

# Neural Monkey

Jindřich Helcl, Dušan Variš

Institute of Formal and Applied Linguistics  
Faculty of Mathematics and Physics  
Charles University

August 30, 2017



# Outline

---

## Introduction

- General Architecture
- Configuration Files

## Getting Started

- Installation
- Simple Exercise

## Exercises

# Neural Monkey

---

- Toolkit for training neural models for sequence-to-sequence tasks
- Implemented in Python 3 using Tensorflow 1.3
- GPU support using CUDA, cuDNN
- Modular implementation of parts of computational graph → easy composition of new models
- Applications in research [Kreutzer et al., 2017, Libovický and Helcl, 2017]

# Model Workflow

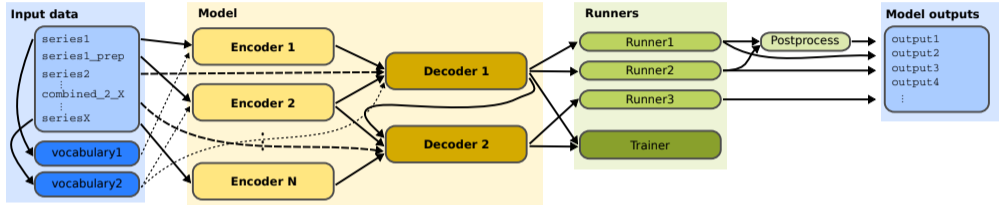


Figure 1: Model workflow.

Each step of the workflow can be modified/expanded by changing a corresponding section in the model configuration file.

# Configuration Files

---

- Experiment specification:
  - model definition, training, inference
  - data location, preprocessing, output postprocessing
  - experiment metaparameters, evaluation metrics
- INI file syntax
  - Sections defining separate objects
  - Key-value pairs separated by '='
  - Values can be atomic (int, boolean, string) or composite (list, objects)
  - Sections are interpreted as Python dictionaries

# Configuration Files

---

```
; configuration snippet example
[main]
name="example"
tf_manager=<tf_manager>
output="output/dir"
batch_size=16
epochs=2
train_dataset=<train_data>
val_dataset=<val_data>
runners=[<runner>]
evaluation=[("target", <bleu>)]

[tf_manager]
class=tf_manager.TensorFlowManager
num_sessions=1
; (...)
```

# Installation

---

1. (Optional) Set up a Python virtualenv (with Python<sub>i</sub>=3.5)
2. `$ git clone https://github.com/ufal/neuralmonkey`
3. `$ cd neuralmonkey`
4. `$ pip install -r requirements.txt`
5. `$ ./run_tests.sh` # (Optional) Check the installation

# Running the Monkey

---

- Training: `bin/neuralmonkey-train <experiment_config>`
  - e.g. `bin/neuralmonkey-train tests/small.ini`
- Running the model: `bin/neuralmonkey-run <experiment_config>`  
`<data_config>`
  - e.g. `bin/neuralmonkey-run tests/small.ini tests/test_data.ini`
  - `experiment_config` can be the same file that was used during training
  - `data_config` specifies the dataset for the inference and (if on non-default location) variable files to load the model from



## Exercises: Directory Structure

---

- Configuration files: `~/experiments`
- Experiment data: `~/data`
- Experiment output: `~/experiments/<experiment_name>`

## Exercises: Prepared Config Files

---

- Machine Translation: `~/experiments/translation.ini`
- Text Summarization: `~/experiments/summarization.ini`
-

## Exercises: Modifying the Experiments

---

Choose a task from previous slide and try changing the existing config file.

1. Run the task on character-level architecture. Use `neuralmonkey.processors.helpers` classes to pre/postprocess the input sentence.
2. Use two encoders (with a similar architecture) to encode both word representation and character representation of the sentence.
3. Replace the GreedyRunner by BeamSearchRunner, and Decoder by BeamSearchDecoder.
4. Use different encoder (see `neuralmonkey/encoders` for possible substitutes).