



EBook Gratuito

APPENDIMENTO

nlp

Free unaffiliated eBook created from
Stack Overflow contributors.

#nlp

Sommario

Di.....	1
Capitolo 1: Iniziare con nlp.....	2
Osservazioni.....	2
Examples.....	2
Stanford CoreNLP.....	2
Capitolo 2: N-GRAMMI.....	4
introduzione.....	4
Sintassi.....	4
Osservazioni.....	4
Examples.....	4
Calcolo della probabilità condizionale.....	4
Capitolo 3: OpenNLP.....	6
Sintassi.....	6
Osservazioni.....	6
Examples.....	6
Rilevamento della frase usando openNLP usando CLI e Java API.....	6
Capitolo 4: Rilevamento del limite di frase in Python.....	9
Examples.....	9
Con Stanford CoreNLP, da Python.....	9
Con python-ucto.....	9
Utilizzando la libreria NLTK.....	10
Titoli di coda.....	11

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

Capitolo 1: Iniziare con nlp

Osservazioni

Questa sezione fornisce una panoramica di ciò che nlp è, e perché uno sviluppatore potrebbe voler usarlo.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno di nlp e collegarsi agli argomenti correlati. Poiché la Documentazione per nlp è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Stanford CoreNLP

[Stanford CoreNLP](#) è un popolare toolkit per l'elaborazione del linguaggio naturale che supporta molti compiti core della PNL.

Per scaricare e installare il programma, scaricare un pacchetto di rilascio e includere i file *.jar necessari nel classpath, oppure aggiungere la dipendenza da Maven central. Vedi [la pagina di download](#) per maggiori dettagli. Per esempio:

```
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`/*
```

Esistono tre modi supportati per eseguire gli strumenti CoreNLP: (1) utilizzando l' [API completamente personalizzabile di base](#) , (2) utilizzando l'API [Simple CoreNLP](#) o (3) utilizzando il [server CoreNLP](#) . Di seguito è riportato un semplice esempio di utilizzo per ciascuno di essi. Come caso d'uso motivante, questi esempi serviranno a predire l'analisi sintattica di una frase.

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. Semplice CoreNLP

```

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. Server CoreNLP

Avviare il server con il seguente (impostazione del percorso di classe in modo appropriato):

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

Ottieni un output in formato JSON per un determinato set di annotatori e stampalo come standard:

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

```

Per ottenere il nostro albero di analisi da JSON, possiamo navigare il JSON alle `sentences[i].parse`.

Leggi Iniziare con nlp online: <https://riptutorial.com/it/nlp/topic/2613/iniziare-con-nlp>

Capitolo 2: N-GRAMMI

introduzione

Gli N-GRAM sono modelli statistici che predicono la parola successiva nella frase usando le precedenti parole n-1. Questo tipo di modelli statistici che utilizzano sequenze di parole sono anche chiamati modelli di linguaggio. Ad esempio abbiamo una frase "Non posso leggere senza la mia lettura _____", possiamo dire che la prossima parola più probabile sarebbe "occhiali". N-GRAMS predice la parola successiva nella sequenza usando la probabilità condizionale della parola successiva. Il modello N-GRAM è molto essenziale nell'elaborazione del linguaggio e del linguaggio.

Sintassi

- La probabilità condizionale della prossima parola più probabile può essere ottenuta utilizzando un corpus grande (raccolta gestita di dati di testo o vocali), è tutto il conteggio delle cose (parole) dal corpus. L'obiettivo è trovare $P(w | h)$, che la probabilità della parola successiva nella sequenza abbia una certa storia h.
- Il concetto del modello N-GRAM è che invece di calcolare la probabilità di una parola data la sua intera storia, riduce la cronologia a poche parole precedenti. Quando usiamo una sola parola precedente per predire la parola successiva, si chiama modello Bi-GRAM. Ad esempio, abbiamo $P(\text{occhiali} | \text{lettura})$, la probabilità che la parola "occhiali" data la parola precedente "lettura" sia calcolata come: (Fare riferimento all'esempio)

Osservazioni

I modelli N-GRAM sono molto importanti quando dobbiamo identificare le parole in un input rumoroso e ambiguo. I modelli N-GRAM sono utilizzati in:

- Riconoscimento vocale
- Riconoscimento della scrittura a mano
- Correzione ortografica
- Traduzione automatica
- molte altre applicazioni

Puoi leggere di più sui modelli N-GRAM in:

- Libro di elaborazione del linguaggio e delle parole di Daniel Jurafsky e James H. Martin

Examples

Calcolo della probabilità condizionale

$$P(\text{occhiali} | \text{lettura}) = \text{Conta}(\text{occhiali da lettura}) / \text{Conta}(\text{lettura})$$

Contiamo le sequenze che `reading glasses` e `glasses` dal corpo e calcoliamo la probabilità.

Leggi N-GRAMMI online: <https://riptutorial.com/it/nlp/topic/8851/n-grammi>

Capitolo 3: OpenNLP

Sintassi

- `opennlp SentenceDetector ./en-sent.bin <./input.txt> output.txt`
- Inizializza `SentenceDetectorME` come questo: `SentenceDetectorME sentenceDetector = new SentenceDetectorME (modello);`
- Usa il metodo `'sentDetect'` per ottenere frasi come questa: `String sentence [] = sentenceDetector.sentDetect ("stringa di informazioni");`

Osservazioni

scaricare i modelli (come `en-sent.bin`) dal seguente [link](#)

Examples

Rilevamento della frase usando openNLP usando CLI e Java API

usando CLI:

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

usando l'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.requireNonNull(fileName);
        try {
            String data = new String(readAllBytes(get(fileName)));
            return data;
        } catch (IOException e) {
            System.out.println(e.getMessage());
            return null;
        }
    }
}
```


classe 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);
    }
}
```

classe principale:

```
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");
    }
}
```

l'output sarà:

Benvenuti in Documentazione StackOverflow.

Questo è il primo esempio in OpenNLP.

Leggi OpenNLP online: <https://riptutorial.com/it/nlp/topic/6052/opennlp>

Capitolo 4: Rilevamento del limite di frase in Python

Examples

Con Stanford CoreNLP, da Python

Per prima cosa è necessario eseguire un server [Stanford CoreNLP](#) :

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

Ecco un frammento di codice che mostra come passare i dati al server di Stanford CoreNLP, utilizzando il `pycorenlp` package Python.

```
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](#) è un tokeniser basato su regole per più lingue. Rileva anche il confine delle frasi. Sebbene sia scritto in C ++, esiste un'interconnessione Python [Python-ucto](#) per interfacciarlo con esso.

```
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)
```

Utilizzando la libreria NLTK

Puoi trovare maggiori informazioni sul tokenizer a livello di frase Python [Natural Language Toolkit \(NLTK\)](#) sul loro [wiki](#) .

Dalla tua riga di 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!']
```

[Leggi Rilevamento del limite di frase in Python online:](#)

<https://riptutorial.com/it/nlp/topic/3833/rilevamento-del-limite-di-frase-in-python>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con nlp	Community , Gabor Angeli
2	N-GRAMMI	M Monis Ahmed Khan , thepurpleowl
3	OpenNLP	caffeinator13
4	Rilevamento del limite di frase in Python	cgl , Franck Deroncourt , JGreenwell , proycon