EBook Gratis

APRENDIZAJE

nlp

Free unaffiliated eBook created from
Stack Overflow contributors.

#nlp

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: nlp

It is an unofficial and free nlp ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official nlp.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con nlp

Observaciones

Esta sección proporciona una descripción general de qué es nlp y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de nlp y vincular a los temas relacionados. Dado que la Documentación para nlp es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Stanford CoreNLP

Stanford CoreNLP es un popular kit de herramientas de procesamiento de lenguaje natural que admite muchas tareas básicas de PNL.

Para descargar e instalar el programa, descargue un paquete de lanzamiento e incluya los archivos `*.jar` necesarios en su ruta de clases, o agregue la dependencia de Maven central. Vea la página de descarga para más detalles. Por ejemplo:

```
curl http://nlp.stanford.edu/software/stanford-corenlp-full-2015-12-09.zip -o corenlp.zip
unzip corenlp.zip
cd corenlp
export CLASSPATH="$CLASSPATH:`pwd`/*
```

Hay tres formas compatibles de ejecutar las herramientas CoreNLP: (1) usando la API base completamente personalizable , (2) usando la API Simple CoreNLP , o (3) usando el servidor CoreNLP . Un ejemplo de uso simple para cada uno se da a continuación. Como un caso de uso motivador, estos ejemplos serán para predecir el análisis sintáctico de una oración.

1. **API CoreNLP**

```
public class CoreNLPDemo {
  public static void main(String[] args) {

    // 1. Set up a CoreNLP pipeline. This should be done once per type of annotation,
    //    as it's fairly slow to initialize.
    // creates a StanfordCoreNLP object, with POS tagging, lemmatization, NER, parsing,
and coreference resolution
    Properties props = new Properties();
    props.setProperty("annotators", "tokenize, ssplit, parse");
    StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

    // 2. Run the pipeline on some text.
    // read some text in the text variable
    String text = "the quick brown fox jumped over the lazy dog"; // Add your text here!
    // create an empty Annotation just with the given text
```

```
    Annotation document = new Annotation(text);
    // run all Annotators on this text
    pipeline.annotate(document);

    // 3. Read off the result
    // Get the list of sentences in the document
    List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);
    for (CoreMap sentence : sentences) {
      // Get the parse tree for each sentence
      Tree parseTree = sentence.get(TreeAnnotations.TreeAnnotation.class);
      // Do something interesting with the parse tree!
      System.out.println(parseTree);
    }

  }
}
```

2. **CoreNLP simple**

```
public class CoreNLPDemo {
  public static void main(String[] args) {
    String text = "The quick brown fox jumped over the lazy dog");  // your text here!
    Document document = new Document(text);  // implicitly runs tokenizer
    for (Sentence sentence : document.sentences()) {
      Tree parseTree = sentence.parse();  // implicitly runs parser
      // Do something with your parse tree!
      System.out.println(parseTree);
    }
  }
}
```

3. **Servidor CoreNLP**

   Inicie el servidor con lo siguiente (configurando su classpath apropiadamente):

```
java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer [port] [timeout]
```

   Obtenga una salida con formato JSON para un conjunto dado de anotadores, e imprímala en una salida estándar:

```
 wget --post-data 'The quick brown fox jumped over the lazy dog.'
'localhost:9000/?properties={"annotators":"tokenize,ssplit,parse","outputFormat":"json"}'
-O -
```

   Para obtener nuestro árbol de análisis desde el JSON, podemos navegar el JSON a las `sentences[i].parse` .

Lea Empezando con nlp en línea: https://riptutorial.com/es/nlp/topic/2613/empezando-con-nlp

Capítulo 2: Detección de límites de oraciones en Python

Examples

Con Stanford CoreNLP, desde Python

Primero necesita ejecutar un servidor Stanford CoreNLP :

```
java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 9000 -timeout 50000
```

Aquí hay un fragmento de código que muestra cómo pasar datos al servidor Stanford CoreNLP, usando el paquete `pycorenlp` .

```
from pycorenlp import StanfordCoreNLP
import pprint

if __name__ == '__main__':
    nlp = StanfordCoreNLP('http://localhost:9000')
    fp = open("long_text.txt")
    text = fp.read()
    output = nlp.annotate(text, properties={
        'annotators': 'tokenize,ssplit,pos,depparse,parse',
        'outputFormat': 'json'
    })
    pp = pprint.PrettyPrinter(indent=4)
    pp.pprint(output)
```

Con python-ucto

Ucto es un tokeniser basado en reglas para múltiples idiomas. Hace la detección del límite de la oración también. Aunque está escrito en C ++, hay un Python -ucto de enlace de Python para interactuar con él.

```
import ucto

#Set a file to use as tokeniser rules, this one is for English, other languages are available
too:
settingsfile = "/usr/local/etc/ucto/tokconfig-en"

#Initialise the tokeniser, options are passed as keyword arguments, defaults:
#   lowercase=False,uppercase=False,sentenceperlineinput=False,
#   sentenceperlineoutput=False,
#   sentencedetection=True, paragraphdetection=True, quotedetection=False,
#   debug=False
tokenizer = ucto.Tokenizer(settingsfile)

tokenizer.process("This is a sentence. This is another sentence. More sentences are better!")

for sentence in tokenizer.sentences():
```

```
    print(sentence)
```

Usando la biblioteca NLTK

Puede encontrar más información sobre el tokenizador de nivel de oración Python Natural Language Toolkit (NLTK) en su wiki .

Desde su línea de comando:

```
$ python
>>> import nltk
>>> sent_tokenizer = nltk.tokenize.PunktSentenceTokenizer()
>>> text = "This is a sentence. This is another sentence. More sentences are better!"
>>> sent_tokenizer.tokenize(text)
Out[4]:
['This is a sentence.',
 'This is another sentence.',
 'More sentences are better!']
```

Lea Detección de límites de oraciones en Python en línea:
https://riptutorial.com/es/nlp/topic/3833/deteccion-de-limites-de-oraciones-en-python

Capítulo 3: N-GRAMS

Introducción

Los N-GRAM son modelos estadísticos que predicen la siguiente palabra en la oración usando las palabras n-1 anteriores. Este tipo de modelos estadísticos que usan secuencias de palabras también se denominan modelos de lenguaje. Por ejemplo, tenemos una frase "No puedo leer sin leer _____", podemos decir que la siguiente palabra más probable sería "anteojos". N-GRAMS predice la siguiente palabra en la secuencia usando la probabilidad condicional de la siguiente palabra. El modelo N-GRAM es muy esencial en el procesamiento del habla y el lenguaje.

Sintaxis

- La probabilidad condicional de la siguiente palabra más probable puede obtenerse utilizando un gran corpus (Colección administrada de datos de texto o habla), todo se trata de contar cosas (palabras) del corpus. El objetivo es encontrar P (w | h), que es la probabilidad de la siguiente palabra en la secuencia dada cierta historia h.
- El concepto del modelo N-GRAM es que, en lugar de calcular la probabilidad de una palabra dada su historia completa, acorta la historia a pocas palabras anteriores. Cuando usamos una sola palabra anterior para predecir la siguiente palabra, se llama modelo Bi-GRAM. Por ejemplo, tenemos P (gafas | lectura), la probabilidad de la palabra "gafas" dada la palabra anterior "lectura" se calcula como: (Consulte el ejemplo)

Observaciones

Los modelos N-GRAM son muy importantes cuando tenemos que identificar palabras en una entrada ruidosa y ambigua. Los modelos N-GRAM se utilizan en:

- Reconocimiento de voz
- Reconocimiento de escritura a mano
- Corrección de hechizos
- Máquina traductora
- muchas otras aplicaciones

Puedes leer más sobre los modelos N-GRAM en:

- Libro de procesamiento del habla y lenguaje de Daniel Jurafsky y James H. Martin

Examples

Calcular la probabilidad condicional

P (*gafas* | *lectura*) = Cuenta (*gafas de lectura*) / Cuenta (*lectura*)

Contamos las secuencias de `reading glasses` y `glasses` de corpus y calculamos la probabilidad.

Lea N-GRAMS en línea: https://riptutorial.com/es/nlp/topic/8851/n-grams

Capítulo 4: OpenNLP

Sintaxis

- opennlp SentenciaDetector ./en-sent.bin <./input.txt> output.txt
- Inicialice el Detector de tiempo de una manera como esta: Detector de tiempo de la frase. Detector de sentencia = nuevo Detector de la oración de ME (modelo);
- Use el método 'sentDetect' para obtener oraciones como esta: oraciones de cadena [] = sentenceDetector.sentDetect ("cadena de información");

Observaciones

descargar modelos (como en-sent.bin) desde el siguiente enlace

Examples

Detección de oraciones usando openNLP usando CLI y API de Java

utilizando CLI:

```
$ opennlp SentenceDetector ./en-sent.bin < ./input.txt > output.txt
```

utilizando API:

```
import static java.nio.file.Files.readAllBytes;
import static java.nio.file.Paths.get;

import java.io.IOException;
import java.util.Objects;

public class FileUtils {
/**
 * Get file data as string
 *
 * @param fileName
 * @return
 */
    public static String getFileDataAsString(String fileName) {
        Objects.nonNull(fileName);
        try {
            String data = new String(readAllBytes(get(fileName)));
            return data;
        } catch (IOException e) {
            System.out.println(e.getMessage());
            return null;
        }
    }
}
```

clase sentecedetectorutil:

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.Objects;

import opennlp.tools.sentdetect.SentenceDetectorME;
import opennlp.tools.sentdetect.SentenceModel;

public class SentenceDetectorUtil {
    private SentenceModel model = null;
    SentenceDetectorME sentenceDetector = null;

    public SentenceDetectorUtil(String modelFile) {
        Objects.nonNull(modelFile);
        initSentenceModel(modelFile);
        initSentenceDetectorME();
    }

    private void initSentenceDetectorME() {
        sentenceDetector = new SentenceDetectorME(model);
    }

    private SentenceModel initSentenceModel(String file) {
        InputStream modelIn;
        try {
            modelIn = new FileInputStream(file);
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
            return null;
        }

        try {
            model = new SentenceModel(modelIn);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (modelIn != null) {
                try {
                    modelIn.close();
                } catch (IOException e) {
                }
            }
        }
        return model;
    }

    public String[] getSentencesFromFile(String inputFile) {
        String data = FileUtils.getFileDataAsString(inputFile);
        return sentenceDetector.sentDetect(data);
    }

    public String[] getSentences(String data) {
        return sentenceDetector.sentDetect(data);
    }

}
}
```

clase principal:

```
public class Main {
 public static void main(String args[]) {
  SentenceDetectorUtil util = new SentenceDetectorUtil(
    "path//to//your//en-sent.bin");

  String data = "Welcome to Stackoverflow Documentation.This is the first example in OenNLP.";

  String[] sentences = util.getSentences(data);

  for (String s : sentences)
   System.out.println(s +"\n");
 }
}
```

la salida será:

> Bienvenido a la documentación de Stackoverflow.
>
> Este es el primer ejemplo en OpenNLP.

Lea OpenNLP en línea: https://riptutorial.com/es/nlp/topic/6052/opennlp

Creditos

S. No	Capítulos	Contributors
1	Empezando con nlp	Community, Gabor Angeli
2	Detección de límites de oraciones en Python	cgl, Franck Dernoncourt, JGreenwell, proycon
3	N-GRAMS	M Monis Ahmed Khan, thepurpleowl
4	OpenNLP	caffeinator13