Pengenalan Natural Language Toolkit (NLTK) Bagian 1

Yunita Sari

yunita.sari@ugm.ac.id

Departemen Ilmu Komputer dan Elektronika, UGM

September 2019

Natural Language Toolkit atau yang biasa disingkat dengan NLTK adalah sebuah platform berbasis Python yang dikembangkan untuk memproses data text. NLTK dilengkapi dengan lebih dari 50 *corpora* dan *lexical resources* seperti Wordnet. Selain itu NLTK juga menyediakan librari untuk *text processing* mulai dari klasifikasi, tokenisasi, stemming, tagging, parsing dll. NLTK tersedia adalah salah satu *open source tools* yang bisa diakses secara gratis, dan tersedia baik untuk sistem operasi Windows, Mac OS X dan Linux. Dalam artikel kali ini, akan ditunjukkan tentang beberapa fungsi dari NLTK. Step pertama yang harus dilakukan sebelum mengikuti tutorial ini adalah menginstall NLTK. Step untuk instalasi bisa dilihat pada *link* berikut https://www.nltk.org/install.html. Untuk meng-install NLTK, ada beberapa software/library yang harus diinstall terlebih dahulu seperti Python (saat ini versi 3) dan Numpy. Beberapa *text processing* yang akan kita pelajari kali ini yaitu:

1. mengubah teks ke huruf kecil
2. membuang tanda baca/*punctuation* dan spasi
3. tokenisasi kata
4. tokenisasi kalimat
5. tokenisasi twitter
6. stop words removal
7. pos tags

1 Mengubah teks ke huruf kecil

Operasi yang pertama yaitu mengubah semua karakter yang ada pada teks/dokumen ke huruf kecil.

```
In [23]: input_str = "China, India, United States, Indonesia, dan Brazil adalah 5 negara \
         dengan populasi terbanyak di dunia"
         input_str = input_str.lower()
         print(input_str)
```

```
china, india, united states, indonesia, dan brazil adalah 5 negara dengan populasi terbanyak di
```

2 Membuang tanda baca/punctuation dan spasi

Berikutnya kita akan menghapus semua tanda baca atau *punctuation*

```
In [40]: import string
         input_str = "Ini &adalah [sebuah] contoh? {dari} string. dengan.? punctuation!!!!"
         result = input_str.translate(str.maketrans('','',string.punctuation))
         print(result)
```

```
Ini adalah sebuah contoh dari string dengan punctuation
```

Dan menghapus *white-space*

```
In [41]: input_str = "\t contoh string\t"
         input_str = input_str.strip()
         input_str
```

```
Out[41]: 'contoh string'
```

3 Tokenisasi kata

Fungsi berikutnya yaitu tokenisasi. Tokenisasi adalah proses membagi text ke dalam token. Token merupakan rangkaian karakter yang bisa dipisahkan oleh spasi atau tanda baca/*punctuation*. Pada NLTK, ada 2 jenis *tokenizer* untuk level kata yang paling sering digunakan yaitu *word_tokenize* dan *wordpunct_tokenize*. Perbedaan dari kedua tokenizer tersebut adalah, *word_tokenize* sebetulnya menggunakan Treebank *tokenizer*. Sedangkan *wordpunct_tokenize* memisahkan token dengan menggunakan *regular expression*. Ada perbedaan dari hasil tokenisasi dari kedua *tokenizer* tersebut. *Word_tokenize* akan memisahkan *standard contraction* menjadi 2 token yang berbeda, dimana salah satu token akan mengandung tanda petik (’). Sedangkan *wordpunct_tokenize* akan memisahkan menjadi 3 token, dimana tanda petik (’) menjadi token tersendiri. Contoh penggunaan dari kedua *tokenizer* pada level kata bisa dilihat sebagai berikut:

```
In [42]: from nltk.tokenize import word_tokenize, wordpunct_tokenize,sent_tokenize
```

```
In [58]: my_string = "Two plus two is four, minus one that's three - quick maths. \
         Every day man's on the block. Smoke trees. \
         See your girl in the park, that girl is an uckers. \
         When the thing went quack quack quack, your men were ducking! \
         Hold tight Asznee, my brother. He's got a pumpy. Hold tight my man, my guy."
         print (word_tokenize(my_string))
```

```
['Two', 'plus', 'two', 'is', 'four', ',', 'minus', 'one', 'that', "'s", 'three', '-', 'quick', '
```

```
In [59]: print (wordpunct_tokenize(my_string))
```

```
['Two', 'plus', 'two', 'is', 'four', ',', 'minus', 'one', 'that', "'", 's', 'three', '-', 'quick
```

4 Tokenisasi Kalimat

Selain pada level kata, NLTK juga menyediakan tokenisasi pada level kalimat.

```
In [60]: print(sent_tokenize(my_string))
```

```
["Two plus two is four, minus one that's three - quick maths.", "Every day man's on the block.",
```

5 Tokenisasi Tweet

NLTK juga menyediakan *tokenizer* untuk teks dalam bentuk tweet. Dengan *TweetTokenizer* , teks dalam tweet akan dinormalkan. Sebagai contoh *mentions* dan karakter yang terlalu banyak dalam sebuah kata akan dihapus.

```
In [61]: from nltk.tokenize import TweetTokenizer
         Tokenizer = TweetTokenizer(strip_handles=True, reduce_len=True)
         tweet = "@Kirana_Sutanto I am so happppppy"
         print(Tokenizer.tokenize(tweet))
```

```
['I', 'am', 'so', 'happpy']
```

6 Stopword removal

Berikutnya, NLTK bisa digunakan untuk menghapus semua *stopwords* yang ada pada sebuah kalimat. *Stopwords* merupakan kata-kata yang secara makna tidak terlalu berarti. Dalam bahasa Inggris, contoh *stopwords* antara lain: *at, the, and, to, in*

```
In [46]: from nltk.corpus import stopwords
         stopwords.readme().replace('\n', ' ')
```

```
Out[46]: 'Stopwords Corpus  This corpus contains lists of stop words for several languages.  The
```

```
In [65]: input_str = "NLTK is a leading platform for building Python programs to work with \
         human language data."
         stop_words = set(stopwords.words("english"))
         tokens = word_tokenize(input_str)
         result = [i for i in tokens if not i in stop_words]
         print (result)
```

```
['NLTK', 'leading', 'platform', 'building', 'Python', 'programs', 'work', 'human', 'language', '
```

NLTK sendiri menyediakan list dari *stopwords* untuk beberapa bahasa, termasuk didalamnya Bahasa Indonesia. Total ada 758 kata dalam Bahasa Indonesia yang dikategorikan sebagai *stopwords* oleh NLTK.

```
In [63]: stopwords.fileids()
```

```
Out[63]: ['arabic',
          'azerbaijani',
          'danish',
          'dutch',
          'english',
          'finnish',
          'french',
          'german',
          'greek',
          'hungarian',
          'indonesian',
          'italian',
          'kazakh',
          'nepali',
          'norwegian',
          'portuguese',
          'romanian',
          'russian',
          'slovene',
          'spanish',
          'swedish',
          'tajik',
          'turkish']

In [54]: stopwords.words('indonesian')[:10]

Out[54]: ['ada',
          'adalah',
          'adanya',
          'adapun',
          'agak',
          'agaknya',
          'agar',
          'akan',
          'akankah',
          'akhir']

In [55]: len(stopwords.words(['indonesian']))

Out[55]: 758
```

7 Part-of-speech tagging

Fitur dari NLTK berikutnya yaitu *Part-of-speech tagging* (PoS tagging). Fitur ini digunakan untuk meng-*assign* PoS tag ke tiap kata dalam sebuah kalimat. Kalimat harus dipisahkan menjadi token terlebih dahulu sebelum PoS tag di-*assign* ke setiap kata. Hanya ada beberapa bahasa yang sudah ter-*cover* oleh NLTK untuk fitur PoS tag ini. Bahasa Indonesia sendiri belum termasuk salah satunya.

```
In [66]: from nltk import pos_tag
         input_str="Parts of speech examples: an article, to write, interesting,\
         easily, and, of"
         result = pos_tag(word_tokenize(input_str))
         print(result)
```

```
[('Parts', 'NNS'), ('of', 'IN'), ('speech', 'NN'), ('examples', 'NNS'), (':', ':'), ('an', 'DT')
```

Demikian tutorial pengenalan NLTK untuk bagian yang pertama. Fitur-fitur lain dari NLTK akan dibahas pada artikel selanjutnya.

Referensi: 1. https://www.nltk.org/