In [1]:
```
pip install python-docx
```

```
Collecting python-docx
  Downloading https://files.pythonhosted.org/packages/e4/83/c66a1934ed5ed8ab1dbb9931f1779079f8bca0f6bbc5793c06c4b5e7d671/python-docx-0.8.10.tar.gz (5.5MB)
     |████████████████████████████████| 5.5MB 4.4MB/s
Requirement already satisfied: lxml>=2.3.2 in /usr/local/lib/python3.6/dist-packages (from python-docx) (4.2.6)
Building wheels for collected packages: python-docx
  Building wheel for python-docx (setup.py) ... done
  Created wheel for python-docx: filename=python_docx-0.8.10-cp36-none-any.whl size=184491 sha256=5962e1e616189f3af3071ea4e6c89c1456dcc4369e2004d4a1a2953e9a3b2da4
  Stored in directory: /root/.cache/pip/wheels/18/0b/a0/1dd62ff812c857c9e487f27d80d53d2b40531bec1acecfa47b
Successfully built python-docx
Installing collected packages: python-docx
Successfully installed python-docx-0.8.10
```

In [0]:
```
import pandas as pd
import numpy as np
import docx
import unicodedata
import os
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

In [0]:
```
def getDocx(filename):
    doc = docx.Document(filename)
    fullText = []
    for para in doc.paragraphs:
        fullText.append(para.text)
    return ' '.join(fullText)
```

In [0]:
```
def list_text(count_files,list_text):
    for i in range(count_files):
        new_str=unicodedata.normalize("NFKD",getDocx(str(i+1)+".docx"))
        new_str=new_str.lower()
        list_text.append(new_str)
    return list_text
```

In [0]:
```
all_offres=[]
all_offres=list_docx(11,all_offres)
```

In [0]:
```
offres=pd.DataFrame({"Offre":all_offres})
offres.to_csv("offres.csv",sep=";")
```

In [0]:
```
import re
alphabets= "([A-Za-z])"
prefixes = "(Mr|St|Mrs|Ms|Dr)[.]"
suffixes = "(Inc|Ltd|Jr|Sr|Co)"
starters = "(Mr|Mrs|Ms|Dr|He\s|She\s|It\s|They\s|Their\s|Our\s|We\s|But\s|However\s|That\s|This\s|Wherever)"
acronyms = "([A-Z][.][A-Z][.](?:[A-Z][.])?)"
websites = "[.](com|net|org|io|gov)"

def split_into_sentences(text):
    text = " " + text + "  "
    text = text.replace("\n"," ")
    text = re.sub(prefixes,"\\1<prd>",text)
    text = re.sub(websites,"<prd>\\1",text)
    if "Ph.D" in text: text = text.replace("Ph.D.","Ph<prd>D<prd>")
    text = re.sub("\s" + alphabets + "[.] "," \\1<prd> ",text)
    text = re.sub(acronyms+" "+starters,"\\1<stop> \\2",text)
    text = re.sub(alphabets + "[.]" + alphabets + "[.]" + alphabets + "[.]","\\1<prd>\\2<prd>\\3<prd>",text)
    text = re.sub(alphabets + "[.]" + alphabets + "[.]","\\1<prd>\\2<prd>",text)
    text = re.sub(" "+suffixes+"[.] "+starters," \\1<stop> \\2",text)
    text = re.sub(" "+suffixes+"[.]"," \\1<prd>",text)
    text = re.sub(" " + alphabets + "[.]"," \\1<prd>",text)
    if "”" in text: text = text.replace(".”","”.")
    if "\"" in text: text = text.replace(".\"","\".")
    if "!" in text: text = text.replace("!\"","\"!")
    if "?" in text: text = text.replace("?\"","\"?")
    if ";" in text: text = text.replace(";\"","\";")
    text = text.replace(".",".<stop>")
    text = text.replace(";",";<stop>")
    text = text.replace("?","?<stop>")
    text = text.replace("!","!<stop>")
    text = text.replace("<prd>",".")
    sentences = text.split("<stop>")
    sentences = sentences[:-1]
    sentences = [s.strip() for s in sentences]
    return sentences
```

In [0]:
```
sent = []
i = 0
for s in all_offres:
  i = i + 1
  sen = split_into_sentences(s)
  for ss in sen:
    sent.append(["Job"+str(i),ss])

sent
```

In [0]:
```
sent0 = []
sent1 = []
for s in sent:
  sent0.append(s[0])
  sent1.append(s[1])
sentence=pd.DataFrame({"Document":sent0,"Sentence":sent1})
sentence.to_csv("sentence.csv",sep=";")
```

In [35]:
```
sentence.head()
```

Out[35]:

	Document	Sentence
0	Job1	3d graphics software engineer delair delair ...
1	Job1	we enable enterprises to monitor and digitize ...
2	Job1	our solutions are used globally by customers i...
3	Job1	by joining delair, you will participate in wha...
4	Job1	the combination of drones, cloud-based service...

In [0]:
```
```