NLP with Python

Authored by

M. Namgyal BRISSON

Independant Consultant for NXP

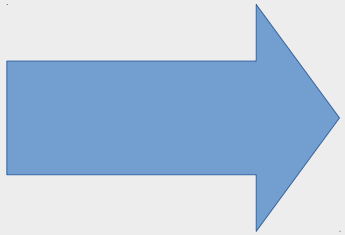
Python Mentor on OpenClassrooms

NLP with Python

- One-Hot Encoding
- Basic Features Extraction
  - Readability Tests
    - Flesch Reading Ease Score
    - Gunning Fog Index Score
- Tokenization & Lemmatization
- Part-Of-Speech (POS) Tagging
- Named Entity Recognition (NER)
- Bag of Words (BoW)
- N-grams
- Chatbot Example
- To go further

One-Hot Encoding

...	Gender
...	male
...	female
...	female
...	male



...	Gender_male	Gender_female
...	1	0
...	0	1
...	0	1
...	1	0

https://colab.research.google.com/drive/13X5jeN3oY3CFJL3ok5rwP6tifiRhIiz-

Basic Features Extraction

- **Words count**
- **Characters count**
- **Words average length**
- **Pattern specific count (for instance, [hash]tags)**

Other Features Extraction

- *Sentences count*
- *Paragraph count*
- *Capitalized words*
- *Uppercased words*
- *Quantities (Numerical)*
- *Etc.*

Readability Tests

- Flesch Reading Ease Score
  - Greater the average sentence length, harder the text is to read
    - « Quick & short example »
    - « Pretty much longer sentence, therefore harder to read »
  - Greater the number of syllables in a word, harder the text is to read
    - « I feel good at home »
    - « I’m positively affected by being at my domicile »

Higher the score is, greater the readability is!

Readability Tests

Reading Ease Score	Descriptive Categories	Estimated Reading Grade
90 – 100	Very Easy	5th Grade
80 – 90	Easy	6th Grade
70 – 80	Fairly Easy	7th Grade
60 – 70	Standard / Plain English	8th and 9th Grade
50 – 60	Fairly Difficult	10th to 12th Grade (High School Sophomore to Senior)
30 – 50	Difficult	In College
0 - 30	Very Difficult	College Graduate

Readability Tests

- Gunning Fog Index Score
  - Based on following principles:
    - Average sentence length
    - Percentage of complex words

Lesser the score is, greater the readability is!

Readability Tests

Gunning Fog Score

The index estimates the years of formal education needed to understand the text on a first reading.

The fog index is commonly used to confirm that text can be read easily by the intended audience.

Formula:
(average_words_sentence + number_words_ three_syllables_plus)
* 0.4

The lower the number, the more understandable the content will be to your visitors.

*Results over 17 are reported as seventeen, where 17 is considered post-graduate level.

Fog Index	Reading level by grade
17	College graduate
16	College senior
15	College junior
14	College sophomore
13	College freshman
12	High school senior
11	High school junior
10	High school sophomore
9	High school freshman
8	Eighth grade
7	Seventh grade
6	Sixth grade

https://colab.research.google.com/drive/1ZyIU1BZEP5WVS-EE1vghHJqMjgHWe1Uc

Tokenization & Lemmatization

- **Tokenization is splitting a sentence into its constituant parts**
  - « Hello, my name is Namgyal. »
  - → **[**`Hello`**,** `,`**,** `my`**,** `name`**,** `is`**,** `Namgyal`**,** `.`**]**
- **Lemmatization is converting words into its base form**
  - « is », « am », « are » → **« be »**
  - « deleting », « deletes », « deleted », « deletion » → **« delete »**
  - « n’t » → **« not »**
  - « ‘ve » → **« have »**

https://colab.research.google.com/drive/10HQ-OHeSSRHTcVkETPPaRkZqnQpm8mK4

Part-Of-Speach (POS) Tagging

Assigning every word, its corresponding part of speech.

Used for:

- **Word-sense disambiguation**
  - « The bear is an animal »
  - « Bear it up! »
- **Sentiment analysis**
- **Question answering**
- **Opinion spam detection**

Part-Of-Speach (POS) Tagging

WORD	POS
I	Pronoun
have	Verb
a	Article
cat	Noun

https://spacy.io/api/annotation#pos-universal

https://colab.research.google.com/drive/1i_Q-QNhCOUBtmNeE6_CmUdsB3nAvAVLo

Named Entity Recognition (NER)

Identifying & classifying named entities into predifined categories.

- Person
- Country
- Organization
- ...

Can be used for:

- News article classification
- Efficient search algorithms
- Question answering
- Customer service
- ...

https://colab.research.google.com/drive/1Tyl_7tmz8j7ByUN_HjlQMRLIIKzilDiN

Bag of Words (BoW)

ML algorithms needs tabular data and numerical training features

- However, it is not the case for textual data (ie. movie reviews)
- Therefore one needs to convert words into vectors

Here comes the « Bag of words model »which allows to

- Extract word as token
- Compute the word tokens' frequency
- Build a word vector out of these

https://colab.research.google.com/drive/1xQ6bwhRwaa7zBIZU82U6J1RVkHUjnhuS

N-Grams

BoW shortcomings

- « The moment was nice and not boring » → **Positive**
- « The moment was not good and boring » → **Negative**

In the BoW approach one will get the same vector as it contains exactly the same words!

Unfortuately, BoW approach looses the context of the words…

N-grams is a contiguous sequence of « n » elements and will help us to handle those cases.

N-Grams

The BoW approach is nothing more than a n-gram model where « n » equals 1.

Let’s see some examples where « n » is superior to 1

If one says, « The movie was not good and boring » with n=2, it produces :

[

« The movie »,

« movie was »,

« was not »,

« not good »,

« good and »,

« and boring »

]

And so forth…

It adds context to the words, like here « was not »

N-Grams

Shortcomings

It increases the dimensions and in ML it will have a severe impact

It is known as the « **Curse of Dimensionality** »

It is then recommended to keep « n » small

Chatbot Example

Minimalistic chatbot based on flight suggestions.

Libraries & techniques used:

- RASA NLU
- SQL database
- Chatito data generation

https://github.com/nam4dev/chatbot_rasa_nlu_presentation

To go further

Not developped in this course:

- Tf-idf weight
- Cosine similarity
- ...

Thanks for your attention :)