FastText Tutorial – Learn NLP Library Tools

FastText

- Introduction
- Build FastText
- Capabilities
- Text Classification
- Word Representations Learning

Introduction

FastText is an opensource and freeware library, built by Facebook, for making the natural language processing tasks like Word Representation & Sentence Classification (/Text Classification/Document Classification/Sentiment Analysis) much more efficient.

FastText is being distributed under BSD licence, which means you may modify the source code, and use it for your private projects, distribute it and also use for commercial purpose. But keep in mind that it comes with no warranties and liabilities. You should never point back to its developers or facebook if you mess up things during its inclusion in your project.

Build FastText

Currently FastText could be built from source on Linux distributions and Mac OS. For compiling the sources, it requires either gcc-4.6.3 (or newer) or clang-3.3 (or newer), python 2.6 or higher, numpy and scipy.

Build FastText [icon name="link" class="" unprefixed_class=""] – FastText Tutorial to make a build from source, on Linux Distribution(like Ubuntu, CentOS, etc.) or MacOS

FastText Tutorial

There are a series of sections on FastText to help you understand the different modules present in the FastText library and help you get started using this in your own projects.

Capabilities of FastText

FastText library provides following capabilities [FastText command_name is provide in the bracket] through its tools.

- **Training Supervised Classifier** [supervised] Supervised Classifier Training for Text Classification.
- **Training SkipGram Model** [skipgram] Learning Word Representations/Word Vectors using skipgram technique.
- **Quantization** [quantize] Quantization is a process applied on a model so as to reduce the memory usage during prediction.
- **Predictions** [predict] Predicting labels for a given text : Text Classification.
- **Predictions with Probabilities** [predict-prob] Predicting probabilities in addition to labels for a given text : Text Classification.
- **Training of CBOW model** [cbow] Learning Word Representations/Word Vectors using CBOW (Continuous Bag Of Words) technique.
- **Print Word Vectors** [print-word-vectors] Printing of Word Vectors for a trained model with each line representing a word vector.
- **Print Sentence Vectors** [print-sentence-vectors] Printing of Sentence Vectors for a trained model with each line representing a vector for a paragraph.
- **Query Nearest Neighbors** [nn]
- **Query for Analogies** [analogies]

Text Classification

Text Classification or Document Classification (also called Sentiment Analysis) is an NLP (Natural Language Processing) task of predicting the amount of chance a given text belongs to each possible categories. An Email classification to SPAM or NOT-A-SPAM is a classic example for Text Classification.

FastText provides following commands for all the required functionalities like training, testing etc., in Text Classification :

Training

supervised – to train a supervised model

[Train and Test Supervised Text Classifier using fasttext](#) [icon name="link" class="" unprefixed_class=""] – Fasttext Tutorial to train a supervised text classifier using labelled data and test the generated model for accuracy and performance numbers.

Optimization

quantize – to reduce the memory usage

Testing

test – to test a model

Prediction

predict – to estimate the categories for a given text**predict-prob** – to estimate the categories with probabilities for a given text

Word Representations Learning

Vector representation of words, in a language, captures information like analogies or semantic. Also this technique of word representation as a vector is being used to improve text classifiers' performance. Therefore, Word Representations Learning works as an aid to Text Classifiers in increasing their performance.

[Learn Word Representations in FastText](#) [icon name="link" class="" unprefixed_class=""] using Unsupervised Learning techniques – CBOW (Continuous Bag Of Words) and SkipGram.

Conclusion

FastText is being used by Facebook in optimizing the advertisements for its users based on the users' posts and status updates. FastText is obtaining a lot of traction in the natural language processing community. Also with the wide user base of Facebook, it is really taking advantage of the data flowing into its data servers to create better and diversified models required for sentiment analysis and text classification. And this FastText Tutorial will help you to get started and learn the capabilities provided by FastText library.

FastText Tutorial

⇨ **FastText Tutorial Home**