**Pykaldi**: A Python Wrapper for Kaldi

Doğan Can (dogancan@usc.edu), Victor R. Martinez, Pavlos Papadopoulos, and Shrikanth Narayanan (shri@sipi.usc.edu)

**About**
Pykaldi is a fast and modular scripting layer for Kaldi. It relies on CLIF to wrap Kaldi C++ libraries into CPython extensions with minimal overhead. CLIF wrappers are lovingly extended in Python to provide a “Pythonic” API. It was designed from the ground up to be a joy to use in Python. We hope you like it as much as we do!

**Alternatives**

- /esnet/esnet: End-to-End Speech Processing Toolkit
- /janchorowski/kaldi-python: Dedicated to Kaldi I/O
- /UFA-DGS/pykaldi: Only wraps OnlineLatticeRecognizer
- /bob/bob.kaldi: Features (MFCC, i-vector), VAD and UBM.

**Good Old Kaldi in Python**

```python
from kaldi.feature import Mfcc
MfccOptions()
```

**Architecture**

![Architecture Diagram](image)

**Features**
- Exposes nearly all of Kaldi C++ library API to Python code
- Perfect for quick experimentation, debugging, visualization
- Low overhead vs. Kaldi native C++ code
- Seamless integration with NumPy ndarrays
- First class support for Kaldi and Openfst objects
- Extensive API documentation – work in progress
- Open License: Apache 2.0

https://github.com/pykaldi/pykaldi
docker pull pykaldi/pykaldi
http://pykaldi.github.io/

**Cheap Conversion to/from NumPy**

```python
import numpy as np
from kaldi.matrix import Matrix, SubMatrix
```

**Modular ASR in Python**

```python
from kaldi.feature import Mfcc
MfccOptions()
```

**RNNLM Rescoring**

```python
from kaldi.cudamatrix import CuMatrix
```

**Good Old Kaldi in Python**

```python
In from kaldi.feature mfcc import Mfcc, MfccOptions
```

```python
mb = Mfcc(MfccOptions())
```

```python
with SequentialWaveReader("scp:wav.scp") as reader,
  MatrixWriter("ark:mfccark") as writer:
  for wav in reader:
    if wav.duration < 60:
      continue
    v = wav.data()[8]
    f = mfcc.compute_features(v, wav.samp_freq, 1.0)
    writer(key) = f
```

```python
# Convert to/from NumPy ndarray
m = Matrix([[1,2,3],[4,5,6],[7,8,9]])
```

```python
# No explicit conversion needed
v = SubVector(np.mean(m, axis=0))
```

```python
# Define the feature extraction pipeline
```

```python
# Define the decoder
```

```python
# Define the recognizer
```

```python
# Decode wav files
```

http://sail.usc.edu/

Work supported by DARPA and NSF