

```
In [79]:  
import spacy  
nlp = spacy.load("fr_core_news_sm")  
doc = nlp("Bonjour Bonne après-midi Je m'appelle Mondly e suis ravi  
de vous rencontrer Comment ça va")
```

```
In [80]: doc
```

```
Out[80]: Bonjour Bonne après-midi Je m'appelle Mondly e suis ravi de vous rencontrer Comm  
ent ça va
```

```
In [81]:  
for token in doc:  
    print(token.text)
```

```
Bonjour  
Bonne  
après-midi  
Je  
m'  
appelle  
Mondly  
e  
suis  
ravi  
de  
vous  
rencontrer  
Comment  
ça  
va
```

```
In [82]:  
for token in doc:  
    print(token.pos_)
```

```
PROPN  
ADJ  
NOUN  
PRON  
PRON  
VERB  
PROPN  
AUX  
AUX  
VERB  
ADP  
PRON  
VERB  
ADV  
PRON  
VERB
```

In [83]:

```
for token in doc:  
    print(token.text,token.pos_)
```

```
Bonjour PROPN  
Bonne ADJ  
après-midi NOUN  
Je PRON  
m' PRON  
appelle VERB  
Mondly PROPN  
e AUX  
suis AUX  
ravi VERB  
de ADP  
vous PRON  
rencontrer VERB  
Comment ADV  
ça PRON  
va VERB
```

In [84]:

```
for token in doc:  
    print(token.text,token.pos_,token.lemma_)
```

```
Bonjour PROPN Bonjour  
Bonne ADJ bon  
après-midi NOUN après-midi  
Je PRON je  
m' PRON me  
appelle VERB appeler  
Mondly PROPN Mondly  
e AUX e  
suis AUX être  
ravi VERB ravir  
de ADP de  
vous PRON vous  
rencontrer VERB rencontrer  
Comment ADV comment  
ça PRON cela  
va VERB aller
```

In [85]:

```
text = "Il existe essentiellement deux types de tâches de résumé  
extractif en fonction de ce sur quoi se concentre le programme de  
résumé."
```

In [86]:

```
text
```

Out[86]: 'Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.'

```
In [87]: import spacy  
from spacy.lang.fr.stop_words import STOP_WORDS  
from string import punctuation
```

```
In [88]: stopwords = list(STOP_WORDS)
```

```
In [89]: stopwords
```

```
Out[89]: ['mienne',
'differente',
'etc',
'treize',
'huitième',
'attendu',
'sous',
'enfin',
'on',
'cependant',
'relativement',
'quelle',
'suffisant',
'gens',
'sixième',
'pour',
'pres',
'est',
'sent',
'j',
'cinquante',
'lesquels',
'compris',
'veingt',
'lors',
'merci',
'cette',
'notre',
'possibles',
'da',
'quinze',
'ayant',
'etais',
'i',
'quoi',
'veôtres',
'parler',
'déjà',
'avons',
'devra',
'néanmoins',
'dits',
'eu',
'uns',
'dessus',
'voici',
'ils',
'faisaient',
'dixième',
'douze',
'tous',
'auxquels',
'auquel',
'nôtres',
'celles',
```

'ceux-ci',  
'par',  
'lui-même',  
'toujours',  
'avaient',  
"s'",  
'moi-même',  
'ceci',  
'etait',  
'derrière',  
'du',  
'etaient',  
'precisement',  
'unes',  
'quarante',  
'ma',  
'mille',  
'nous-mêmes',  
'alors',  
'es',  
'vôtre',  
'lorsque',  
'celles-la',  
'd',  
'différente',  
'huit',  
'ouias',  
'souvent',  
'différentes',  
'celui',  
'revoilà',  
'tant',  
'aurait',  
'envers',  
'une',  
'o',  
'certes',  
'mais',  
'etant',  
'durant',  
'as',  
'cinq',  
'auxquelles',  
'l',  
'autrement',  
'memes',  
'près',  
'nous',  
'parce',  
'juste',  
'spécifiques',  
'dire',  
'rendre',  
'feront',  
'dejà',

'où',  
'ne',  
'va',  
'restent',  
'plutot',  
'mêmes',  
'afin',  
'toi-meme',  
'préalable',  
'diverse',  
's',  
'peuvent',  
'sait',  
'maintenant',  
'dont',  
'prealable',  
'directe',  
'sienne',  
'dix',  
'quelconque',  
'vos',  
'pourrais',  
'également',  
'celle',  
'ha',  
'cela',  
'semblable',  
'tout',  
'nul',  
'sans',  
'suivants',  
'quatre-vingt',  
'ton',  
'ses',  
'y',  
'sinon',  
'ouverte',  
'premièrement',  
'avais',  
'tres',  
'seulement',  
'rend',  
'les',  
'mon',  
'sept',  
'cent',  
'quatorze',  
'nôtre',  
'sur',  
'suffisante',  
'quel',  
'ainsi',  
'importe',  
'ni',  
'tente',

'vers',  
'hormis',  
'douzième',  
'vé',  
'vas',  
'cinquantième',  
'quelques',  
'seuls',  
'tend',  
'moindres',  
'que',  
'dix-neuf',  
'ci',  
'au',  
'quant',  
"n'",  
'pourrait',  
'vous',  
'bat',  
'seraient',  
'suivant',  
'leurs',  
'pendant',  
'plutôt',  
'vu',  
'dessous',  
'chacun',  
'â',  
'desquelles',  
'des',  
'via',  
'tels',  
'diverses',  
'entre',  
'être',  
'touchant',  
'restant',  
'je',  
'quant-à-soi',  
'soit',  
'dix-huit',  
'autrui',  
'effet',  
'eh',  
'elle-même',  
'différents',  
'pouvait',  
'allons',  
'différent',  
'première',  
'puis',  
'depuis',  
'plus',  
'certain',  
'façon',

'specifique',  
'dix-sept',  
'votre',  
'nombreux',  
'peut',  
'abord',  
'tienne',  
'ai',  
"m'",  
'miens',  
'tiennes',  
'déjà',  
'même',  
'delà',  
'hue',  
'dehors',  
'ce',  
'quatrième',  
'aie',  
'tellement',  
'bas',  
"j'",  
'ah',  
'ces',  
'six',  
'excepté',  
'seules',  
"qu'",  
'ouste',  
'tenant',  
'hem',  
'pas',  
'moi-meme',  
'parle',  
'exactement',  
'me',  
'suis',  
'semble',  
"d'",  
'onzième',  
'ouverts',  
'concernant',  
'hui',  
'maint',  
'avoir',  
'sa',  
'ès',  
'suivantes',  
'tien',  
'précisement',  
'avant',  
'assez',  
'désormais',  
'quoique',  
'leur',

'encore',  
'allaient',  
'duquel',  
'elle-même',  
'peux',  
'miennes',  
'vont',  
'malgré',  
'peu',  
'devant',  
'autre',  
'ouvert',  
'mes',  
'longtemps',  
'était',  
'et',  
'doit',  
'elles-mêmes',  
'sont',  
'troisième',  
'trente',  
'specifiques',  
'dit',  
'anterieure',  
'puisque',  
'tu',  
'parlent',  
'aura',  
'combien',  
'malgre',  
'en',  
'quels',  
'font',  
'partant',  
'neanmoins',  
'étaient',  
'c''',  
'nombreuses',  
'certains',  
'environ',  
'votres',  
'moins',  
'quiconque',  
'ça',  
'personne',  
'celui-là',  
'jusqu',  
'sien',  
'vais',  
'notamment',  
"t'",  
'eux',  
'antérieures',  
'spécifique',  
'hé',

'pense',  
'soixante',  
'ont',  
'celui-la',  
'autres',  
'or',  
'laisser',  
'toute',  
'celle-là',  
'deuxième',  
'houp',  
'toi',  
'très',  
'n'',  
'seront',  
'differentes',  
'celle-la',  
'toi-même',  
'elles-memes',  
'na',  
'suivre',  
'anterieures',  
'm'',  
'deuxièmement',  
'outre',  
'dans',  
'revoici',  
'ait',  
'ou',  
'chacune',  
'certaines',  
'semblaient',  
'donc',  
'celles-là',  
'aupres',  
'selon',  
'suffit',  
'elle',  
'nouveau',  
"quelqu'un",  
"c'",  
'siennes',  
'stop',  
'différents',  
'comment',  
'certaine',  
'qui',  
'semblent',  
'sera',  
'cinquante',  
'plusieurs',  
'derriere',  
'cinquième',  
'quelque',  
'possible',

'celui-ci',  
'celle-ci',  
'devers',  
'seize',  
'troisièmement',  
'antérieur',  
'different',  
'de',  
'soi-même',  
'anterieur',  
'ho',  
'auraient',  
'voilà',  
'pourquoi',  
'cet',  
'fais',  
'quelles',  
'facon',  
'proche',  
'lequel',  
'trois',  
'lui',  
'divers',  
'à',  
'septième',  
'telle',  
'hors',  
'surtout',  
'hou',  
'nos',  
'étais',  
'quand',  
'jusque',  
'ceux',  
'ta',  
'avait',  
'seul',  
'sauf',  
'moi',  
'chaque',  
'dedans',  
'chez',  
'fait',  
'tenir',  
'suivante',  
'auront',  
'comme',  
'meme',  
'un',  
'basee',  
'directement',  
'reste',  
'hep',  
'desormais',  
'faisant',

'dite',  
'doivent',  
'il',  
'aussi',  
'egalement',  
't',  
'là',  
'debout',  
'desquels',  
'lès',  
'retour',  
'procedant',  
'hi',  
'parmi',  
'après',  
'apres',  
'celles-ci',  
'suit',  
'voila',  
'premier',  
'lesquelles',  
'relative',  
'quatrièmement',  
'vous-mêmes',  
'car',  
'eux-mêmes',  
'étant',  
'avec',  
'té',  
'la',  
'toutes',  
'ô',  
'si',  
'seule',  
'deja',  
'aux',  
'se',  
'ceux-là',  
'neuvième',  
'soi',  
'deux',  
'quatre',  
'te',  
'dès',  
'tiens',  
'permet',  
'a',  
'pu',  
'revoila',  
'laquelle',  
'etre',  
'tes',  
'le',  
'parfois',  
'mien',

```
'telles',
'tel',
"l'",
'lui-même',
'siens',
'qu\'',
'serait',
'son',
'elles',
'antérieure',
'soi-même',
'onze']
```

In [90]: `len(stopwords)`

Out[90]: 507

In [91]: `nlp = spacy.load("fr_core_news_sm")`

In [92]: `nlp`

Out[92]: <spacy.lang.fr.French at 0x15083a9e380>

In [93]: `text`

Out[93]: 'Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.'

In [94]: `doc = nlp(text)`  
`doc`

Out[94]: Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.

In [95]: `tokens = [token.text for token in doc]`  
`print(tokens)`

```
['Il', 'existe', 'essentiellement', 'deux', 'types', 'de', 'tâches', 'de', 'résumé', 'extractif', 'en', 'fonction', 'de', 'ce', 'sur', 'quoi', 'se', 'concentre', 'le', 'programme', 'de', 'résumé', '.']
```

In [96]: `tokens`

```
Out[96]: ['Il',  
          'existe',  
          'essentiellement',  
          'deux',  
          'types',  
          'de',  
          'tâches',  
          'de',  
          'résumé',  
          'extractif',  
          'en',  
          'fonction',  
          'de',  
          'ce',  
          'sur',  
          'quoi',  
          'se',  
          'concentre',  
          'le',  
          'programme',  
          'de',  
          'résumé',  
          '..']
```

```
In [97]: len(tokens)
```

```
Out[97]: 23
```

```
In [98]: punctuation
```

```
Out[98]: '!"#$%&\'()*+, -./:; <=>?@[\\]^_`{|}~'
```

```
In [99]: doc
```

```
Out[99]: Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.
```

```
In [100...]: word_frequencies = {}  
for word in doc:  
    if word.text.lower() not in stopwords:  
        if word.text.lower() not in punctuation:  
            if word.text not in word_frequencies.keys():  
                word_frequencies[word.text] = 1  
            else:  
                word_frequencies[word.text] += 1
```

```
In [101...]: print(word_frequencies)
```

```
{'existe': 1, 'essentiellement': 1, 'types': 1, 'tâches': 1, 'résumé': 2, 'extractif': 1, 'fonction': 1, 'concentre': 1, 'programme': 1}
```

```
In [102...]: len(word_frequencies)
```

```
Out[102]: 9
```

```
In [103...]: max_frequency = max(word_frequencies.values())
```

```
In [104...]: max_frequency
```

```
Out[104]: 2
```

```
In [105...]: for word in word_frequencies.keys():
    word_frequencies[word] = word_frequencies[word]/max_frequency
```

```
In [106...]: print(word_frequencies)
```

```
{'existe': 0.5, 'essentiellement': 0.5, 'types': 0.5, 'tâches': 0.5, 'résumé': 1.0, 'extractif': 0.5, 'fonction': 0.5, 'concentre': 0.5, 'programme': 0.5}
```

```
In [107...]: sentence_tokens = [sent for sent in doc.sents]
sentence_tokens
```

```
Out[107]: [Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.]
```

```
In [108...]: len(sentence_tokens)
```

```
Out[108]: 1
```

```
In [109...]: sentence_score= {}
for sent in sentence_tokens:
    for word in sent:
        if word.text.lower() in word_frequencies.keys():
            if sent not in sentence_score.keys():
                sentence_score[sent] =
word_frequencies[word.text.lower()]
            else:
                sentence_score[sent] +=
word_frequencies[word.text.lower()]
```

```
In [110...]: sentence_score
```

```
Out[110]: {Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.: 6.0}
```

```
In [111...]: from heapq import nlargest
```

```
In [112...]: select_length = int(len(sentence_tokens)*0.4)
```

```
In [113...]: select_length
```

```
Out[113]: 0
```

```
In [114...]: summary = nlargest(select_length,sentence_score,key=sentence_score.get)
```

```
In [115...]: summary
```

```
Out[115]: []
```

```
In [116...]: sentence_score
```

```
Out[116]: {Il existe essentiellement deux types de tâches de résumé extractif en fonction de ce sur quoi se concentre le programme de résumé.: 6.0}
```

```
In [117...]: final_summary = [word.text for word in summary]
```

```
In [118...]: final_summary
```

```
Out[118]: []
```

```
In [119...]: print(summary)
```

```
[]
```