

# Deep Speech decoder breakdown

- [Generate trie](#)
- [Deep Speech Scorer Refactor](#)
- [Potentially using tensorflow api in python instead of c++](#)

# Generate trie

This issues a call to `scorer::setup()`?

# Deep Speech Scorer Refactor

I have successfully renamed all variables dealing with the scorer and reran everything. The next step is to refactor into a generic and an inherited version.

# Potentially using tensorflow api in python instead of c++

the tensorflow api in c++ is a bear to use. no one really uses it much and it has some noteworthy differences from the ubiquitous python api.

the issue is that the scorer in `ds_ctcdecoder` is built using a bunch of c++ code with swig wrappers to expose the functionality to python.

much of my previous work has involved writing a new scorer in c++ using the c++ tensorflow api and then wrapping this in swig methods like the existing scorer.

i've spent countless hours just trying to get bazel to build things correctly and I have had very limited success.

in particular the error i'm running into now is that I don't know how to build against the flatbuffer library that is required for reading in tensorflow lite model files.

one alternative course of action is to use the python api for tensorflow which comes with the benefit of having more documentation and community support behind it.

the issue is that it's unclear how to integrate python in with the wrapped c++ code in a way that's maintainable for the future of deepspeech.

I cannot modify the `swigwrapper.py` file as this is autogenerated and is removed upon each make clean in the `ctc_decoder` directory.