

Baseline: Strong, Extensible, Reproducible, Deep Learning Baselines for NLP

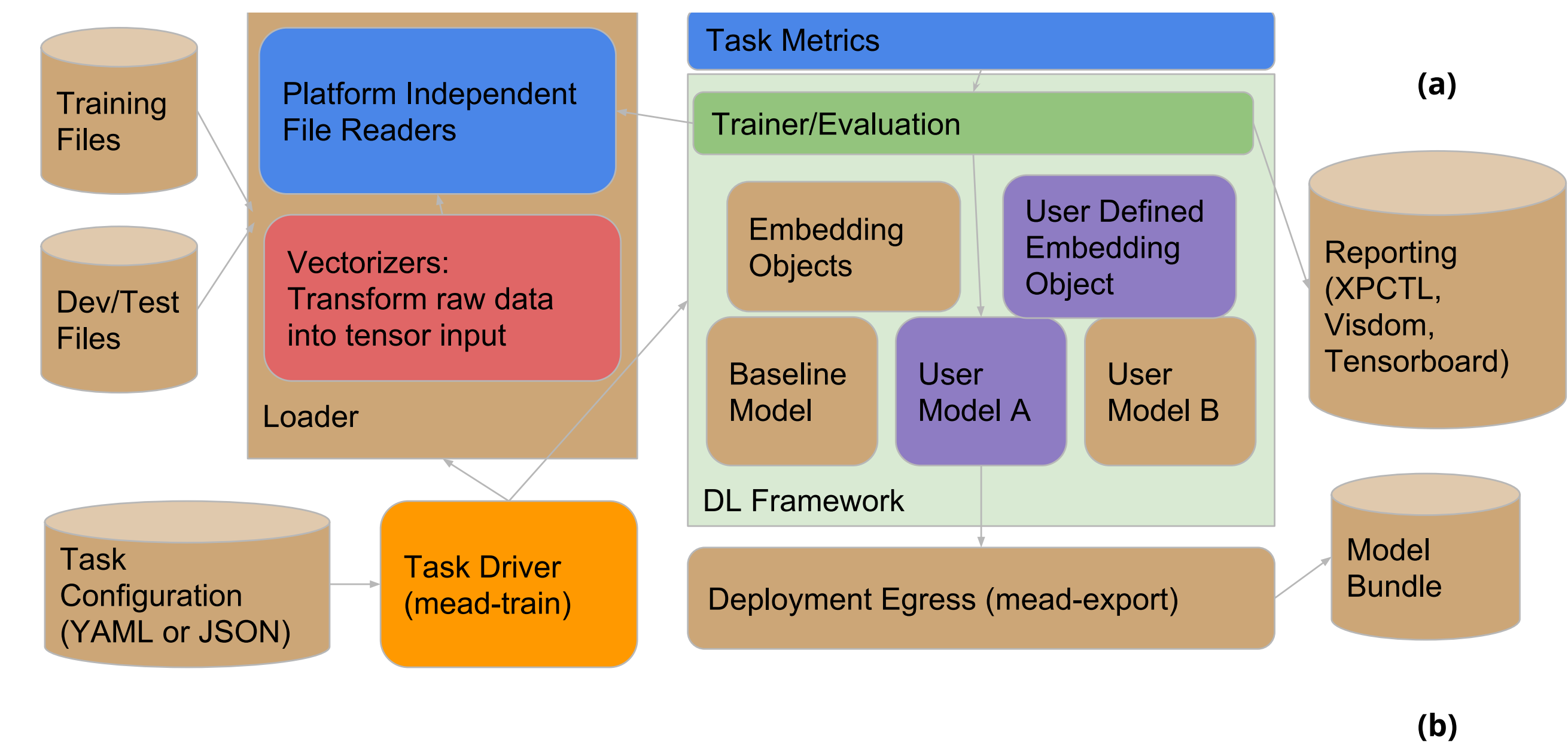
Daniel Pressel, Brian Lester, Sagnik Ray Choudhury, Matt Barta, Yanjie Zhao, Amy Hemmeter
Interactions Digital Roots, 330 E Liberty St, Ann Arbor, MI 48104
{dpressel, blester, schoudhury, mbarta, yzhao, ahemmeter}
@interactions.com



Introduction

Problems in current NLP research : common yet error-prone parts of the research process are re-implemented, hyperparameters are not tuned exhaustively, and low-performing baselines are used to show large relative gains. In addition, building deep learning models for research and evaluation is cumbersome.

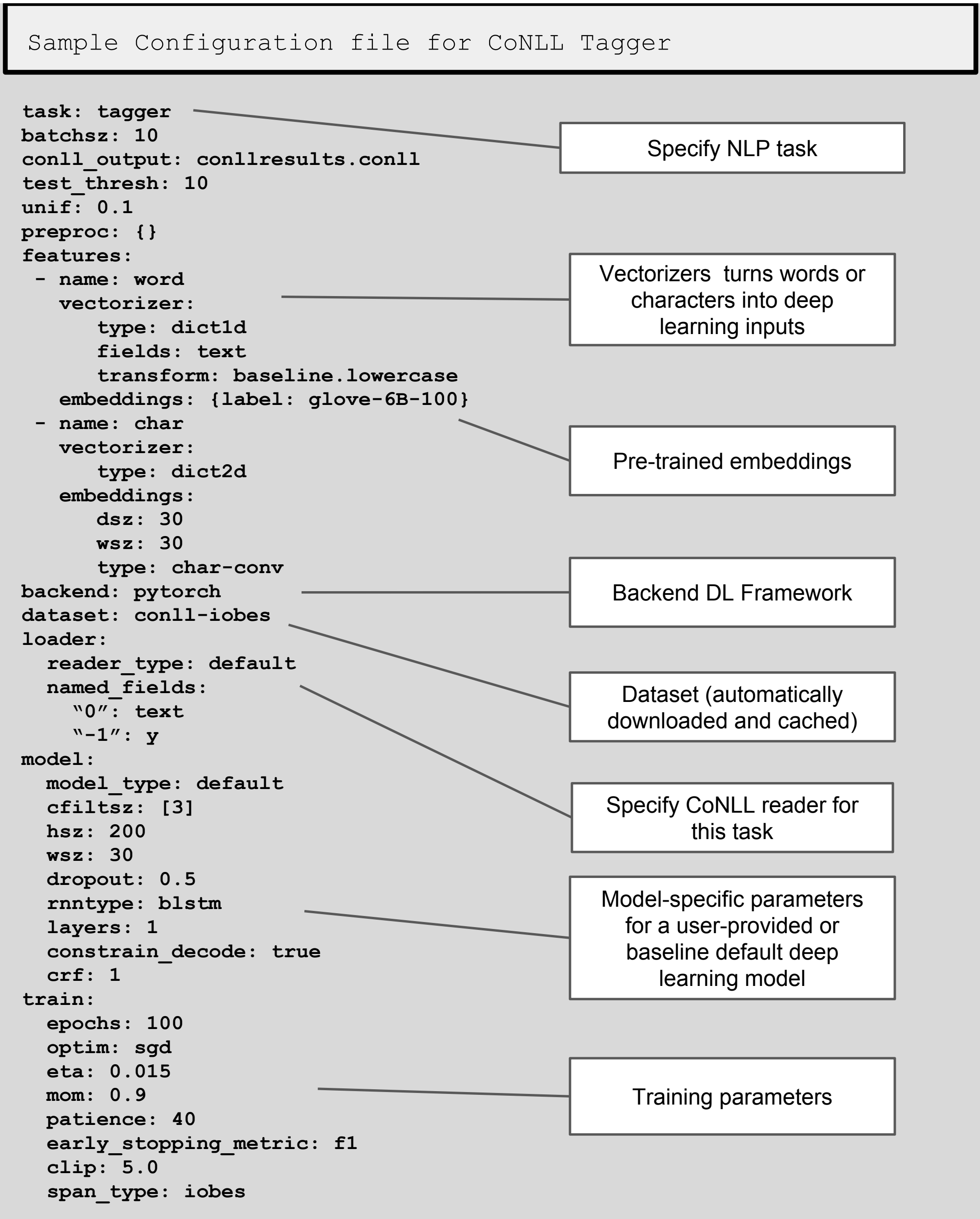
Baseline¹ provides a plug and play architecture to solve these problems



- **Comprehensive pipeline:** select which aspects of Baseline to use, allowing for easy swapping of datasets, models, and other aspects of the deep learning process
- **Strong baselines:** models provided for classification, sequence tagging, encoder-decoder, and language modeling
- **Encourages reproducible research:** XPCTL (the experimental control module) saves model results and experiment details through a simple leaderboard interface
- **Flexibility with frameworks:** choose between three popular deep learning frameworks — Tensorflow, PyTorch and DyNet
- **Automatic hyperparameter tuning:** HPCTL (hyperparameter control) helps find optimal hyperparameters given a search template
- **Leverages complex embedding strategies:** Represents embeddings as arbitrary sub-graphs
- **Easy model exporter:** easily export the models to production environments such as Tensorflow Serving

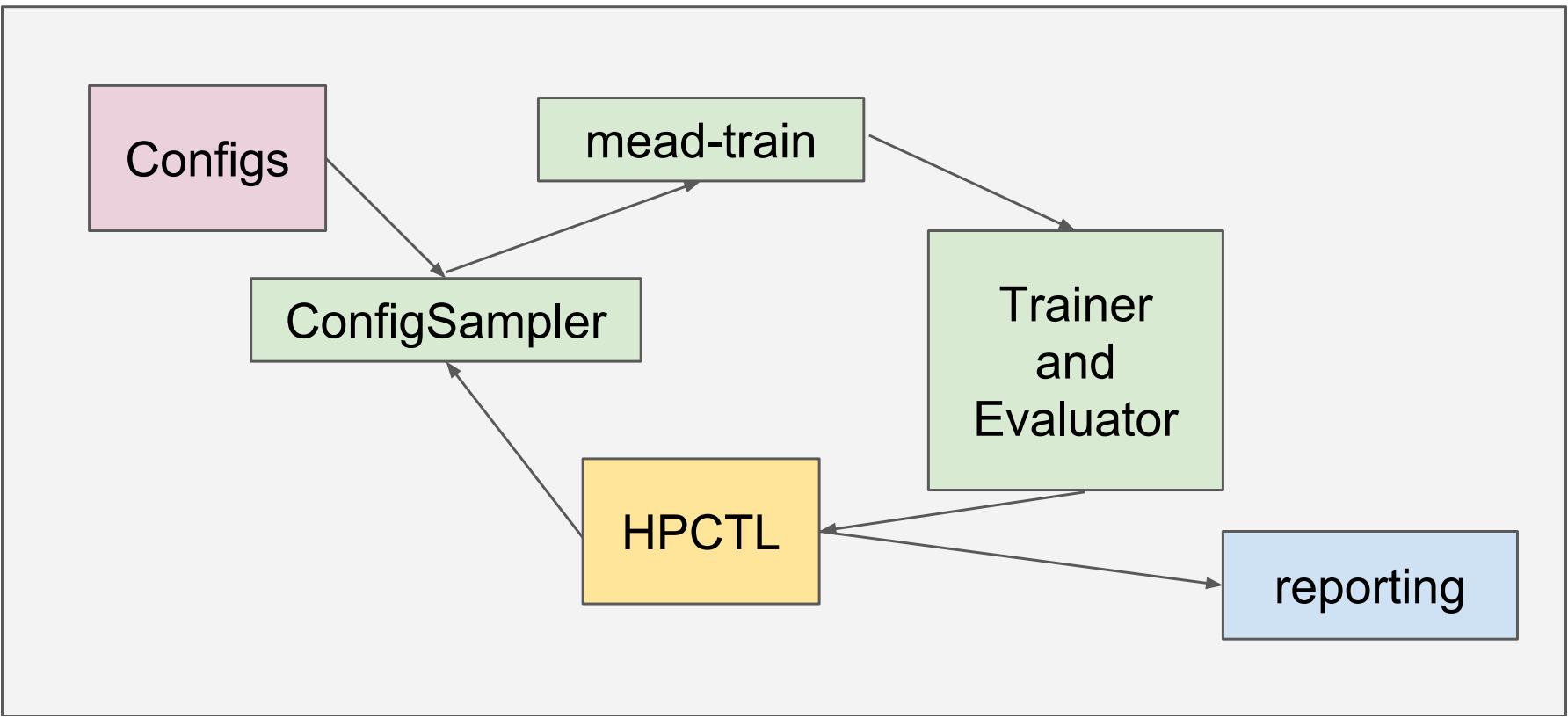
User Add-ons

- Define a subclass of a Baseline objects like a model, vectorizer, or reader.
- Decorate with the appropriate `@register` decorator and give it a name to use in the configuration file.
- Use the name from the decorator and add the file name to the `modules` list in the configuration file; make sure the python file is in your `PYTHONPATH`
- Run with `mead-train` like you would with any of the supplied baselines



Components of Baseline

- **Core API** - a Python library of extensible classes and baselines for the most common NLP tasks
- **MEAD** - Modeling, Experimentation And Development - contains driver programs to run experiments using JSON or YAML configuration files (as above)
- **XPCTL** - a command-line interface to track experimental results and provide access to a global leaderboard
- **HPCTL** - a program built on MEAD to support hyper-parameter searching for finding optimal parameters using a template (see figure below)



Conclusion

- Baseline automates the frustrating parts of the process of model development and deployment so researchers can focus on innovation in both research and production
- We will continue to improve our baselines as the state-of-the-art changes (see the repository for most updated results).

¹<https://github.com/dpressel/baseline>