DeepSpeech by Mozilla

ESC 2K18

Hi, I’m Stefania Delprete, nice to meet you!

Data Scientist in TOP-IX, BIG DIVE co-organizer

stefania.delprete@top-ix.org // @astrastefania

Mozilla volunteer,

helping developing communities in Berlin and Turin/Italy

Why speech recognition? Why now?

- Project Vaani, voice assistant for Firefox OS
- Connected devices
- Need for a open source alternative
- End-to-end Machine Learning system

Studying available researches

Baidu’s Deep Speech paper v2

Further publications

Deep Speech v2 for English and Mandarin
https://arxiv.org/abs/1512.02595

Deep Speech v3
http://research.baidu.com/Blog/index-view?id=90
https://arxiv.org/abs/1707.07413

Deep Speech: Scaling up end-to-end speech recognition

Awni Hannun*, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng

Baidu Research – Silicon Valley AI Lab

Abstract

We present a state-of-the-art speech recognition system developed using end-to-end deep learning. Our architecture is significantly simpler than traditional speech systems, which rely on laboriously engineered processing pipelines; these traditional systems also tend to perform poorly when used in noisy environments. In contrast, our system does not need hand-designed components to model background noise, reverberation, or speaker variation, but instead directly learns a function that is robust to such effects. We do not need a phoneme dictionary, nor even the concept of a “phoneme.” Key to our approach is a well-optimized RNN training system that uses multiple GPUs, as well as a set of novel data synthesis techniques that allow us to efficiently obtain a large amount of varied data for training. Our system, called Deep Speech, outperforms previously published results on the widely studied Switchboard Hub5’00, achieving 16.0% error on the full test set. Deep Speech also handles challenging noisy environments better than widely used, state-of-the-art commercial speech systems.

1 Introduction

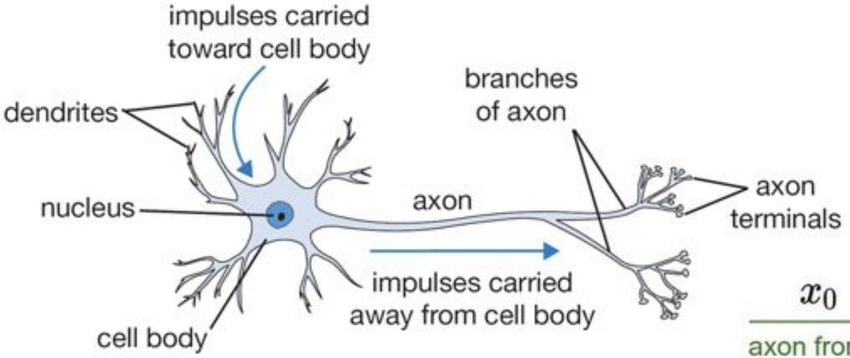
Top speech recognition systems rely on sophisticated pipelines composed of multiple algorithms and hand-engineered processing stages. In this paper, we describe an end-to-end speech system, called “Deep Speech”, where deep learning supersedes these processing stages. Combined with a language model, this approach achieves higher performance than traditional methods on hard speech recognition tasks while also being much simpler. These results are made possible by training a large recurrent neural network (RNN) using multiple GPUs and thousands of hours of data. Because this system learns directly from data, we do not require specialized components for speaker adaptation or noise filtering. In fact, in settings where robustness to speaker variation and noise are critical, our system excels: Deep Speech outperforms previously published methods on the Switchboard

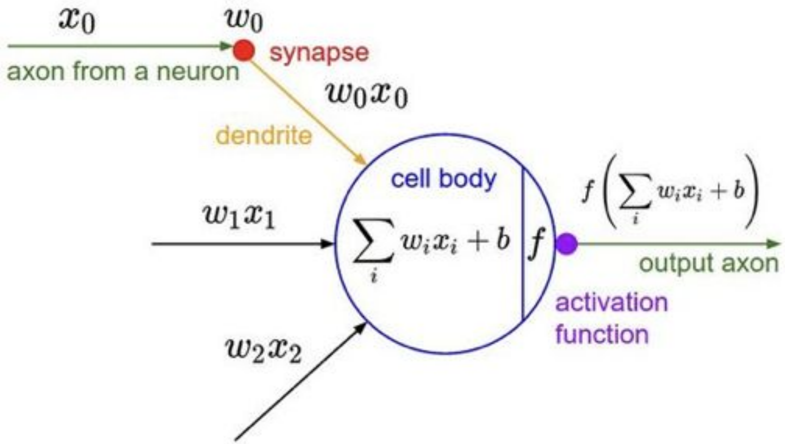
Baidu’s Deep Speech paper

The main points to create a speech-to-text algorithm

- No complex pipelines
- Using only Deep Learning through RNN
- Leverage GPUs and parallel processes
- A big dataset to learn even with background noises

ANN, Artificial Neural Network





RNN, Recurrent Neural Network

“Recurrent Neural Networks in a class of ANN where connections between nodes form a directed graph along a sequence.

This allows it to exhibit temporal dynamic behavior for a time sequence.

Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.”

https://en.wikipedia.org/wiki/Recurrent_neural_network

Baidu's Deep Speech paper, RNN

Recurrent Neural Networks in Baidu's paper

- 3 forward-layers
- 1 bi-directional recurrent layer
- 1 forward using forward and backward recurrent layers as input

$$h_{t,k}^{(6)} = \hat{y}_{t,k} \equiv \mathbb{P}(c_t = k|x) = \frac{\exp(W_k^{(6)} h_t^{(5)} + b_k^{(6)})}{\sum_j \exp(W_j^{(6)} h_t^{(5)} + b_j^{(6)})}$$

- h, inputs
- W, weight matrix for layers
- b, bias matrix for layers

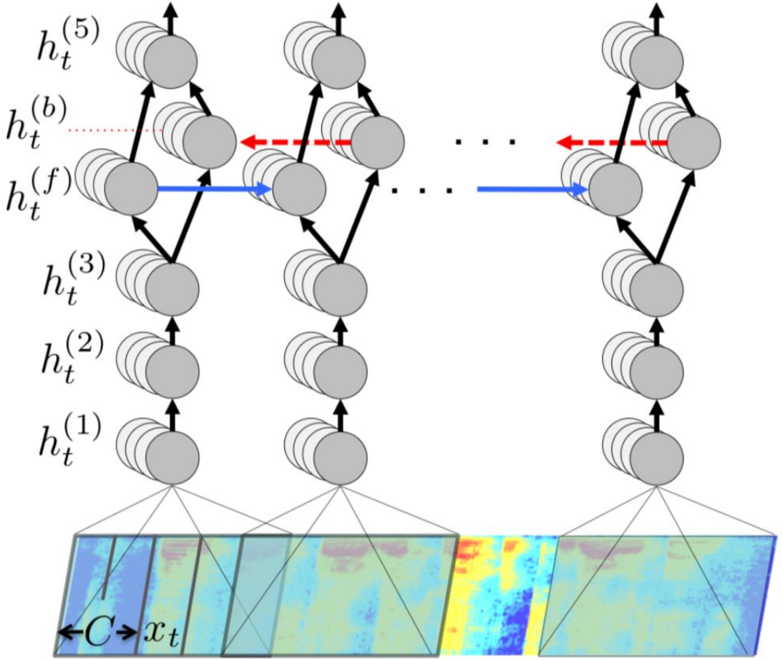
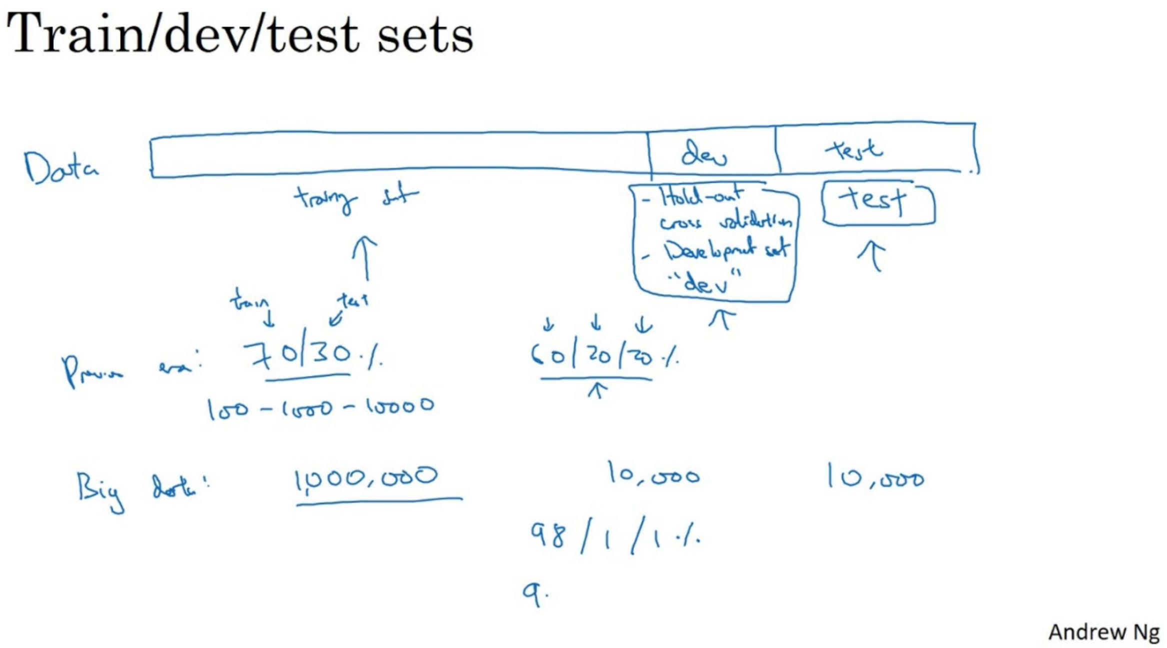


Figure 1: Structure of our RNN model and notation.

Dataset to train and test the algorithm



https://www.coursera.org/specializations/deep-learning

Background noises and pitch effect

Problems

- Different background sounds
- “Lombard Effect” when speakers actively change their pitch or inflections of their voice to overcome noise around them

Solutions

- Have a dataset with a mix of tracks with a clean and a noisy background
- Analyse and test the divergence from average behaviour in a real speech

Developing DeepSpeech

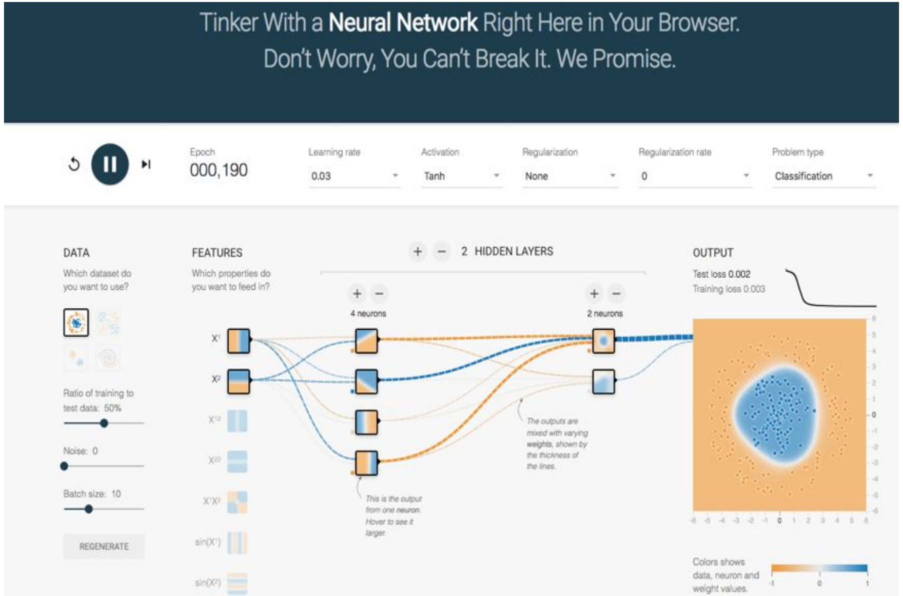
How Mozilla team develop this?

- Goal: WER (Word Error Rate) below 10%
- Implement Neural Networks using TensorFlow
- Tuning of the Hyperparameters
- Found or create a decent dataset for training and testing

TensorFlow

To get an idea try online on https://playground.tensorflow.org

Right now has been used use of LSTM and Adam optimizer (https://arxiv.org/pdf/1412.6980.pdf)



Models and hyperparameters

The hyperparameters used to train the model are useful for fine tuning.

Models and docs https://github.com/mozilla/DeepSpeech/releases/tag/v0.1.1

Last version v0.2.0-alpha.9

- `train_files` Fisher, LibriSpeech, and Switchboard training corpora.
- `dev_files` LibriSpeech clean dev corpus
- `test_files` LibriSpeech clean test corpus
- `train_batch_size` 12
- `dev_batch_size` 8
- `test_batch_size` 8
- `epoch` 13
- `learning_rate` 0.0001
- `display_step` 0
- `validation_step` 1
- `dropout_rate` 0.2367
- `default_stddev` 0.046875
- `checkpoint_step` 1
- `log_level` 0
- `checkpoint_dir` value specific to hardware setup
- `wer_log_pattern` "GLOBAL LOG: logwer('${COMPUTE_ID}', '%s', '%s', %f)"
- `decoder_library_path` value specific to hardware setup
- `n_hidden` 2048

Creating a new dataset

Common Voice

You can also contribute with your own voice!

Available dataset 12 TB, 500 hours https://voice.mozilla.org/en/data

(also on Kaggle https://www.kaggle.com/mozillaorg/common-voice)

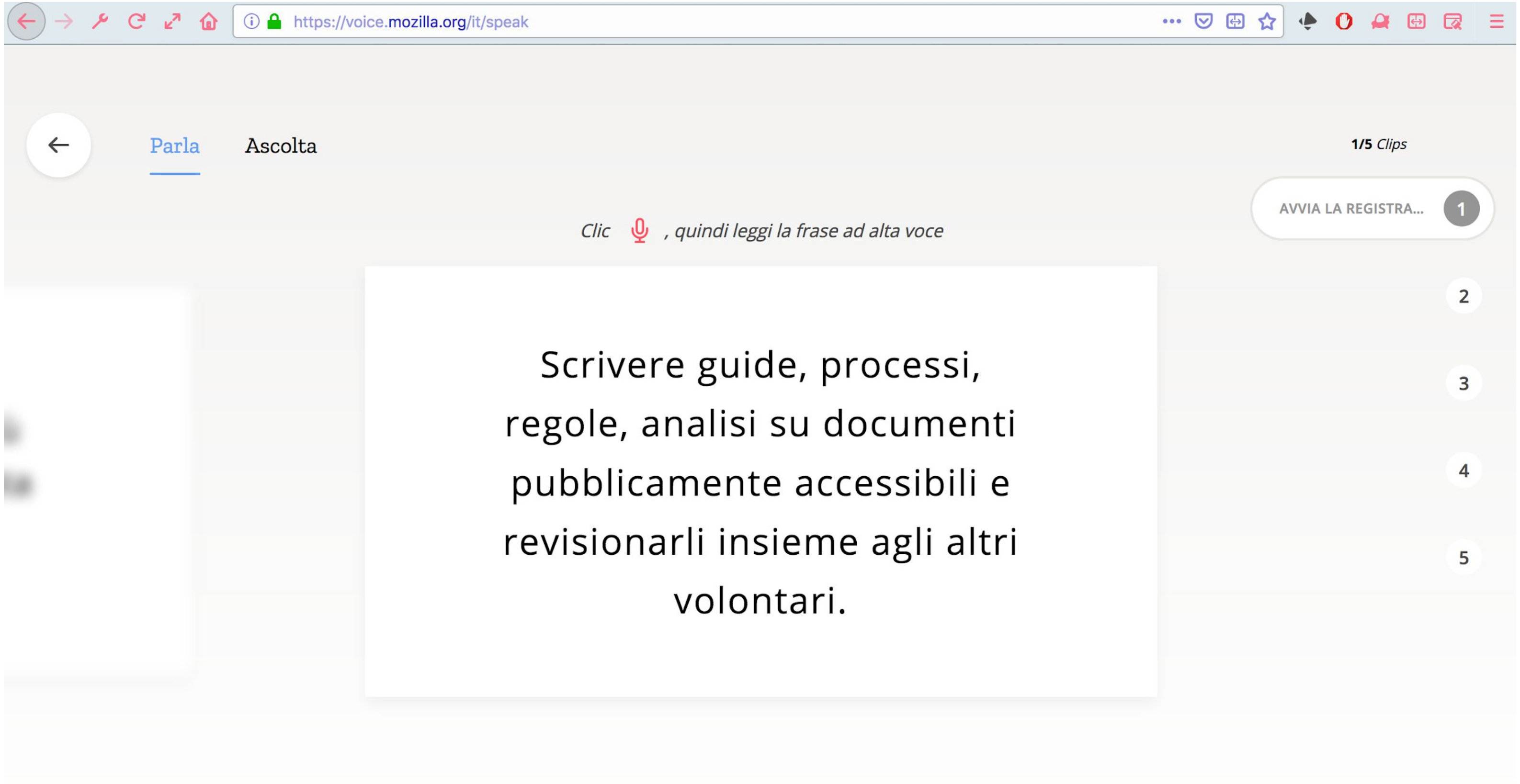
Different languages (currently around 1000 hours among all the languages).

Soon train DeepSpeech with new languages too.

Common Voice in Italian too!

Thanks to the effort of the Italian Mozilla community,
the Common Voice and DeepSpeech teams > https://voice.mozilla.org/it

Common Voice



Where can you start?

Where can you start?

Follow along the documentation

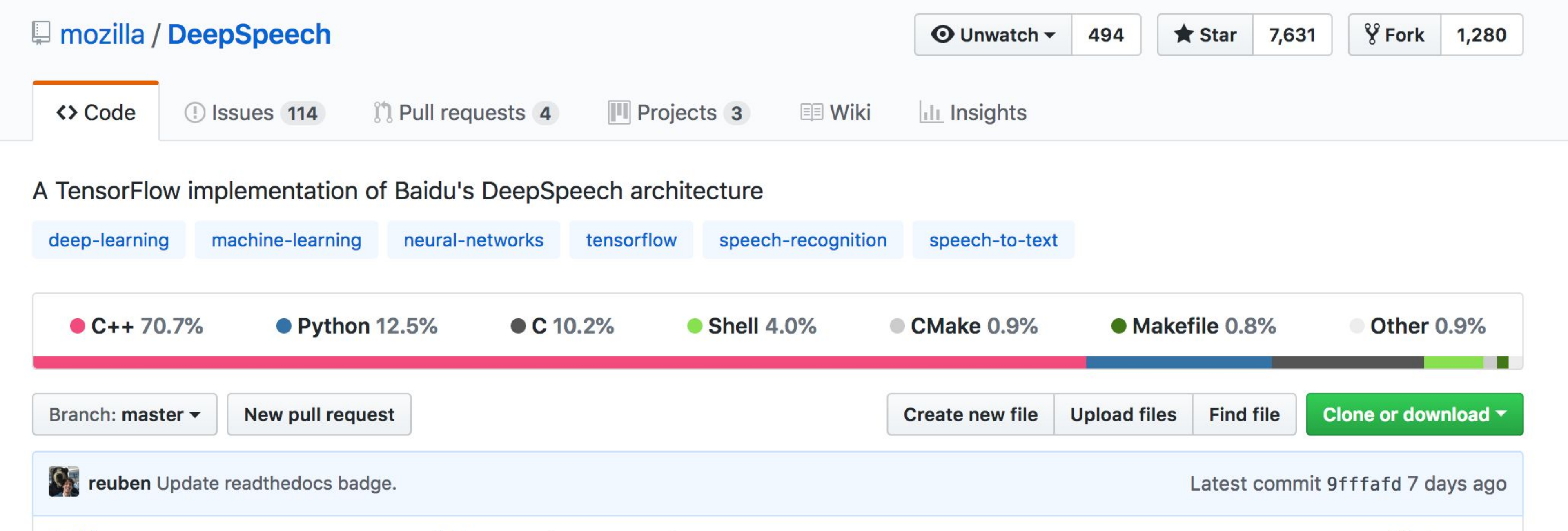
https://github.com/mozilla/DeepSpeech

```
(demo) Kellys-MacBook-Pro:DeepSpeech kdavis$ pip install deepspeech
Collecting deepspeech
  Using cached deepspeech-0.1.0-cp27-cp27m-macosx_10_12_x86_64.whl
Requirement already satisfied: numpy in ./demo/lib/python2.7/site-packages (from deepspeech)
Requirement already satisfied: scipy==0.19.1 in ./demo/lib/python2.7/site-packages (from deepspeech)
Installing collected packages: deepspeech
Successfully installed deepspeech-0.1.0
(demo) Kellys-MacBook-Pro:DeepSpeech kdavis$ deepspeech -h
usage: deepspeech [-h] model audio alphabet [lm] [trie]

Benchmarking tooling for DeepSpeech native_client.

positional arguments:
  model       Path to the model (protocol buffer binary file)
  audio       Path to the audio file to run (WAV format)
  alphabet    Path to the configuration file specifying the alphabet used by
              the network
  lm          Path to the language model binary file
  trie        Path to the language model trie file created with
              native_client/generate_trie

optional arguments:
  -h, --help  show this help message and exit
(demo) Kellys-MacBook-Pro:DeepSpeech kdavis$ tar xfvz deepspeech-0.1.0-models.tar.gz
x models/
x models/lm.binary
x models/output_graph.pb
x models/trie
x models/alphabet.txt
(demo) Kellys-MacBook-Pro:DeepSpeech kdavis$ tar xfvz audio-0.1.0.tar.gz
x audio/
x audio/2830-3980-0043.wav
x audio/._4507-16021-0012.wav
x audio/4507-16021-0012.wav
x audio/8455-210777-0068.wav
(demo) Kellys-MacBook-Pro:DeepSpeech kdavis$ deepspeech models/output_graph.pb audio/2830-3980-0043.wav models/alphabet.txt models/lm.binary models/trie
Loading model from file models/output_graph.pb
Loaded model in 1.837s.
Loading language model from files models/lm.binary models/trie
Loaded language model in 4.089s.
Running inference.
experience proves this
Inference took 9.146s for 1.975s audio file.
(demo) Kellys-MacBook-Pro:DeepSpeech kdavis$ deepspeech models/output_graph.pb audio/4507-16021-0012.wav models/alphabet.txt models/lm.binary models/trie
Loading model from file models/output_graph.pb
Loaded model in 1.795s.
Loading language model from files models/lm.binary models/trie
Loaded language model in 4.000s.
Running inference.
```

Table of Contents

How you can contribute even more?

- Written mostly in C++, plus some Python and C
- https://github.com/mozilla/DeepSpeech/issues



Deep Speech and other languages

- Rust https://github.com/RustAudio/deepspeech-rs
- Go https://github.com/asticode/go-astideepspeech
- GStreamer https://github.com/Elleo/gst-deepspeech

Resources

On Discourse https://discourse.mozilla.org/c/deep-speech

Mozilla on irc #machinelearning

FOSDEM 2018 talk
https://archive.fosdem.org/2018/schedule/event/mozilla_deepspeech_common_voice_projects

For contribution in Italian and to get in touch with the community, look for **Mozilla Italia - HOME** on Telegram or https://forum.mozillaitalia.org

Thank you!

Stefania Delprete

stefania.delprete@top-ix.org

@astrastefania