FAQ

What is fastText? Are there tutorials?

FastText is a library for text classification and representation. It transforms text into continuous vectors that can later be used on any language related task. A few tutorials are available.

How can I reduce the size of my fastText models?

fastText uses a hashtable for either word or character ngrams. The size of the hashtable directly impacts the size of a model. To reduce the size of the model, it is possible to reduce the size of this table with the option '-hash'. For example a good value is 20000. Another option that greatly impacts the size of a model is the size of the vectors (-dim). This dimension can be reduced to save space but this can significantly impact performance. If that still produce a model that is too big, one can further reduce the size of a trained model with the quantization option.

```
./fasttext quantize -output model
```

What would be the best way to represent word phrases rather than words?

Currently the best approach to represent word phrases or sentence is to take a bag of words of word vectors. Additionally, for phrases like “New York”, preprocessing the data so that it becomes a single token “New_York” can greatly help.

Why does fastText produce vectors even for unknown words?

Why is the hierarchical softmax slightly worse in performance than the full softmax?

The hierarchical softmax is an approximation of the full softmax loss that allows to train on large number of class efficiently. This is often at the cost of a few percent of accuracy. Note also that this loss is thought for classes that are unbalanced, that is some classes are more frequent than others. If your dataset has a balanced number of examples per class, it is worth trying the negative sampling loss (-loss ns -neg 100). However, negative sampling will still be very slow at test time, since the full softmax will be computed.

Can we run fastText program on a GPU?

As of now, fastText only works on CPU. Please note that one of the goal of fastText is to be an efficient CPU tool, allowing to train models without requiring a GPU.

Can I use fastText with python? Or other languages?

Python is officially supported. There are few unofficial wrappers for javascript, lua and other languages available on github.

Can I use fastText with continuous data?

FastText works on discrete tokens and thus cannot be directly used on continuous tokens. However, one can discretize continuous tokens to use fastText on them, for example by rounding values to a specific digit ("12.3" becomes "12").

There are misspellings in the dictionary. Should we improve text normalization?

If the words are infrequent, there is no need to worry.

I'm encountering a NaN, why could this be?

My compiler / architecture can't build fastText. What should I do?

Try a newer version of your compiler. We try to maintain compatibility with older versions of gcc and many platforms, however sometimes maintaining backwards compatibility becomes very hard. In general, compilers and tool chains that ship with LTS versions of major linux distributions should be fair game. In any case, create an issue with your compiler version and architecture and we'll try to implement compatibility.

How do I run fastText in a fully reproducible way? Each time I run it I get different results.

If you run fastText multiple times you'll obtain slightly different results each time due to the optimization algorithm (asynchronous stochastic gradient descent, or Hogwild). If you need to get the same results (e.g. to confront different input params set) you have to set the 'thread' parameter to 1. In this way you'll get exactly the same performances at each run (with the same input params).

Why do I get a probability of 1.00001?

This is a known rounding issue. You can consider it as 1.0.

← PYTHON MODULE API →