In [4]:

```
!pip install konlpy
```

```
Collecting konlpy
  Using cached https://files.pythonhosted.org/packages/e5/3d/4e983cd98d87b50b2ab03
87d73fa946f745aa8164e8888a714d5129f9765/konlpy-0.5.1-py2.py3-none-any.whl
Requirement already satisfied: JPype1>=0.5.7 in c:WprogramdataWanaconda3WlibWsite-
packages (from konlpy) (0.6.3)
Installing collected packages: konlpy
Successfully installed konlpy-0.5.1
```

In [3]:

```
!pip install JPype1-0.6.3-cp37-cp37m-win_amd64.whl
```

```
Processing f:Wjpype1-0.6.3-cp37-cp37m-win_amd64.whl
Installing collected packages: JPype1
Successfully installed JPype1-0.6.3
```

In [24]:

```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:WprogramdataWanaconda3WlibWsite-packages
(3.4)
Requirement already satisfied: six in c:WprogramdataWanaconda3WlibWsite-packages
(from nltk) (1.12.0)
Requirement already satisfied: singledispatch in c:WprogramdataWanaconda3WlibWsite
-packages (from nltk) (3.4.0.3)
```

In [25]:

```
import nltk

nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:WUsersWUSERWAppDataWRoamingWnltk_data...
[nltk_data]   Unzipping tokenizersWpunkt.zip.
```

Out[25]:

```
True
```

In [26]:

```
import nltk

nltk.download('brown')
nltk.download('gutenberg')
```

```
[nltk_data] Downloading package brown to
[nltk_data]     C:WUsersWUSERWAppDataWRoamingWnltk_data...
[nltk_data]   Unzipping corporaWbrown.zip.
[nltk_data] Downloading package gutenberg to
[nltk_data]     C:WUsersWUSERWAppDataWRoamingWnltk_data...
[nltk_data]   Unzipping corporaWgutenberg.zip.
```

Out[26]:

```
True
```

In [27]:

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:WUsersWUSERWAppDataWRoamingWnltk_data...
[nltk_data]   Unzipping corporaWstopwords.zip.
```

Out[27]:

```
True
```

In [68]:

```
nltk.download('Text')
```

```
[nltk_data] Error loading Text: Package 'Text' not found in index
```

Out[68]:

```
False
```

Normalization

1. 대소문자 통합(소문자)

2. 구두점 처리(I'd, I'm) => tokenizing => 대안: 형태소 분석기

3. 불용어(stopwords) 처리

In [6]:

```
sentence = "I'd like to learn more somthing."

sentence.lower() #(1번 처리)
```

Out[6]:

```
"i'd like to learn more somthing."
```

In [10]:

```
from string import punctuation
import re

print(punctuation) #ex ) Finland's => finland, finlands which?
#한국어 ex) '오늘'의 => 오늘 의, 오늘의 which?
print(re.escape(punctuation))
print(re.sub("[{0}]".format(re.escape(punctuation)), "", sentence))
#Id로 나오는 부분이 문제가 될 수 있음
```

```
!"#$%&'()*+,-./:;<=>?@[W]^_`{|}~
W!W"W#W$W%W&W'W(W)W*W+W,W-W.W/W:W;W<W=W>W?W@W[WWW]W^_W`W{W|W}W~
Id like to learn more somthing Id like to learn Id
```

In [11]:

```
sentence.lower()
pattern = re.compile("[{0}]".format(re.escape(punctuation)))
pattern.sub("",sentence.lower())
re.sub("[{0}]".format(re.escape(punctuation)), "", sentence)
```

Out[11]:

```
'Id like to learn more somthing Id like to learn Id'
```

In [15]:

```
pattern.sub("",sentence.lower())
```

Out[15]:

```
'id like to learn more somthing'
```

In [29]:

```
import nltk
from nltk.tokenize import word_tokenize
print(" ".join(word_tokenize(sentence.lower())))
```

```
오늘은 ' 목'요일
```

In [35]:

```
sentence = "오늘은 '목'요일"
#pattern = re.compile("[{0}]".format(re.escape(punctuation)))
#print(pattern.sub("",sentence.lower()))
print(re.sub("[{0}]".format(re.escape(punctuation)), "", sentence))
```

```
오늘은 목요일
```

In [21]:

```
re.sub("Ws{2,}"," ", "i   d    sasdf")
```

Out[21]:

```
'i d sasdf'
```

In [34]:

```
from nltk.corpus import stopwords

stop = stopwords.open("english").read()

print(len(stop))
print(stop)
```

936
i
me
my
myself
we
our
ours
ourselves
you
you're
you've
you'll
you'd
your
yours
yourself
yourselves
he
him
his
himself
she
she's
her
hers
herself
it
it's
its
itself
they
them
their
theirs
themselves
what
which
who
whom
this
that
that'll
these
those
am
is
are
was
were
be
been
being
have
has
had
having
do
does
did
doing

a
an
the
and
but
if
or
because
as
until
while
of
at
by
for
with
about
against
between
into
through
during
before
after
above
below
to
from
up
down
in
out
on
off
over
under
again
further
then
once
here
there
when
where
why
how
all
any
both
each
few
more
most
other
some
such
no
nor
not
only
own

```
same
so
than
too
very
s
t
can
will
just
don
don't
should
should've
now
d
ll
m
o
re
ve
y
ain
aren
aren't
couldn
couldn't
didn
didn't
doesn
doesn't
hadn
hadn't
hasn
hasn't
haven
haven't
isn
isn't
ma
mightn
mightn't
mustn
mustn't
needn
needn't
shan
shan't
shouldn
shouldn't
wasn
wasn't
weren
weren't
won
won't
wouldn
wouldn't
```

In [43]:

```
sentence = "I love you."

# for _ in sentence.lower().split():
#     if _ in stop:
#         print("Skipped")
#     else:
#         print(_)

for _ in word_tokenize(sentence.lower()):
    if pattern.sub("", _) in stop:
# for _ in pattern.sub("", sentence.lower()).split():
#     if _ in stop:
        print("Skipped")
    else:
        print(_)
```

```
Skipped
love
Skipped
Skipped
```

In [42]:

```
sentence = "Beautiful is better than ugly."

for _ in word_tokenize(sentence.lower()):
    if pattern.sub("", _) in stop:
# for _ in pattern.sub("", sentence.lower()).split():
#     if _ in stop:
        print("Skipped")
    else:
        print(_)

#한국의 경우는 함부로 날릴 수 없음. 특히 1음절의 경우 ex) 이
#섞어 쓰는게 ngram. 한국어는 stopwords 만드는게 쉽지 않음
```

```
beautiful
Skipped
better
Skipped
ugly
Skipped
```

In [12]:

```
from nltk.corpus import gutenberg

corpus = gutenberg.open("austen-emma.txt").read()
len(word_tokenize(corpus)) #19만개

words = list()
for _ in word_tokenize(corpus.lower()):
    if pattern.sub("", _) not in stop:
        words.append(_)
        #print("Skipped")
#     else:
#         print(_)
len(words) #6만 9천개
```

Out[12]:

```
191781
```

In [56]:

```
korstop = {"은","는","이","가","께","을","를","고","께서","게","에게", "어"}

sentence = "어머니 는 짜장면 이 싫다 고 하셨 어."
sentence = pattern.sub("", sentence)


print(len(word_tokenize(sentence)))
print(word_tokenize(sentence))
print(len([_ for _ in word_tokenize(sentence) if _ not in korstop]))
[_ for _ in word_tokenize(sentence) if _ not in korstop]
```

```
8
['어머니', '는', '짜장면', '이', '싫다', '고', '하셨', '어']
4
```

Out[56]:

```
['어머니', '짜장면', '싫다', '하셨']
```

In [57]:

```
#불용어 처리를 최소화하는 것이 좋음
#ex ) to be or not to be
```

길이 정규화

In [59]:

```
sentence = "I'd like to learn more somthing."
for _ in pattern.sub("", sentence.lower()).split():
    if 2 < len(_) < 6:
        print(_)

#정규식
#정규식 연습 = https://regexr.com
minimum = 3
maximum = 5
pattern2 = re.compile(r"WbWw{%d,%d}Wb" % (minimum, maximum))
#앞에 r은 이스케이프처리하지 않도록
pattern2.findall(sentence)
```

```
like
learn
more
```

Out[59]:

```
['like', 'learn', 'more']
```

빈도 정규화

In [13]:

```
from nltk import Text

sentence = "I'd like to learn more somthing. I'd like to learn. I'd"
obj = Text(word_tokenize(pattern.sub("", sentence.lower())))
for _ in obj.vocab():
    if 1 < obj.vocab().get(_) < 3:
        print(_, obj.vocab().get(_))
```

```
like 2
to 2
learn 2
```

In [19]:

```
original = Text(word_tokenize(corpus))

original.vocab().most_common(50)
#len(set(original.vocab()))
original.vocab().N()
original.vocab().B()

lowercase = Text(word_tokenize(corpus.lower()))
print(Text(word_tokenize(corpus.lower())).vocab().N())
print(Text(word_tokenize(corpus.lower())).vocab().B())

punct1 =  Text(word_tokenize(pattern.sub("", corpus.lower())))
print(punct1.vocab().N(), punct1.vocab().B())

punct2 =  Text(word_tokenize(pattern.sub("", corpus.lower())))
print(punct2.vocab().N(), punct2.vocab().B())

stops = Text([_ for _ in word_tokenize(pattern.sub(" ", corpus.lower())) if _ not in stop])

print(stops.vocab().N(), stops.vocab().B())

original.vocab().most_common(10)

obj = Text(word_tokenize(pattern.sub("", corpus.lower())))

minimum = 3
pattern3 = re.compile(r"WbWw{%d,}Wb" % (minimum))
#앞에 r은 이스케이프처리하지 않도록
length = Text([_ for _ in word_tokenize(pattern.sub("", corpus.lower())) if _ not in stop and r
e.search(r"WbWw{4,}Wb", _)])

print(length)

#freq = [(_, f)]
# K = pattern3.findall(corpus)

# print(K.vocab().N(), K.vocab().B())

# for _ in obj.vocab():
#     if obj.vocab().get(_) < 10:
#         print(_, obj.vocab().get(_))
```

```
191781
7944
158270 9311
158270 9311
162122 7102
<Text: emma jane austen 1816 volume chapter emma woodhouse...>
```

필터링

In [1]:

```
# => lexicon resource (X, 돈)

def ngram(data, n=2):
    result = defaultdict(int)

    for term, freq in data.items():
        tokens = term.split()
        for i in range(len(tokens) - (n-1)):
            result[' '.join(tokens[i:i+n])] += freq
    return result

def umjeol(text, n =2):
    ngram = list()

    for i in range(len(text)-(n-1)):
        ngram.append(''.join(text[i:i+n]))
    return ngram

stop = []
sentence = ""
result = list()

[_ for _ in sentence.split() if _ not in stop]
for _ in sentence.split():
#     if _ not in stop:
    if not re.search(stop[0], re.sub(r"Wb[0-9+Wb]", "", _)):
        result.append(_)
    else:
        result.append("*"*len(_))

" ".join(result)

for _ in sentence.split():
    for ngram in umjeol(_):
        if ngram in stop:
            flag = True

    if not flag:
        result.append(_)

    else:
        result.append("*", len(_))
```

In [ ]:

```
data = {
    splitTerm('low'):5,
    splitTerm('lowest'):2,
    splitTerm('newer'):6,
    splitTerm('wider'):3
}
for _ in range(5):
    bigram = ngram(data)
    maxKey = max(bigram, key=bigram.get)
    data = mergerNgram(maxKey, data)
print(data)

pattern= defaultdict()
for _ in data:
    for token n _.split():
        pattern[token] += data[_]
pattern
```

In [2]:

```
# stopwords => list (dictionary)
# BPE
# tokenizing
from nltk.tag.stanford import StanfordPOSTagger

MODEL = r"C:WUsersWJWDesktopWstanford-postagger-full-2018-10-16Wstanford-postagger-full-2018-10-16WmodelsWenglish-bidirectional-distsim.tagger"
PARSER = r"C:WUsersWJWDesktopWstanford-postagger-full-2018-10-16Wstanford-postagger-full-2018-10-16Wstanford-postagger-3.9.2.jar"
pos = StanfordPOSTagger(MODEL, PARSER)
```

In [ ]:

```

```