rockingdingo / **deepnlp**

Deep Learning NLP Pipeline implemented on Tensorflow

40 commits | **4** branches | **1** release | **1** contributor | MIT

Branch: **master** ▾ | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download** ▾

rockingdingo Update README.md — Latest commit `9a5717e` on 5 May

deepnlp	update	7 months ago
docs	add python3 compatibility support	a year ago
test	update to 0.1.6 release	9 months ago
LICENSE	update to 0.1.6 release	9 months ago
MANIFEST.in	update to 0.1.6 release	9 months ago
README.md	Update README.md	7 months ago
RELEASE.md	update to 0.1.6 release	9 months ago
setup.py	update to 0.1.6 release	9 months ago

README.md

Deep Learning NLP Pipeline implemented on Tensorflow. Following the 'simplicity' rule, this project aims to use the deep learning library of Tensorflow to implement new NLP pipeline. You can extend the project to train models with your own corpus/languages. Pretrained models of Chinese corpus are distributed. Free RESTful NLP API are also provided. Visit http://www.deepnlp.org/api/v1.0/pipeline for details.

Brief Introduction

Modules

- NLP Pipeline Modules:
  - Word Segmentation/Tokenization
  - Part-of-speech (POS)
  - Named-entity-recognition(NER)
  - textsum: automatic summarization Seq2Seq-Attention models
  - textrank: extract the most important sentences
  - textcnn: document classification
  - Web API: Free Tensorflow empowered web API
  - Planed: Parsing, Automatic Summarization
- Algorithm(Closely following the state-of-Art)
  - Word Segmentation: Linear Chain CRF(conditional-random-field), based on python CRF++ module
  - POS: LSTM/BI-LSTM network, based on Tensorflow
  - NER: LSTM/BI-LSTM/LSTM-CRF network, based on Tensorflow
  - Textsum: Seq2Seq with attention mechanism
  - Texncnn: CNN
- Pre-trained Model
  - Chinese: Segmentation, POS, NER (1998 china daily corpus)
  - English: POS (brown corpus)
  - For your Specific Language, you can easily use the script to train model with the corpus of your language choice.

Installation

- Requirements
  - CRF++ (>=0.54)
  - Tensorflow(1.0) This project is up to date with the latest tensorflow release. For tensorflow (<=0.12.0), use deepnlp <=0.1.5 version. See RELEASE.md for more details
- Pip

```
# linux, run the script:
pip install deepnlp
```

Due to pkg size restriction, english pos model, ner model files are not distributed on pypi You can download the pre-trained model files from github and put in your installation directory .../site-packages/.../deepnlp/... model files: ../pos/ckpt/en/pos.ckpt ; ../ner/ckpt/zh/ner.ckpt

- Source Distribution, e.g. deepnlp-0.1.6.tar.gz: https://pypi.python.org/pypi/deepnlp

```
# linux, run the script:
tar zxvf deepnlp-0.1.6.tar.gz
cd deepnlp-0.1.6
python setup.py install
```

- Running Examples

```
# ./deepnlp/test folder
cd test
python test_pos_en.py
python test_segmenter.py
```

```
    python test_pos_zh.py
    python test_api_v1_module.py
    python test_api_v1_pipeline.py
```

Tutorial

Set Coding

设置编码 For python2, the default coding is ascii not unicode, use **future** module to make it compatible with python3

```
#coding=utf-8
from __future__ import unicode_literals # compatible with python3 unicode
```

Download pretrained models

下载预训练模型 If you install deepnlp via pip, the pre-trained models are not distributed due to size restriction. You can download full models for 'Segment', 'POS' en and zh, 'NER' zh, 'Textsum' by calling the download function.

```
import deepnlp
# Download all the modules
deepnlp.download()

# Download only specific module
deepnlp.download('segment')
deepnlp.download('pos')
deepnlp.download('ner')
deepnlp.download('textsum')
```

Segmentation

分词模块

```
#coding=utf-8
from __future__ import unicode_literals

from deepnlp import segmenter

text = "我刚刚在浙江卫视看了电视剧老九门，觉得陈伟霆很帅"
segList = segmenter.seg(text)
text_seg = " ".join(segList)

print (text.encode('utf-8'))
print (text_seg.encode('utf-8'))

#Results
#我 刚刚 在 浙江卫视 看 了 电视剧 老九门 ， 觉得 陈伟霆 很 帅
```

POS

词性标注

```
#coding:utf-8
from __future__ import unicode_literals

import deepnlp
```

```
deepnlp.download('pos')

## English Model
from deepnlp import pos_tagger
tagger = pos_tagger.load_model(lang = 'en')  # Loading English model, lang code 'en', English Model Brown C

text = "I want to see a funny movie"
words = text.split(" ")     # unicode
print (" ".join(words).encode('utf-8'))

tagging = tagger.predict(words)
for (w,t) in tagging:
    str = w + "/" + t
    print (str.encode('utf-8'))

#Results
#I/nn want/vb to/to see/vb a/at funny/jj movie/nn

## Chinese Model
from deepnlp import segmenter
from deepnlp import pos_tagger
tagger = pos_tagger.load_model(lang = 'zh') # Loading Chinese model, lang code 'zh', China Daily Corpus

text = "我爱吃北京烤鸭"
words = segmenter.seg(text) # words in unicode coding
print (" ".join(words).encode('utf-8'))

tagging = tagger.predict(words)  # input: unicode coding
for (w,t) in tagging:
    str = w + "/" + t
    print (str.encode('utf-8'))

#Results
#我/r 爱/v 吃/v 北京/ns 烤鸭/n
```

NER

命名实体识别

```
#coding:utf-8
from __future__ import unicode_literals

# Download pretrained NER model
import deepnlp
deepnlp.download('ner')

from deepnlp import segmenter
from deepnlp import ner_tagger
tagger = ner_tagger.load_model(lang = 'zh') # Loading Chinese NER model

text = "我爱吃北京烤鸭"
words = segmenter.seg(text)
print (" ".join(words).encode('utf-8'))

tagging = tagger.predict(words)
for (w,t) in tagging:
    str = w + "/" + t
    print (str.encode('utf-8'))

#Results
#我/nt 爱/nt 吃/nt 北京/p 烤鸭/nt
```

Pipeline

```
#coding:utf-8
from __future__ import unicode_literals

from deepnlp import pipeline
p = pipeline.load_model('zh')

#Segmentation
text = "我爱吃北京烤鸭"
res = p.analyze(text)

print (res[0].encode('utf-8'))
print (res[1].encode('utf-8'))
print (res[2].encode('utf-8'))

words = p.segment(text)
pos_tagging = p.tag_pos(words)
ner_tagging = p.tag_ner(words)

print (pos_tagging.encode('utf-8'))
print (ner_tagging.encode('utf-8'))
```

## Textsum

自动文摘

See details: README

Textrank

重要句子抽取

See details: README

TextCNN (WIP)

文档分类

Train your model

自己训练模型 ###Segment model See instructions: README

###POS model See instructions: README

###NER model See instructions: README

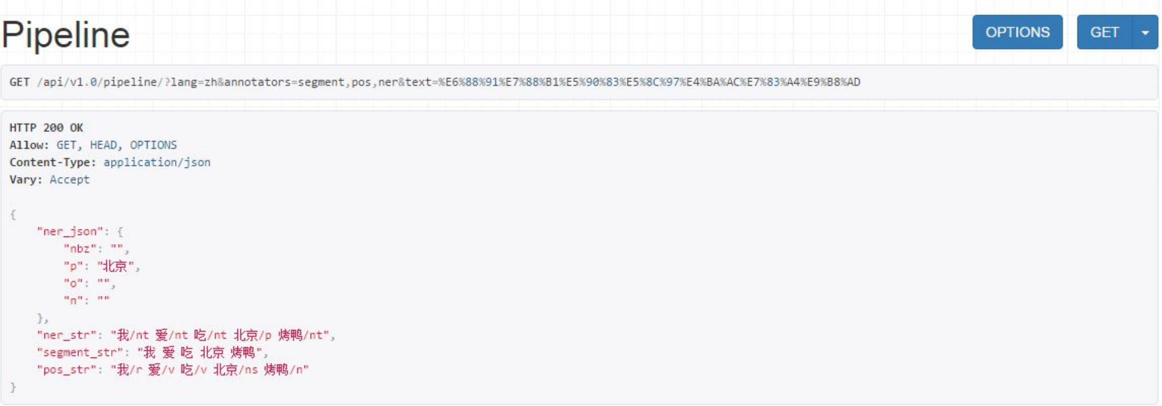
###Textsum model See instructions: README

Web API Service

www.deepnlp.org provides free web API service for common NLP modules of sentences and paragraphs. The APIs are RESTful and based on pre-trained tensorflow models. Chinese language is now supported.

- RESTful API
  - Segmentation: http://www.deepnlp.org/api/v1.0/segment/?lang=zh&text=我爱吃北京烤鸭
  - POS: http://www.deepnlp.org/api/v1.0/pos/?lang=zh&text=我爱吃北京烤鸭
  - NER: http://www.deepnlp.org/api/v1.0/ner/?lang=zh&text=我爱吃北京烤鸭
  - Pipeline: http://www.deepnlp.org/api/v1.0/pipeline/?lang=zh&annotators=segment,pos,ner&text=我爱吃北京烤鸭

Testing API from Browser, Need to log in first



```
GET /api/v1.0/pipeline/?lang=zh&annotators=segment,pos,ner&text=%E6%88%91%E7%88%B1%E5%90%83%E5%8C%97%E4%BA%AC%E7%83%A4%E9%B8%AD
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "ner_json": {
        "nbz": "",
        "p": "北京",
        "o": "",
        "n": ""
    },
    "ner_str": "我/nt 爱/nt 吃/nt 北京/p 烤鸭/nt",
    "segment_str": "我 爱 吃 北京 烤鸭",
    "pos_str": "我/r 爱/v 吃/v 北京/ns 烤鸭/n"
}
```

Calling API from python

See ./deepnlp/test/test_api_v1_module.py for more details.

```
#coding:utf-8
from __future__ import unicode_literals

import json, requests, sys, os
if (sys.version_info>(3,0)): from urllib.parse import quote
else : from urllib import quote

from deepnlp import api_service
login = api_service.init()          # registration, if failed, load default empty login {} with limited acc
conn = api_service.connect(login)   # save the connection with login cookies

# Sample URL
# http://www.deepnlp.org/api/v1.0/pipeline/?lang=zh&annotators=segment,pos,ner&text=我爱吃上海小笼包

# Define text and language
text = ("我爱吃上海小笼包").encode("utf-8")  # convert text from unicode to utf-8 bytes

# Set up URL for POS tagging
url_pos = 'http://www.deepnlp.org/api/v1.0/pos/?"+ "lang=" + quote('zh') + "&text=" + quote(text)
web = requests.get(url_pos, cookies = conn)
tuples = json.loads(web.text)
print (tuples['pos_str'].encode('utf-8'))    # POS json {'pos_str', 'w1/t1 w2/t2'} return string
```

中文简介

deepnlp项目是基于Tensorflow平台的一个python版本的NLP套装, 目的在于将Tensorflow深度学习平台上的模块，结合最新的一些算法，提供NLP基础模块的支持，并支持其他更加复杂的任务的拓展，如生成式文摘等等。

- NLP 套装模块
  - 分词 Word Segmentation/Tokenization
  - 词性标注 Part-of-speech (POS)
  - 命名实体识别 Named-entity-recognition(NER)
  - 自动生成式文摘 Textsum (Seq2Seq-Attention)
  - 关键句子抽取 Textrank
  - 文本分类 Textcnn (WIP)
  - 可调用 Web Restful API
  - 计划中: 句法分析 Parsing

- 算法实现
  - 分词: 线性链条件随机场 Linear Chain CRF, 基于CRF++包来实现
  - 词性标注: 单向LSTM/ 双向BI-LSTM, 基于Tensorflow实现
  - 命名实体识别: 单向LSTM/ 双向BI-LSTM/ LSTM-CRF 结合网络, 基于Tensorflow实现
- 预训练模型
  - 中文: 基于人民日报语料和微博混合语料: 分词, 词性标注, 实体识别

API 服务

http://www.deepnlp.org 出于技术交流的目的, 提供免费API接口供文本和篇章进行深度学习NLP的分析, 简单注册后就可以使用。 API符合RESTful风格, 内部是基于tensorflow预先训练好的深度学习模型。具体使用方法请参考博客: http://www.deepnlp.org/blog/tutorial-deepnlp-api/

API目前提供以下模块支持：

- 分词: http://www.deepnlp.org/api/v1.0/segment/?lang=zh&text=我爱吃北京烤鸭
- 词性标注: http://www.deepnlp.org/api/v1.0/pos/?lang=zh&text=我爱吃北京烤鸭
- 命名实体识别: http://www.deepnlp.org/api/v1.0/ner/?lang=zh&text=我爱吃北京烤鸭
- Pipeline: http://www.deepnlp.org/api/v1.0/pipeline/?lang=zh&annotators=segment,pos,ner&text=我爱吃北京烤鸭

安装说明

- 需要
  - CRF++ (>=0.54) 可以从 https://taku910.github.io/crfpp/ 下载安装
  - Tensorflow(1.0) 这个项目的Tensorflow函数会根据最新Release更新，目前支持Tensorflow 1.0版本，对于老版本的Tensorflow(<=0.12.0), 请使用 deepnlp <=0.1.5版本, 更多信息请查看 RELEASE.md
- Pip 安装

```
pip install deepnlp
```

- 从源码安装, 下载deepnlp-0.1.6.tar.gz文件: https://pypi.python.org/pypi/deepnlp

```
# linux, run the script:
tar zxvf deepnlp-0.1.6.tar.gz
cd deepnlp-0.1.6
python setup.py install
```

Reference

- CRF++ package: https://taku910.github.io/crfpp/#download
- Tensorflow: https://www.tensorflow.org/
- Word Segmentation Using CRF++ Blog: http://www.52nlp.cn/%E4%B8%AD%E6%96%87%E5%88%86%E8%AF%8D%E5%85%A5%E9%97%A8%E4%B9%8B%E5%AD%97%E6%A0%87%E6%B3%A8%E6%B3%954
- Blogs http://www.deepnlp.org/blog/