NewsRecLib: A PyTorch-Lightning Library for Neural News Recommendation

NewsRecLib

Andreea Iana[1], Goran Glavaš[2], Heiko Paulheim[1]

[1]Data and Web Science Group, University of Mannheim, Germany
[2]Center for Artificial Intelligence and Data Science, University of Würzburg, Germany
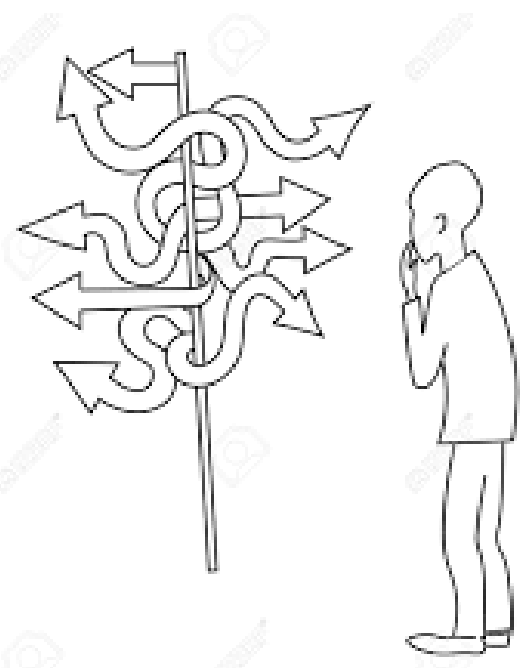andreea.iana@uni-mannheim.de

Code & Docs

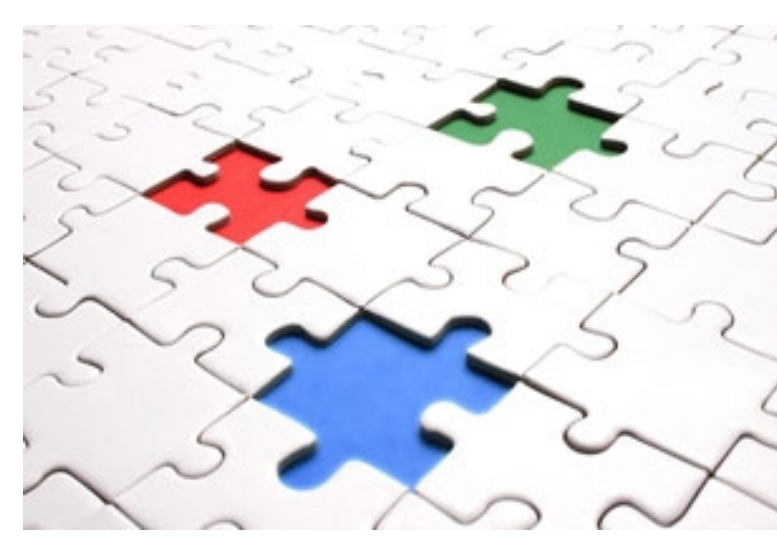## Neural News Recommendation Lacks Reproducibility & Fair Evaluation

Many factors hinder reproducibility & extensibility

Often code N/A or hard to use

Various libraries & implementation differences

Missing hyperparameter details

Many factors inhibit fair model comparisons
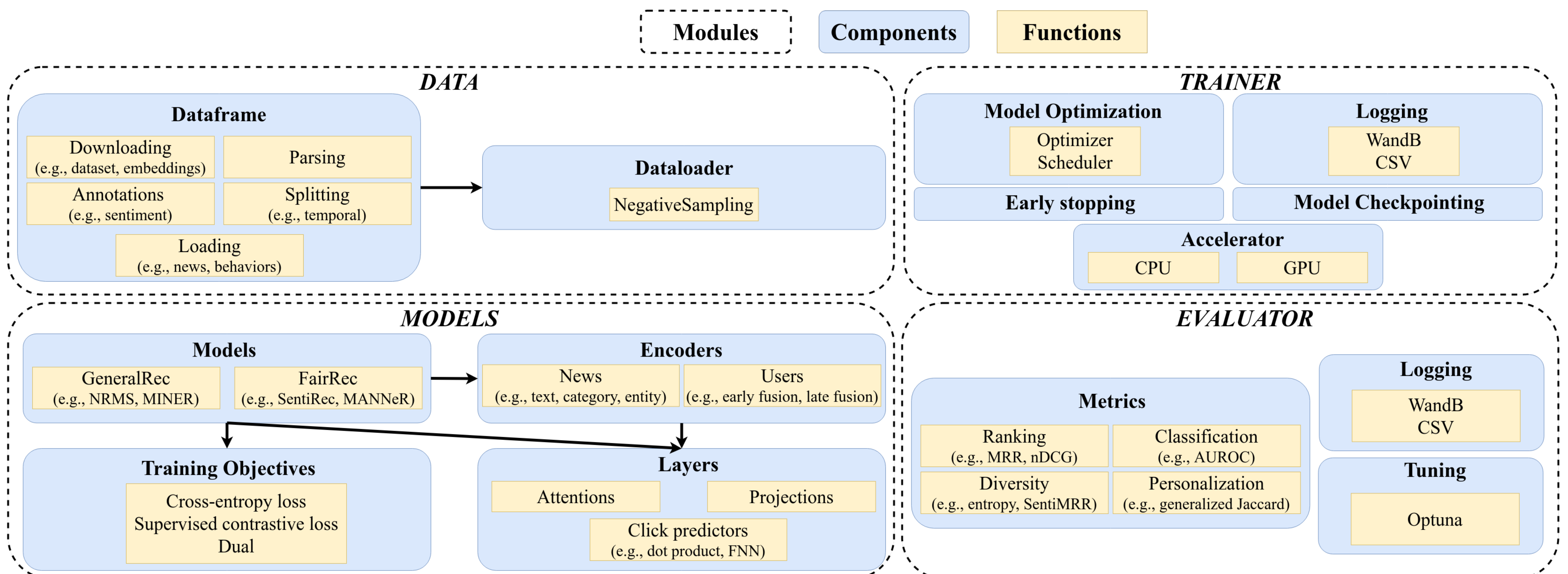
Usage of proprietary datasets

Lack of transparency in experiments

Ad-hoc evaluation protocols

NewsRecLib: Reproducible Research & Rigorous Experimental Evaluation

Modules   Components   Functions

DATA

Dataframe

Downloading (e.g., dataset, embeddings)
Parsing
Annotations (e.g., sentiment)
Splitting (e.g., temporal)
Loading (e.g., news, behaviors)

Dataloader

NegativeSampling

TRAINER

Model Optimization

Optimizer
Scheduler

Early stopping

Logging

WandB
CSV

Model Checkpointing

Accelerator

CPU   GPU

MODELS

Models

GeneralRec (e.g., NRMS, MINER)
FairRec (e.g., SentiRec, MANNeR)

Encoders

News (e.g., text, category, entity)
Users (e.g., early fusion, late fusion)

Training Objectives

Cross-entropy loss
Supervised contrastive loss
Dual

Layers

Attentions
Projections
Click predictors (e.g., dot product, FNN)

EVALUATOR

Metrics

Ranking (e.g., MRR, nDCG)
Classification (e.g., AUROC)
Diversity (e.g., entropy, SentiMRR)
Personalization (e.g., generalized Jaccard)

Logging

WandB
CSV

Tuning

Optuna

⚙️ Modularity & Extensibility

PyTorch Lightning

➲ Decoupled core components

➲ Class extension or new Lightning module definition for new features

⚙️ Configurability & Reproducibility

Hydra

➲ Dedicated configuration file per module

➲ Experiments: dynamically create hierarchical configurations by composition

⚙️ Logging & Profiling

Weights & Biases   TensorBoard

➲ Comprehensive & transparent logging

➲ Model evaluation with TorchMetrics

➲ Any profiler from PyTorch

Easy Benchmarking from Command Line

Access pipeline with simple CLI commands

python train.py experiment=nrms_mind_small **model**=nrms **data**=mind **dataset_size**=small

Experiments on MINDsmall