nlpy Documentation

Release 1.0.0

sunyan

Feb 28, 2019

Contents

CHAPTER 1

Embedding

- `CharEmbedding`:
- `PositionEmbedding`:
- `WordEmbedding`:

CHAPTER 2

Text classification

2.1 Available models

All the following models includes Dropout, Pooling and Dense layers with hyperparameters tuned for reasonable performance across standard text classification tasks. If necessary, they are good basis for further performance tuning.

- `text_cnn`:
- `text_rnn`:
- `attention_rnn`:
- `text_rcnn`:
- `text_han`:

2.2 Examples

Choose a pre-trained word embedding by setting the `embedding_type` and the corresponding embedding dimensions. Set embedding_type=None to initialize the word embeddings randomly (but make sure to set `trainable_embeddings=True` so you actually train the embeddings).

2.2.1 FastText

Several pre-trained FastText embeddings are included. For now, we only have the word embeddings and not the n-gram features. All embedding have 300 dimensions.

- English Vectors: e.g. `fasttext.wn.1M.300d`, check out all avaiable embeddings
- Multilang Vectors: in the format `fasttext.cc.LANG_CODE` e.g. `fasttext.cc.en`
- Wikipedia Vectors: in the format `fasttext.wiki.LANG_CODE` e.g. `fasttext.wiki.en`.en

##Dataset

2.3 segment

1.

CHAPTER 3

Reference

1. keras-text
2. keras_contrib
3. talos
4. delft
5. Keras-Project-Template
6. text-classification-keras
7. Practical Text Classification With Python and Keras
8. NLPMetrics

CHAPTER 4

Dataset and Model

4.1 Reading Comprehension

4.1.1 Dataset

- HistoryQA: Joseon History Question Answering Dataset (SQuAD Style)
- KorQuAD: KorQuAD Machine Reading Comprehension . Wikipedia . Stanford Question Answering Dataset(SQuAD) v1.0 .
- SQuAD: **S**tanford **Qu**estion **A**nswering **D**ataset is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

4.1.2 Model

- BiDAF: Birectional Attention Flow for Machine Comprehension + `No Answer`
- DrQA: Reading Wikipedia to Answer Open-Domain Questions
- DocQA: Simple and Effective Multi-Paragraph Reading Comprehension + `No Answer`
- QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

4.2 Semantic Parsing

4.2.1 Dataset

- WikiSQL: A large crowd-sourced dataset for developing natural language interfaces for relational databases. WikiSQL is the dataset released along with our work Seq2SQL: Generating Structured Queries from Natural

Language using Reinforcement Learning.

4.2.2 Model

- SQLNet: SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning

4.3 Sequence Classification

4.3.1 Dataset

4.3.2 Model

- A Structured Self-attentive Sentence Embedding
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

4.4 Token Classification

4.4.1 Dataset

4.4.2 Model

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

CHAPTER 5

Pretrained Vector

- List on DataServer

5.1 English

- `Counter Fitting`: Counter-fitting Word Vectors to Linguistic Constraints
  - counter_fitted_glove.300d.txt
- `Cove`: Learned in Translation: Contextualized Word Vectors (McCann et. al. 2017)
  - wmtlstm-b142a7f2.pth
- `fastText`: Enriching Word Vectors with Subword Information
  - fasttext.wiki.en.300d.txt
- `GloVe`: GloVe: Global Vectors for Word Representation
  - glove.6B.50d.txt
  - glove.6B.100d.txt
  - glove.6B.200d.txt
  - glove.6B.300d.txt
  - glove.840B.300d.txt
- `ELMo`: Deep contextualized word representations
  - elmo_2x4096_512_2048cnn_2xhighway_weights.hdf5
  - elmo_2x4096_512_2048cnn_2xhighway_options
- `Word2Vec`: Distributed Representations of Words and Phrases and their Compositionality
  - GoogleNews-vectors-negative300.txt