Neural Text Matching Toolkit

Yixing Fan

fanyixing@ict.ac.cn

University of Chinese Academy of Sciences
CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, CAS

20190215

Text matching

How many people live in Melbourne

What’ s the population of Melbourne

Text matching

$$\text{Match}(T1,T2) = F(\phi(T1),\phi(T2))$$

Composition Function

Interaction Function

Task	Text 1	Text 2
Information retrieval	query	document
Question answering	question	answer
Automatic conversation	dialog	response
Paraphrase Identification	string A	string B

Text matching is a core task in natural language processing.

Text matching

A number of deep matching models have been proposed!

Information Retrieval

- ✓ DSSM [Huang et al. 2013]
- ✓ CDSSM [Ye et al. 2014]
- ✓ DRMM [Guo et al. 2016]
- ✓ Duet [Mitra et al. 2017]
- ✓ K-NRM [Xiong et al. 2017]
- ✓ PACRR [Hui et al. 2017]
- ✓ DeepRank [Pang et al. 2017]
- ✓ Conv-KNRM [Dai et al. 2018]
- ✓ HiNT [Fan et al. 2018]
- ✓ …

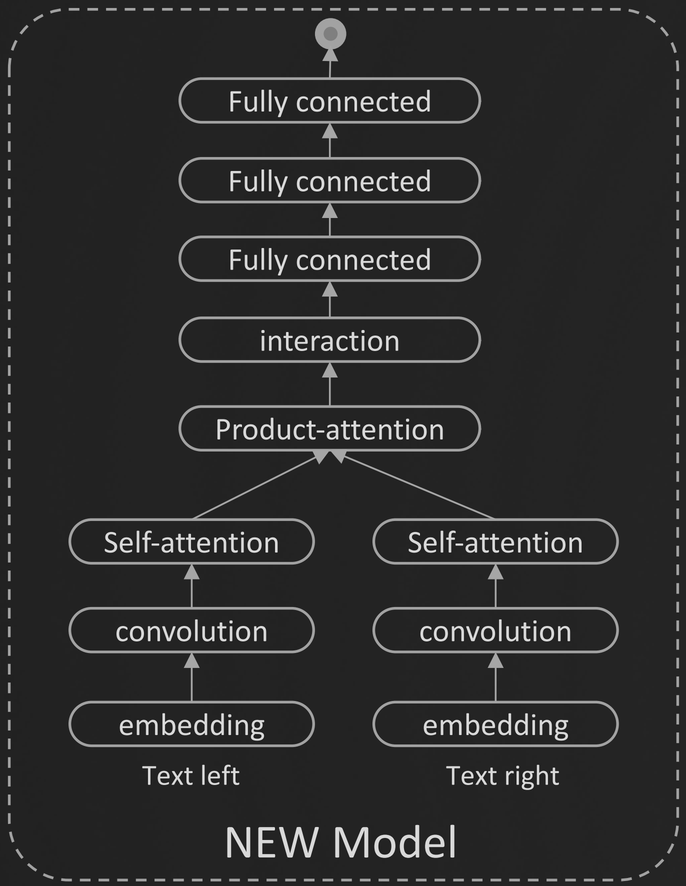
Question Answer

- ✓ Match-LSTM [Wang et al. 2016]
- ✓ BiDAF [Seo et al. 2016]
- ✓ AoA Reader [Cui et al. 2016]
- ✓ DrQA [Chen et al. 2017]
- ✓ R-Net [Wang et al. 2017]
- ✓ SAN [Liu et al. 2017]
- ✓ QANet [Yu et al. 2018]
- ✓ BERT [Jacob et al. 2018]
- ✓ …

Paraphrase Identification

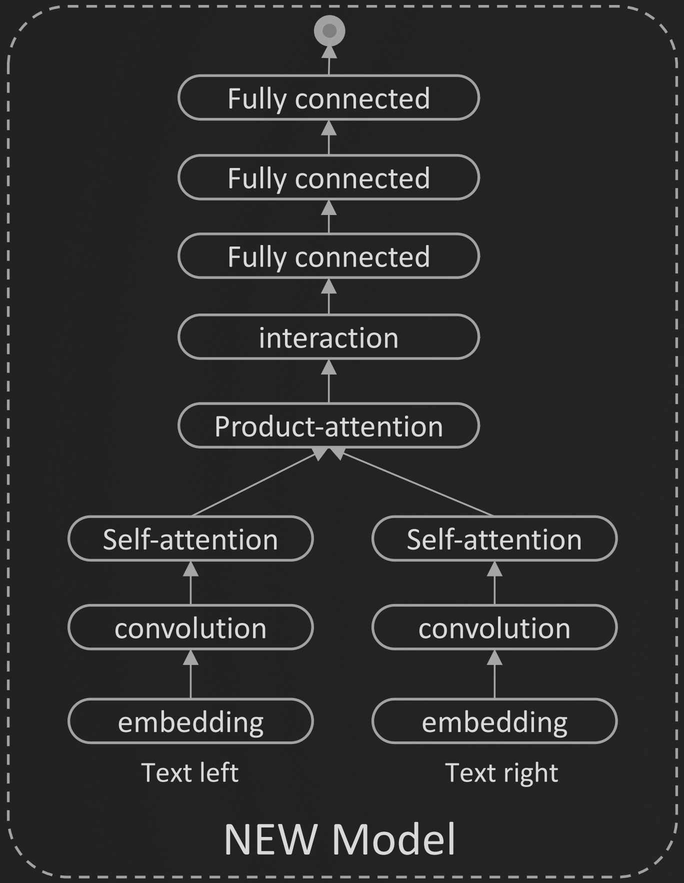
- ✓ DeepMatch [Lu et al. 2013]
- ✓ ARCI [Hu et al. 2014]
- ✓ ARCII [Hu et al. 2014]
- ✓ CNTN [Qiu et al. 2015]
- ✓ MatchPyramid [Pang et al. 2016]
- ✓ MV-LSTM [Wan et al. 2016a]
- ✓ Match-SRNN [Wan et al. 2016b]
- ✓ MIX [Chen et al. 2018]
- ✓ …

Text matching



- DSSM
- CDSSM
- DRMM
- K-NRM
- Duet
- Conv-KNRM
- PACRR
- DeepRank
- HiNT
- ...

Text matching



- DSSM
- CDSSM
- DRMM
- K-NRM
- Duet
- Conv-KNRM
- PACRR
- DeepRank
- HiNT
- ...

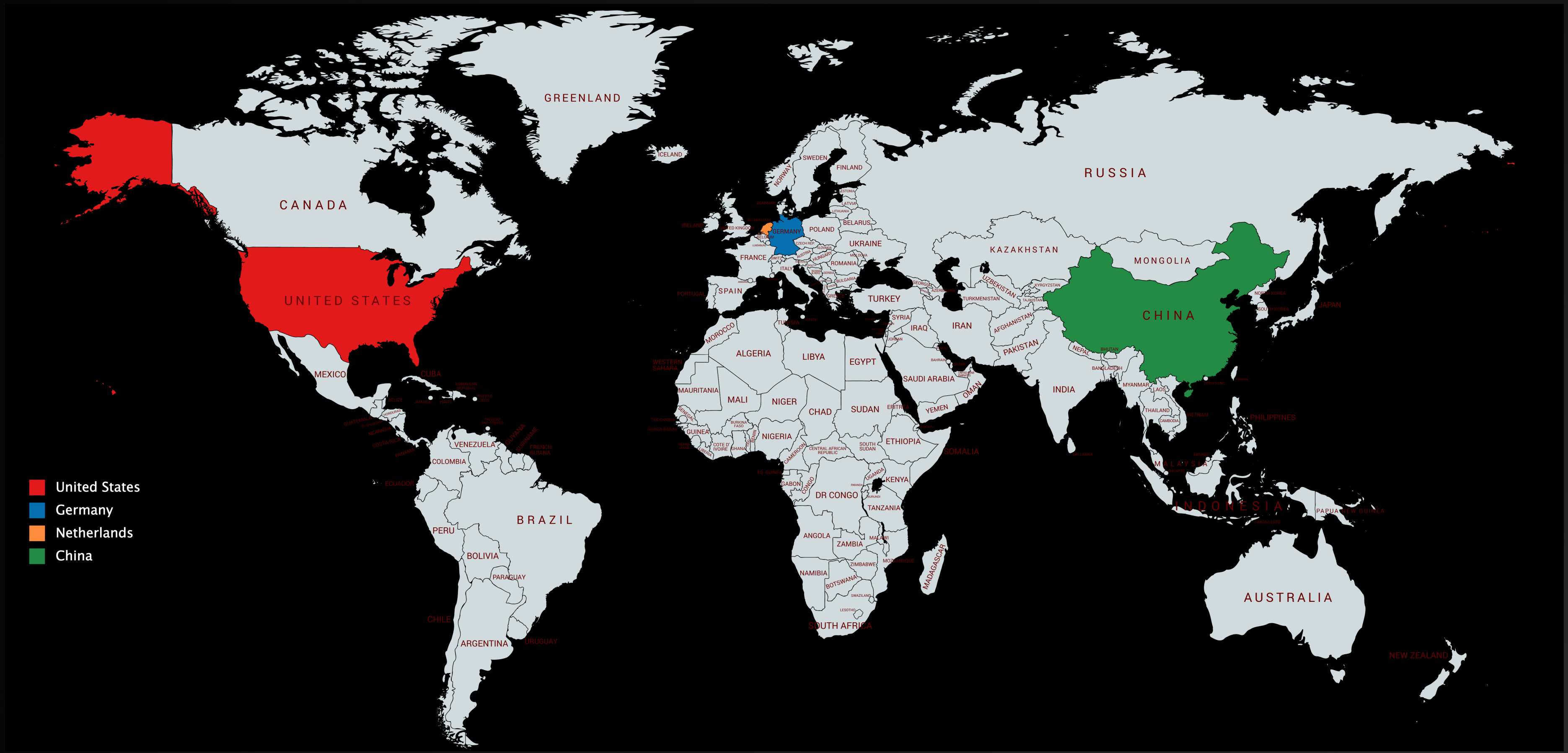
MatchZoo

MatchZoo is a toolkit aims to facilitate the **designing**, **comparing**, **optimizing**, and **deploying** of deep text matching models.

中国科学院网络数据科学与技术重点实验室
Key Laboratory of Network Data Science & Technology, CAS

Opening Source Toolkit & global cooperating

- Organizers: Yixing Fan; Jiafeng Guo; Yanyan Lan; Xueqi Cheng



MatchZoo

Raw Data → **PreProcessor** → **Model** → **Train & Test**

Data Preparation

1. Data cleaning
2. Batch modes:
3. Data generator

Model Construction

1. Representation-focused model
2. Interaction-focused model

Training and Evaluation

1. Objective functions:
   regression
   classification
   ranking
2. Metrics: MAP, NDCG …

Extended Keras Library

Basic Keras Ops:

1. Conv
2. LSTM
3. ……

Extended Ops:

1. 2DGRU
2. Term Gating
3. ……

MatchZoo

1.0 → 2.0

- Unified data processing API
- Simplified model configuration
- Easy to add new models
- Automatic parameter tuning
- Automatic model selection

MatchZoo

- data preprocess:
  - ✓ Tokenization Unit
  - ✓ Lower case Unit
  - ✓ Punctual Removal Unit
  - ✓ Stemming Unit
  - ✓ HistogramUnit
  - ✓ Digit Removal Unit
  - ✓ Stop Word Removal Unit
  - ✓ Word Hash Unit
  - ✓ Frequency Filter Unit
  - ✓ Vocabulary Unit

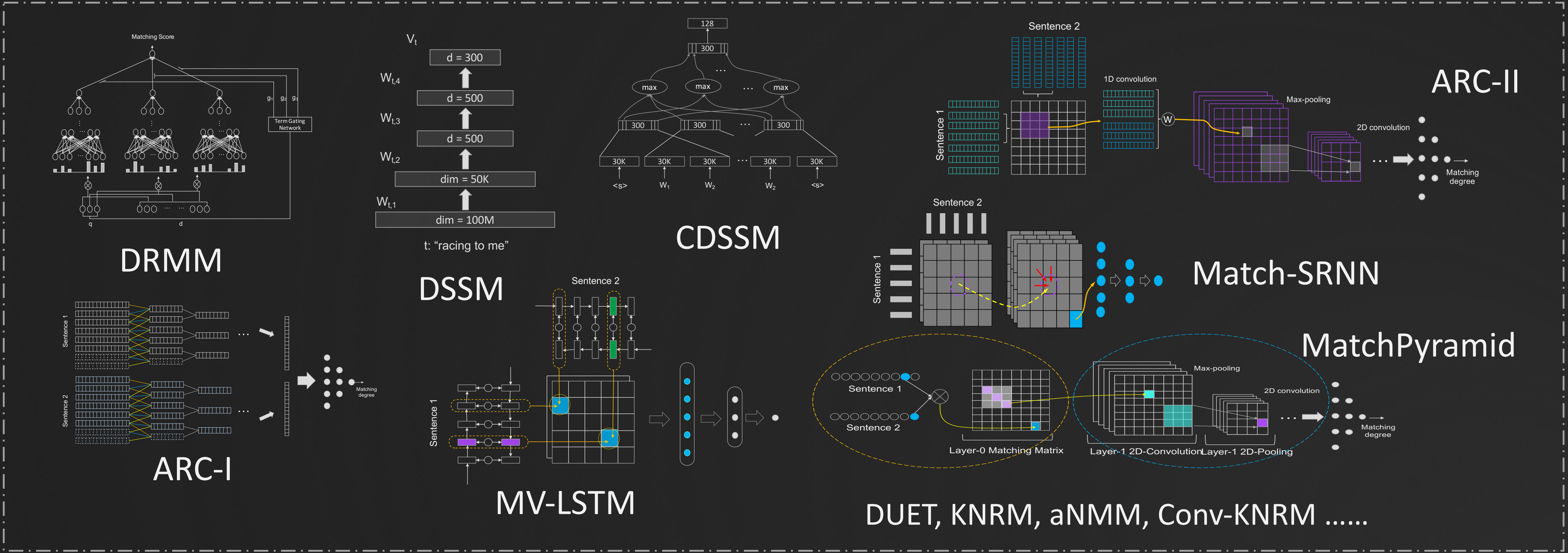
id_left	text_left
Q1	how are glacier caves formed?
Q2	How are the directions of the velocity and for...
Q5	how did apollo creed die
Q6	how long is the term for federal judges
Q7	how a beretta model 21 pistols magazines works

id_left	text_left
Q1	[6248, 3232, 23623, 26906, 18581, 0, 0, 0, 0, ...
Q2	[6248, 3232, 11296, 9779, 4231, 11296, 25020, ...
Q5	[6248, 8466, 5344, 22570, 26752, 0, 0, 0, 0, 0...
Q6	[6248, 18206, 6559, 11296, 12243, 22211, 11936...
Q7	[6248, 18788, 4030, 11359, 12567, 17504, 6486,...

Fruitful preprocessing unit to standardize data

MatchZoo

➢ Model Implementation:



A number of deep matching models have been implemented in the toolkit

MatchZoo

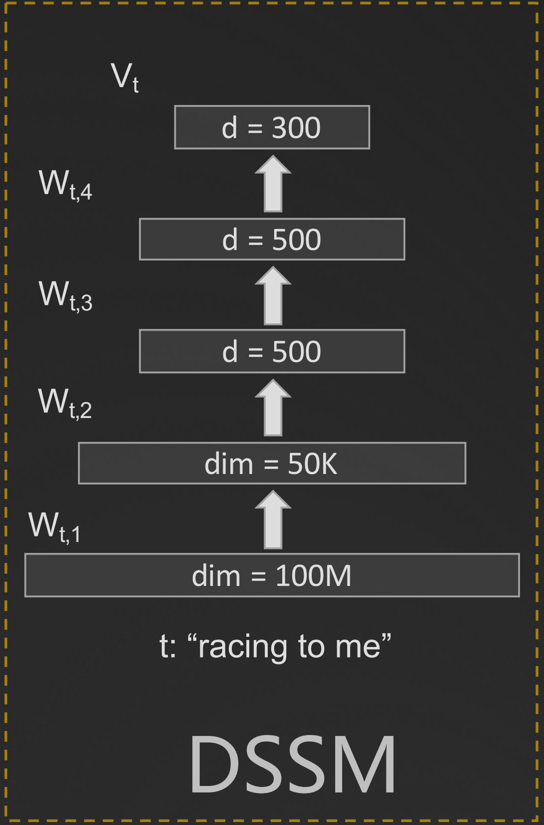
➢ Model Construction

```
import match as mz

train_data = mz.datasets.wiki_qa.load_data('train')
test_ data_= mz.dataset.wiki_qa.load_data('test')

preprocessor = mz.preprocessor.DSSMPreprocessor()
train_processed = preprocessor.fit_transform(train_data)
test_processed = preprocessor.transform(test_data)

dssm = mz.models.DSSM()
model.params['mlp_num_layers'] = 3
model.params['mlp_num_units'] = 300
model.params['mlp_num_fan_out'] = 128
model.params['mlp_activation_func'] = 'relu'
model.guess_and_fill_missing_params()
model.build()
model.compile()

history = model.fit(train_processed.unpack(), epochs=100)
result = model.evaluate(test_processed.unpack())
```

1. Data Process

2. Model Configuration

3. Train & Test



MatchZoo

Add New Model

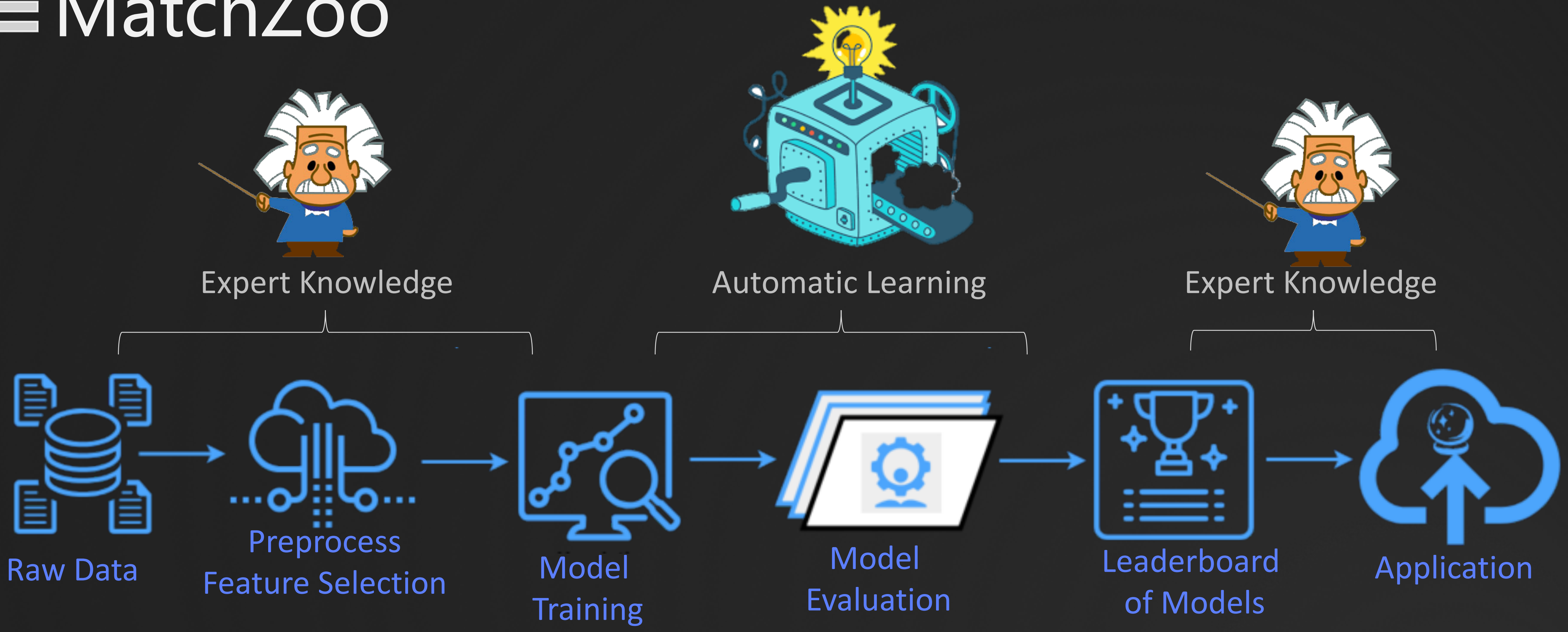
- matchzoo
  - __init__.py
  - auto
  - contrib
  - data_generator
  - data_pack
  - datasets
  - engine
  - layers
  - losses
  - metrics
  - models
    - __init__.py
    - arci.py
    - naive.py
  - preprocessors
  - processor_units
  - tasks
  - utils

```python
import keras
from matchzoo import engine

class NaiveModel(engine.BaseModel):
    def get_default_parsm(cls):
        params = super().get_default_params()
        params['param_1'] = 100
        ...
        return params
    def build(self):
        x_in = self._make_inputs()
        x = keras.layers.Dense(self._params['param_1'])(x_in)
        ...
        '''add more operations'''
        ...
        x_out = self._make_output_layer()(x)
        self._backend = keras.models.Model([x_in, x_out])
```

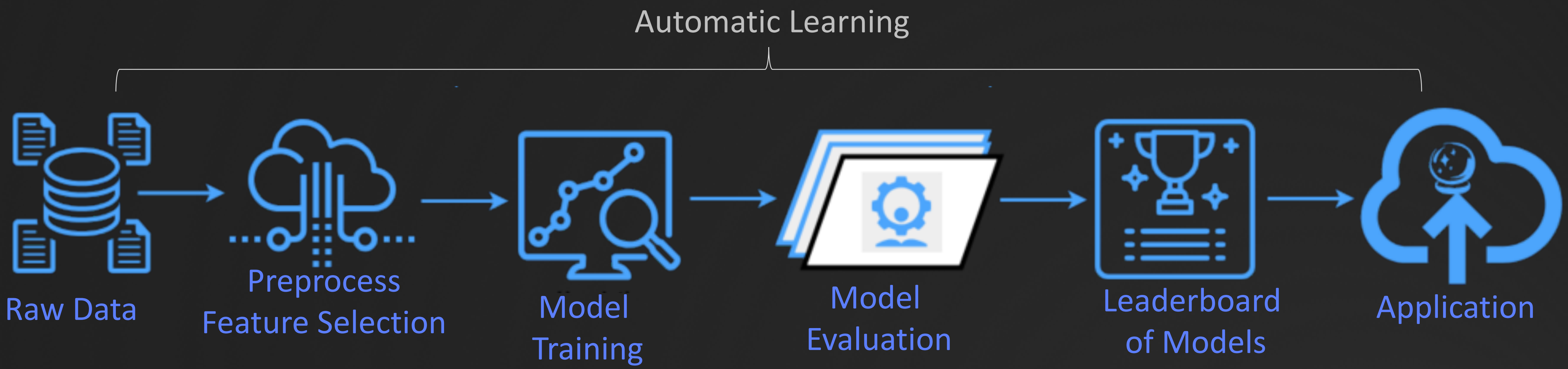
MatchZoo

Tuning machine learning **hyperparameters** is a **tedious** yet crucial task, as the performance of an algorithm can be highly dependent on the choice of **hyperparameters**.

MatchZoo


Expert Knowledge
Automatic Learning
Expert Knowledge
Raw Data
Preprocess
Feature Selection
Model
Training
Model
Evaluation
Leaderboard
of Models
Application

MatchZoo




```python
model, train_data, _ = mz.auto.prepare(
    model=mz.models.DSSM(),
    data_pack=data
)
result = model.fit(train_data)
```

```python
tuner = mz.auto.tuner.Tuner(
    params=params,
    train_data=train,
    test_data=dev
)
results = tuner.tune()
```

From matchzoo.auto import prepare, tuner

MatchZoo

```
import matchzoo as mz
models = [
    mz.models.DSSM,
    mz.models.CDSSM,
    mz.models.DUET,
    mz.models.MatchPyramid,
    mz.models.KNRM
]
task = mz.tasks.Ranking()
outputs = {}
for model in models:
    m = model()
    m.params['task'] = task
    m, train_data, _ = mz.auto.prepare(
        model = m,
        data_pack = data
    )
    result = tuner.tune(
        params = m.params,
        train_data = train_data,
        test_data = test_data
    )
    outputs[model] = result['best']

print(outputs)
```

Models Initialization

⇩

Task Definition

⇩

Data Preparing

⇩

Parameter Tuning

⇩

Result Recording

Automatic machine learning

MatchZoo

https://github.com/NTMC-Community/MatchZoo

Unwatch 144 | Star 2,065 | Fork 563

MatchZoo

A big welcome to join us to develop the text matching toolkit!

Thank You
&
Question

Name : Yixing Fan
Email : fanyixing@ict.ac.cn

Reference

1. [Huang et al. 2013.] Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. CIKM 2013
2. [Ye et al. 2014] A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. CIKM 2014
3. [Guo et al. 2016] A deep relevance matching model for ad-hoc retrieval. CIKM 2016
4. [Mitra et al. 2017] Learning to Match Using Local and Distributed Representations of Text for Web Search. WWW 2017.
5. [Xiong et al. 2017] End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017.
6. [Hui et al. 2017] A Position-Aware Deep Model for Relevance Matching in Information Retrieval. Conference'17
7. [Pang et al. 2017] DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval. CIKM 2017.
8. [Dai et al. 2018] Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. WSDM 2018.
9. [Fan et al. 2018] Modeling Diverse Relevance Patterns in Ad-hoc Retrieval. SIGIR 2018.
10. [Seo et al. 2016] Bidirectional Attention Flow for Machine Comprehension
11. [Cui et al. 2016] Attention-over-Attention Neural Networks for Reading Comprehension.
12. [Chen et al. 2016] Reading Wikipedia to Answer Open-Domain Questions.
13. [Wang et al. 2017] R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS.
14. [Liu et al. 2017] Stochastic Answer Networks for Machine Reading Comprehension
15. [Yu et al. 2018] QANET: COMBINING LOCAL CONVOLUTION WITH GLOBAL SELF-ATTENTION FOR READING COMPREHENSION
16. [Jacob et al. 2018] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
17. [Lu et al. 2013] A Deep Architecture for Matching Short Texts. NIPS 2013.
18. [Hu et al. 2014] convolutional-neural-network-architectures-for-matching-natural-language-sentences. NIPS 2014.
19. [Qiu et al. 2015] Convolutional Neural Tensor Network Architecture for Community-Based Question Answering. IJCAI 2015
20. [Pang et al. 2016] Text Matching as Image Recognition. AAAI 2016.
21. [Wan et al. 2016a] Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. AAAI 2016.
22. [Wan et al. 2016b] Match-SRNN- Modeling the Recursive Matching Structure with Spatial RNN. IJCAI 2016
23. [Chen et al. 2018] MIX: Multi-Channel Information Crossing for Text Matching, KDD 2018