page)

```
                 predicted_langs[lang]['script'],
                 predicted_langs[lang]['score']))
```

3.3 Integrate with Pandas

```
wget https://raw.githubusercontent.com/UBC-NLP/afrolid/main/examples/examples.
↪tsv -O examples.tsv
```

```
import pandas as pd
from tqdm import tqdm
tqdm.pandas()
df = pd.read_csv("examples.tsv", sep="\t")

def get_afrolid_prediction(text):
      predictions = cl.classify(text, max_outputs=1)
      for lang in predictions:
            return lang, predictions[lang]['score'], predictions[lang]['name'],␣
↪predictions[lang]['script']

df['predict_iso'], df['predict_score'], df['predict_name'], df['predict_script'] =␣
↪zip(*df['content'].progress_apply(get_afrolid_prediction))
```

```
{'source': 'As US reaches one million COVID deaths, how are Americans coping?', 'target
↪': ['          -19      ']}
```

3.4 Read and translate text from file

- `-f` or `--input_file`: import the text from file. The translation will saved on the JSON format file
- `-bs` or `--batch_size`: The maximum number of source examples utilized in one iteration (`default value is 25`)
- `gen_options`: Generation options

```
gen_options = {"search_method":"beam", "seq_length": 300, "num_beams":5, "no_repeat_
↪ngram_size":2, "max_outputs":1}
torj.translate_from_file("samples.txt", batch_size=25, **gen_options)
```

3.5 Google Colab Link

You can find the full examples on the Google Colab on the following link https://colab.research.google.com/github/UBC-NLP/afrolid/blob/main/examples/Integrate_afrolid_with_your_code.ipynb