

Learn Git and GitHub without any code!

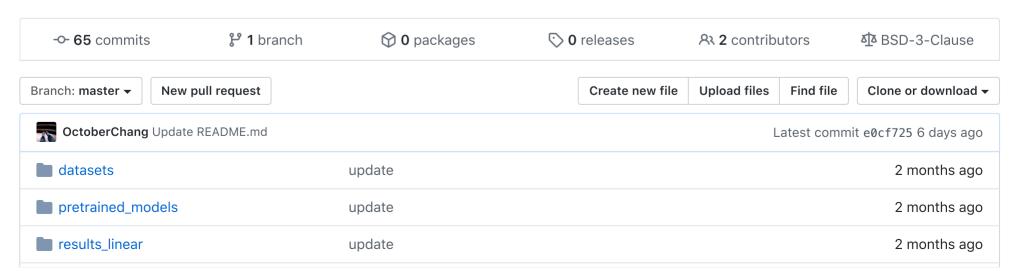
Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

□ OctoberChang / Transformer-XMC

Transformer-XMC: Taming Pretrained Transformers for eXtreme Multi-label Text Classification

#deep-learning #extreme-multi-label-classification #pytorch #nlp #transformers



results_transformer-large	update	2 months ago
■ xbert	update linear pipeline (indexer + ranker)	4 months ago
.gitignore	first commit	11 months ago
LICENSE	first commit	11 months ago
Makefile	update datasets, pretrained models, and scripts	11 months ago
□ README.md	Update README.md	6 days ago
environment.yml	fix conda env when using pip	4 months ago
eval_linear.sh	update eval scripts	2 months ago
eval_transformer.sh	update eval scripts	2 months ago
setup.py	update datasets, pretrained models, and scripts	11 months ago

☐ README.md

Transformer-XMC: Taming Pretrained Transformers for eXtreme Multi-label Text Classification

This is a README for the experimental code in our paper

X-BERT: eXtreme Multi-label Text Classification with BERT

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, Inderjit Dhillon

Preprint 2019

Installation

Depedencies via Conda Environment

```
> conda env create -f environment.yml
> source activate pt1.2_xmlc_transformer
> (pt1.2_xmlc_transformer) pip install -e .
```

Reproduce Evaulation Results in the Paper

We demonstrate how to reproduce the evaluation results in our paper by downloading the raw dataset and pretrained models.

Download Dataset (Eurlex-4K, Wiki10-31K, AmazonCat-13K, Wiki-500K)

Change directory into ./datasets folder, download and unzip each dataset

```
cd ./datasets
bash download-data.sh Eurlex-4K
bash download-data.sh Wiki10-31K
bash download-data.sh AmazonCat-13K
bash download-data.sh Wiki-500K
cd ../
```

Each dataset contains the following files

- X.trn.npz, X.tst.npz: instance's embedding matrix (either sparse TF-IDF or fine-tuned dense embedding)
- Y.trn.npz, Y.tst.npz: instance-to-label assignment matrix

^{**}Notice: the following examples are executed under the > (xbert-env) conda virtual environment

- L.pifa.npz, L.pifa-neural.npz, L.text-emb.npz: label's embedding matrix
- train text.txt, test text.txt: each line is raw_text
- train_labels.txt, test_labels.txt: each line is a seq of labels, seperated by whitespace
- train.txt, test.txt: each line is (label_1,...,label_k) \tab tf-idf features

Download Pretrained Models (Indexing codes, fine-tuned Transformer models)

Change directory into ./pretrained_models folder, download and unzip models for each dataset

```
cd ./pretrained_models
bash download-models.sh Eurlex-4K
bash download-models.sh Wiki10-31K
bash download-models.sh AmazonCat-13K
bash download-models.sh Wiki-500K
cd ../
```

Evaluate Linear Models

Given the provided indexing codes (label-to-cluster assignments), train/predict linear models, and evaluate with Precision/Recall@k:

```
export DATASET=Eurlex-4K
export VERSION=v0
bash eval_linear.sh ${DATASET} ${VERSION}
```

- DATASET: the dataset name such as Eurlex-4K, Wiki10-31K, AmazonCat-13K, or Wiki-500K.
- v0 : instance embedding using sparse TF-IDF features
- v1: instance embedding using sparse TF-IDF features concatenate with dense fine-tuned XLNet embedding

The evaluaiton results should located at ./results_linear/\${DATASET}.\${VERSION}.txt

Evaluate Fine-tuned Transformer-XMC Models

Given the provided indexing codes (label-to-cluster assignments) and the fine-tuned Transformer models, train/predict ranker of the Transformer-XMC framework, and evaluate with Precision/Recall@k:

```
export DATASET=Eurlex-4K
bash eval_transformer.sh ${DATASET}
```

• DATASET: the dataset name such as Eurlex-4K, Wiki10-31K, AmazonCat-13K, or Wiki-500K.

The evaluation results should located at ./results_transformer-large/\${DATASET}/feat-joint_neg-yes_noop.txt

Pipeline for running Transformer-XMC on a new dataset

**To be released. Stay-tuned!

Acknowledge

Some portions of this repo is borrowed from the following repos:

- transformers(v2.2.0)
- liblinear
- TRMF