G2P - Aymara

Manual del Usuario

G2P - Aymara es una utilidad de software que es capaz de entrenar un modelo estadístico para la predicción de phonemas en base a su escritura en grafemas. Esta herramienta utiliza diccionarios fonéticos en formato ARPA para el entrenamiento y cuenta con utilidades para predecir sobre nuevos diccionarios y con un modo interactivo.

La aplicación fue desarrollada sobre Python 2.7.

Dependencias

G2P-Aymara cuenta con las siguientes dependencias de paquetes y librerías de python:

- pip
- h5py (2.6.0)
- Keras (1.2.0)
- numpy (1.11.3)
- pandas (0.19.2)
- recurrentshop (0.0.1)
- scipy (0.18.1)
- seq2seq (0.1.0)
- tensorflow (0.12.0rc1)
- Theano (0.8.2)

Instale las dependencias usando el administrador de paquetes `pip`:

```
pip install -U [nombre del paquete]
```

Instalación

El paquete no necesita un proceso especial de instalación. simplemente copie toda la carpeta a un directorio de trabajo de su preferencia.

Entrenamiento

La utilidad entrena un modelo estadístico en base a diccionarios fonéticos con formato ARPA. los formatos válidos para los diccionarios fonéticos son los formatos: `.dic` y `.dict`.

Navegue hasta el directorio de la aplicación, donde se encuentra el script llamado `trainTest.py` y abra el script en un editor de texto. Al principio del archivo, modifique las siguientes líneas de código:

```
# directorios necesarios para guardar y cargar el dataset y los modelos
```

```
dic_dir = "dic_datasets/aymara.dic"
model_dir ="aymaramodel/"
model_name = "aymara1"
```

Estas 3 variables definen los directorios de trabajo para el entrenamiento.

- `dic_dir` es el directorio en el cual se encuentra el diccionario fonético de entrenamiento en formato ARPA con extensión .dic o .dict
- `model_dir` es la carpeta que se creará para almacenar los parámetros del modelo y otra información del entrenamiento. Procure incluir el caracter `/`al final.
- `model_name` es el nombre del modelo a ser entrenado, este nombre define los nombres de los archivos de parámetros contenidos en *model_dir*

En el ejemplo anterior, el diccionario se encuentra contenido en el mismo directorio que el script `trainTest.py` en la carpeta `dic_datasets`y tiene el nombre `aymara.dic`. La carpeta con los parámetros del modelo se llama `aymaramodel/` y el modelo se llama `aymara1`.

Una vez modificados los directorios se procede al entrenamiento ejecutando el archivo desde una terminal desde el directorio donde se encuentra:

```
~$ python trainTest.py
```

El programa comenzará a entrenar un modelo usando el diccionario fonético y el directorio del modelo indicados en el script.

Al finalizar el entrenamiento, se imprimirá un mensaje con el rendimiento del modelo de la siguiente forma:

```
precision: 400/500 = 0.8000
```

Esto quiere decir que nuestro modelo predijo correctamente 400 palabras de 500 que compone el conjunto de prueba (*test set*), o, dicho de otro modo, con una precisión del 80%.

Hasta el momento, la mayor precisión alcanzada con un diccionario de aproximadamente 2800 palabras es de 73%.

Tiempo de entrenamiento

Dependiendo al tamaño del diccionario fonético (*dataset*), la complejidad del modelo y la plataforma de hardware sobre la que se trabaja, el entrenamiento tomará más tiempo en realizarse. Se recomienda usar tensorflow con capacidad GPU, ya que esto aminora dramáticamente los tiempos de entrenamiento.

Modificación de parámetros

Para datasets con registros de hasta 5000 muestras, se recomienda usar los valores por defecto:

- 1 capa con 128 celdas para el modelo.
- 600 iteraciones para el entrenamiento.

En este caso, no se debe realizar ningún cambio al script de entrenamiento.

En otros casos, puede ser necesario cambiar los parámetros del modelo y del entrenamiento para mejorar el rendimiento del mismo. Para tal cometido se debe modificar la línea que contiene la instrucción `g2pModel.prepareModel()` para cambiar el modelo y la instrucción `trained = g2pModel.trainModel()` para el entrenamiento. Considere los siguientes criterios:

- Si el diccionario de entrenamiento contiene más de 5000 palabras incremente el número de celdas: `g2pModel.prepareModel(cells=200)` Se crea un modelo con 200 celdas.
- Si el diccionario de entrenamiento contiene más de 20000 palabras incremente además el número de capas: `g2pModel.prepareModel(layers=3, cells=256)` Se crea un modelo con 3 capas de 256 celdas cada una.
- Si el diccionario de entrenamiento tiene más de 5000 palabras, pero el rendimiento no mejora, o empeora, incremente el número de iteraciones en el entrenamiento: `trained = g2pModel.trainModel(epoch=800)` Se incrementa el número de iteraciones a 800.

Pese a estas opciones, el rendimiento del sistema depende enromemente de calidad de los datos de entrenamiento definidos en el diccionario fonético, así que se debe procurar alimentar al modelo con los datos más idóneos y numerosos posibles.

Predicción

Si se cuenta con un modelo entrenado que esté en el formato correcto, ya sea que haya sido entrenado con la misma herramienta o de manera externa, la aplicación puede utilizar dicho modelo para realizar predicciones sobre nuevas palabras.

En tal caso, se necesitan los mismos directorios que para el entrenamiento: diccionario fonético y directorio del modelo.

Para ejecutar predicciones en modo interactivo se debe ejecutar el script `predictTest.py`.

Al principio del archivo, modifique las siguientes líneas de código:

```
# directorios necesarios para guardar y cargar el dataset y los modelos
dic_dir = "dic_datasets/aymara.dic"
model_dir ="aymaramodel/"
model_name = "aymara1"
```

Estas 3 variables definen los directorios de trabajo para el entrenamiento.

- `dic_dir` es el directorio en el cual se encuentra el diccionario fonético de entrenamiento en formato ARPA con extensión .dic o .dict

- `model_dir` es la carpeta en la que se encuentran los parámetros del modelo y otra información del entrenamiento. Procure incluir el caracter `/`al final.
- `model_name` es el nombre del modelo entrenado, este nombre define los nombres de los archivos de parámetros contenidos en *model_dir*

En el ejemplo anterior, el diccionario se encuentra contenido en el mismo directorio que el script `trainTest.py` en la carpeta `dic_datasets`y tiene el nombre `aymara.dic`. La carpeta con los parámetros del modelo se llama `aymaramodel/` y el modelo se llama `aymara1`.

Una vez modificados los directorios se procede a probar el modelo e ingresar a un modo interactivo ejecutando el archivo desde una terminal desde el directorio donde se encuentra:

```
~$ python predictTest.py
```