eBook Gratuit

APPRENEZ nlp

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#nlp

Table des matières

À propos

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

Chapitre 1: Démarrer avec nlp

Remarques

Cette section fournit une vue d'ensemble de ce qu'est nlp et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les sujets importants au sein de nlp, et établir un lien avec les sujets connexes. La documentation de nlp étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Examples

Stanford CoreNLP

Stanford CoreNLP est une boîte à outils populaire de traitement du langage naturel prenant en charge de nombreuses tâches de base de la PNL.

Pour télécharger et installer le programme, téléchargez un package de version et incluez les fichiers `*.jar` nécessaires dans votre chemin de classe ou ajoutez la dépendance de Maven Central. Voir la page de téléchargement pour plus de détails. Par exemple:

```
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`/*
```

Il existe trois méthodes prises en charge pour exécuter les outils CoreNLP: (1) utiliser l' API entièrement personnalisable de base , (2) utiliser l'API Simple CoreNLP , ou (3) utiliser le serveur CoreNLP . Un exemple d'utilisation simple pour chacun est donné ci-dessous. En tant que cas d'utilisation motivant, ces exemples serviront à prédire l'analyse syntaxique d'une phrase.

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. **Simple 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. **Serveur CoreNLP**

   Démarrez le serveur avec les éléments suivants (en définissant votre chemin de classe de manière appropriée):

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

   Obtenez une sortie au format JSON pour un ensemble donné d'annotateurs et imprimez-la en sortie standard:

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

   Pour obtenir notre arbre d'analyse à partir du JSON, nous pouvons naviguer dans le JSON en `sentences[i].parse` .

Lire Démarrer avec nlp en ligne: https://riptutorial.com/fr/nlp/topic/2613/demarrer-avec-nlp

Chapitre 2: Détection des limites de phrase en Python

Examples

Avec Stanford CoreNLP, de Python

Vous devez d'abord exécuter un serveur Stanford CoreNLP :

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

Voici un extrait de code montrant comment transmettre des données au serveur Stanford CoreNLP, en utilisant le `pycorenlp` Python `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)
```

Avec python-ucto

Ucto est un tokeniser basé sur des règles pour plusieurs langues. Il détecte également les limites de phrases. Bien qu'il soit écrit en C ++, il existe une liaison Python avec python-ucto pour s'y connecter .

```
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)
```

Utilisation de la bibliothèque NLTK

Vous pouvez trouver plus d'informations sur le tokenizer de niveau phrase de NLPK (Python Natural Language Toolkit) sur leur wiki .

Depuis votre ligne de commande:

```
$ 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!']
```

Lire Détection des limites de phrase en Python en ligne:
https://riptutorial.com/fr/nlp/topic/3833/detection-des-limites-de-phrase-en-python

Chapitre 3: N-GRAMS

Introduction

Les N-GRAM sont des modèles statistiques qui prédisent le mot suivant dans la phrase en utilisant les n-1 mots précédents. Ce type de modèle statistique utilisant des séquences de mots est également appelé modèles linguistiques. Par exemple, nous avons une phrase "Je ne peux pas lire sans lire _____", nous pouvons dire que le prochain mot le plus probable serait "lunettes". N-GRAMS prédit le mot suivant dans la séquence en utilisant la probabilité conditionnelle du mot suivant. Le modèle N-GRAM est essentiel dans le traitement de la parole et du langage.

Syntaxe

- La probabilité conditionnelle du mot le plus probable suivant peut être obtenue en utilisant un grand corpus (collection gérée de données textuelles ou vocales), il s'agit de compter des choses (mots) du corpus. Le but est de trouver P (w | h), ce que la probabilité du mot suivant dans la séquence a donné une certaine histoire h.
- Le concept du modèle N-GRAM est que, au lieu de calculer la probabilité d'un mot compte tenu de son histoire entière, il réduit l'historique à quelques mots précédents. Lorsque nous utilisons un seul mot précédent pour prédire le mot suivant, on l'appelle un modèle Bi-GRAM. Par exemple, nous avons P (lunettes | lecture), la probabilité du mot "lunettes" étant donné le mot précédent "lecture" est calculée comme suit: (Voir l'exemple)

Remarques

Les modèles N-GRAM sont très importants lorsque nous devons identifier des mots dans une entrée bruyante et ambiguë. Les modèles N-GRAM sont utilisés dans:

- Reconnaissance de la parole
- Reconnaissance de la main
- Correction orthographique
- Traduction automatique
- beaucoup d'autres applications

Vous pouvez en savoir plus sur les modèles N-GRAM dans:

- Livre sur le traitement de la parole et du langage par Daniel Jurafsky et James H. Martin

Examples

Calculer la probabilité conditionnelle

P (*lunettes* | *lecture*) = compte (*lunettes de lecture*) / compte (*lecture*)

Nous comptons les séquences en `reading glasses` et les `glasses` du corpus et calculons la probabilité.

Lire N-GRAMS en ligne: https://riptutorial.com/fr/nlp/topic/8851/n-grams

Chapitre 4: OpenNLP

Syntaxe

- opennlp SentenceDetector ./en-sent.bin <./input.txt> output.txt
- Initialiser SentenceDetectorME comme ceci: SentenceDetectorME phraseDetector = new SentenceDetectorME (model);
- Utilisez la méthode 'sentDetect' pour obtenir des phrases comme celles-ci: Phrases de chaînes [] = phraseDetector.sentDetect ("chaîne d'informations");

Remarques

télécharger des modèles (comme en-sent.bin) à partir du lien suivant

Examples

Détection de phrase à l'aide de openNLP à l'aide de l'API CLI et Java

en utilisant CLI:

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

en utilisant 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.nonNull(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");
 }
}
```

la sortie sera:

> Bienvenue dans la documentation de Stackoverflow.
>
> C'est le premier exemple dans OpenNLP.

Lire OpenNLP en ligne: https://riptutorial.com/fr/nlp/topic/6052/opennlp

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec nlp	Community, Gabor Angeli
2	Détection des limites de phrase en Python	cgl, Franck Dernoncourt, JGreenwell, proycon
3	N-GRAMS	M Monis Ahmed Khan, thepurpleowl
4	OpenNLP	caffeinator13