TF-IDF Calculation

Ko, Youngjoong

Sungkyunkwan University

nlp.skku.edu, nlplab.skku.edu

Contents

- ❖ NLTK Part-Of-Speech(POS) Tag
- ❖ Count based Word Representation
  - Bag of Words(BoW)
  - TF-IDF
- ❖ Assignment

POS Tag

❖ POS Tag

- A POS tag is a label assigned to each word in a text to indicate the part of speech.
  Ex) subject, verb, object, etc.

- In general, main components of a sentence are subject, verb, object, and complement, and these are usually verbs or nouns.

- **In this assignment, we use only verbs and nouns tags.**
  - Verb : (VB, VBD, VBG, VBN, VBP, VBZ)
  - Noun : (NN, NNS, NNP, NNPS)

  (See page 4 for more details)

POS Tag

❖ NLTK POS Tag

Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	*there* is
FW	foreign word	d'hoevre
IN	preposition/subordinating conjunction	in, of, like
JJ	adjective	big
JJR	adjective, comparative	bigger
JJS	adjective, superlative	biggest
LS	list marker	1)
MD	Modal	could, will
NN	noun, singular or mass	Door
NNS	noun plural	Doors
NNP	proper noun, singular	John
NNPS	proper noun, plural	Vikings
PDT	Predeterminer	*both* the boys
POS	possessive ending	friend*'s*
PRP	personal pronoun	I, he, it
PRP$	possessive pronoun	my, his
RB	Adverb	however, usually
RBR	adverb, comparative	Better
RBS	adverb, superlative	Best
RP	Particle	give *up*
TO	To	*to* go, *to* him
UH	Interjection	Uhhuhhuhh
VB	verb, base form	Take
VBD	verb, past tense	Took
VBG	verb, gerund/present participle	Taking
VBN	verb, past participle	Taken
VBP	verb, sing. present, non-3d	Take
VBZ	verb, 3rd person sing. Present	Takes
WDT	wh-determiner	Which
WP	wh-pronoun	who, what
WP$	possessive wh-pronoun	Whose
WRB	wh-abverb	where, when

Count based Word Representation

❖ Bag of Words (BoW)

- Numerical expression of text data taking account into the frequency of words without considering the order of words.

Count based Word Representation

❖ Bag of Words (BoW) based Document Features Extraction Method

- Doc0: I/PRP am/VBP a/DT boy/NN
- Doc1: I/PRP am/VBP a/DT girl/NN

1) Remove duplicates from all words in Doc0 and Doc1, and extract verbs and nouns. Then you will get a list which is sorted by alphabet:

   - [‘am/VBP‘, ‘boy/NN’, ’girl/NN’]

2) Give a unique number (index) to each word in the list.

   - { ‘am/VBP‘ : 0, ‘boy/NN’ : 1, ‘girl/NN’ : 2 }

3) Count the number of occurrences of each word in each document.

   - Index : 0 1 2
   - Doc0 : [1, 1 0]
   - Doc1 : [1, 0, 1]

Count based Word Representation

❖ TF-IDF (Term Frequency-Inverse Document Frequency)

- TF(Term Frequency)
  : The number of times that term t occurs in document d ($tf_{t,d}$)
- DF(Document Frequency)
  : The number of documents that contain the term t (df_t)
- IDF(Inverse Document Frequency)
  : Inverse value of DF.

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(tf_{t,d})$	t (idf)	$\log\frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\cdots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times tf_{t,d}}{\max_t tf_{t,d}}$	p (prob idf)	$\max\left\{0,\log\frac{N-df_t}{df_t}\right\}$	u (pivoted unique)	$\frac{1}{u}$
b (boolean)	$\begin{cases}1 & if\ \ tf_{t,d}>0\\ 0 & otherwise\end{cases}$			b (byte size)	$\frac{1}{CharLength^{\alpha}}, \alpha<1$
L (log ave)	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t\in d}(tf_{t,d}))}$				

Count based Word Representation

❖ TF (Term Frequency)

- Doc0 : I/PRP am/VBP a/DT boy/NN
- Doc1 : I/PRP am/VBP a/DT girl/NN
- Doc2 : who/WP is/VBZ a/DT boy/NN

- $tf_{t,d}$: The number of times that term t occurs in document d ($tf_{t,d}$)

	am/VBP	boy/NN	girl/NN	is/VBZ
Doc0	1	1	0	0
Doc1	1	0	1	0
Doc2	0	1	0	1

Count based Word Representation

❖ IDF (Inverse Document Frequency)

- Doc0 : I/PRP am/VBP a/DT boy/NN
- Doc1 : I/PRP am/VBP a/DT girl/NN
- Doc2 : who/WP is/VBZ a/DT boy/NN
  - $log\frac{N}{df_t}$: Inverse value of DF.
  - N : Total number of document
  - df_t : The number of documents that contain the term t (df_t)

  Ex) am/VBP = $log\frac{3}{2} = 0.4054$, girl/NN= $log\frac{3}{1} = 1.0986$

	am/VBP	boy/NN	girl/NN	is/VBZ
IDF	0.4054	0.4054	1.0986	1.0986

Count based Word Representation

❖ TF-IDF(Term Frequency Inverse Document Frequency)

- $tf_{t,d} \times log \frac{N}{df_t}$

Ex) am/VBP : $(1 \times 0.4054) = 0.4054$

	am/VBP	boy/NN	girl/NN	is/VBZ
Doc0	0.4054	0.4054	0	0
Doc1	0.4054	0	1.0986	0
Doc2	0	0.4054	0	1.0986

Assignment

❖ Assignment

1) Use ‘bbc_articles.json’ which was attached in assignment 2 as input file.

- See page 16 for JSON input file format.

2) **Calculate the TF-IDF vectors** of 300 articles taking into accounts only nouns and verbs from the given data.

- Verbs (VB, VBD, VBG, VBN, VBP, VBZ) and nouns (NN, NNS, NNP, NNPS)
- See page 4 for POS tags

3) You are allowed to use only NLTK for morpheme analysis.

4) You are **not allowed to use any libraries (Ex. Scikit-learn) for calculating TF-IDF** on this assignment.

5) You have to create an output text file by the submitted code (file I/O).

- See page 17 for an example of output file.

Assignment

❖ TF-IDF calculation

- Use following formula to calculate TF

$$TF = tf_{t,d}$$

- Use following formula to calculate IDF. (log is natural logarithm)

$$IDF = log \frac{N}{df_t}$$

- Calculate TF-IDF using above two formulas, and then normalize the TF-IDF vector using L2 norm.

L2 Norm : $\|x\|_2 := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \quad where \quad x = (x_1, x_2, \cdots, x_n)$

.

Assignment

❖ Cautions

- Teaching assistants have been checking out “Copy” to protect plagiarism by a copy detecting system. 0 scores will be assigned to all “Copy” results.
- Calculate TF-IDF for all distinguished “word/POS tag” when their POS tags are different even though the words are same.
  Ex) ‘account/NNS’ and ‘account/VBS’ are different.
- Convert all words into lower case after morpheme analysis and before calculating TF-IDF.
- Round down to the fourth digit after the decimal point.
  Ex) 0.45937 → 0.4593 / 0.731435 → 0.7314
- Upload the input file to the source code folder as shown on the right.

- All paths in the code use a relative path as below.
  Ex)

```
with open('./bbc_articles.json') as f
```

Assignment

❖ Cautions

- Remove all duplicated words extracted from all articles to make the distinguished word list for the whole data. Then sort words in the word list with using sorted() function. Calculate TF-IDF after sorting.

  - Ex) POS Tagging

```
Article1 = ['accounts/NNS', 'accounts/VBS', 'accounts/VBS']      # List
Article2 = ['company/NN', 'account/NN', 'accounts/VBS']   # List
Article3 = ['accounts/NNS', 'account/NN']                 # List
```

Remove all duplicated words

```
all_token = ('accounts/NNS', 'company/NN', 'accounts/VBS', 'account/NN')        # Set
```

Sort the output

```
sort_result = ['account/NN', 'accounts/NNS', 'accounts/VBS', 'company/NN']      # List
```

Give a unique number (index) to each word in the list.

```
w2i = {'account/NN' : 0, 'accounts/NNS' : 1, 'accounts/VBS' : 2, 'company/NN' : 3}  # Dict
```

Calculate TF-IDF

```
      Index              0          1          2          3
Article1_TFIDF = [       0          0.4472     0.8944     0          ]
Article2_TFIDF = [       0.3271     0          0.3271     0.8865     ]
Article3_TFIDF = [       0.7071     0.7071     0          0          ]
```

Assignment

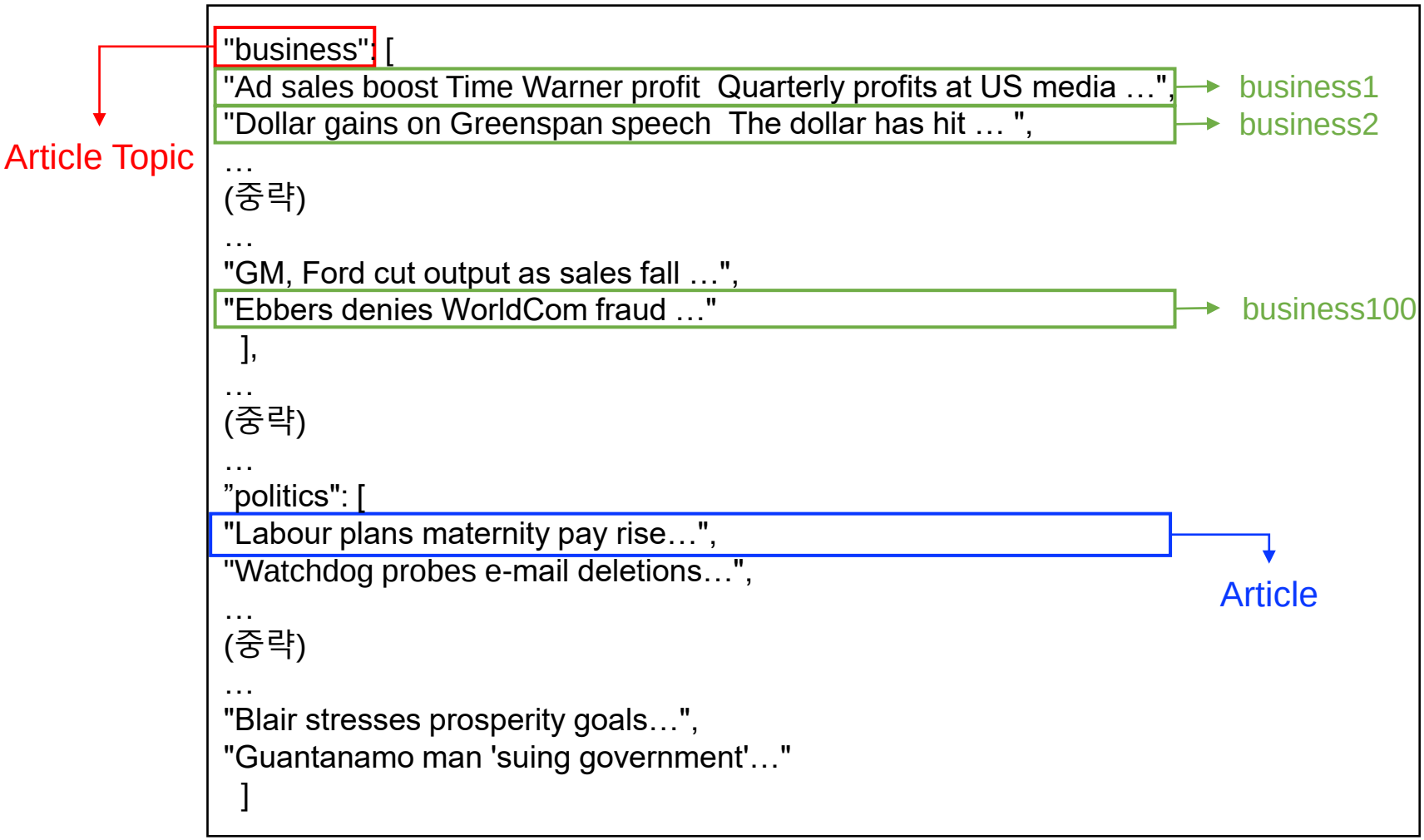
❖ Submission File

1) Python code file (.py) (python version 3.x, not .ipynb file)
   - Format: “Student Number_Name_TFIDF_raw.py”.
     - Ex) “2020000000_홍길동_TFIDF_raw.py”
   - **You will get 0 score if you submit the code with any other format such as .ipynb.**
   - Should develop your code **on Colab**
   - You will get **0 score if** you submit different outputs from different environments (not Colab).

2) TEXT file (.txt)
   - Format: “Student Number_Name_TFIDF_raw.txt”.
   - You will get **0 score if you copy and paste the results using the print() function.** This means that your code creates .txt file by file I/O mechanism as below.

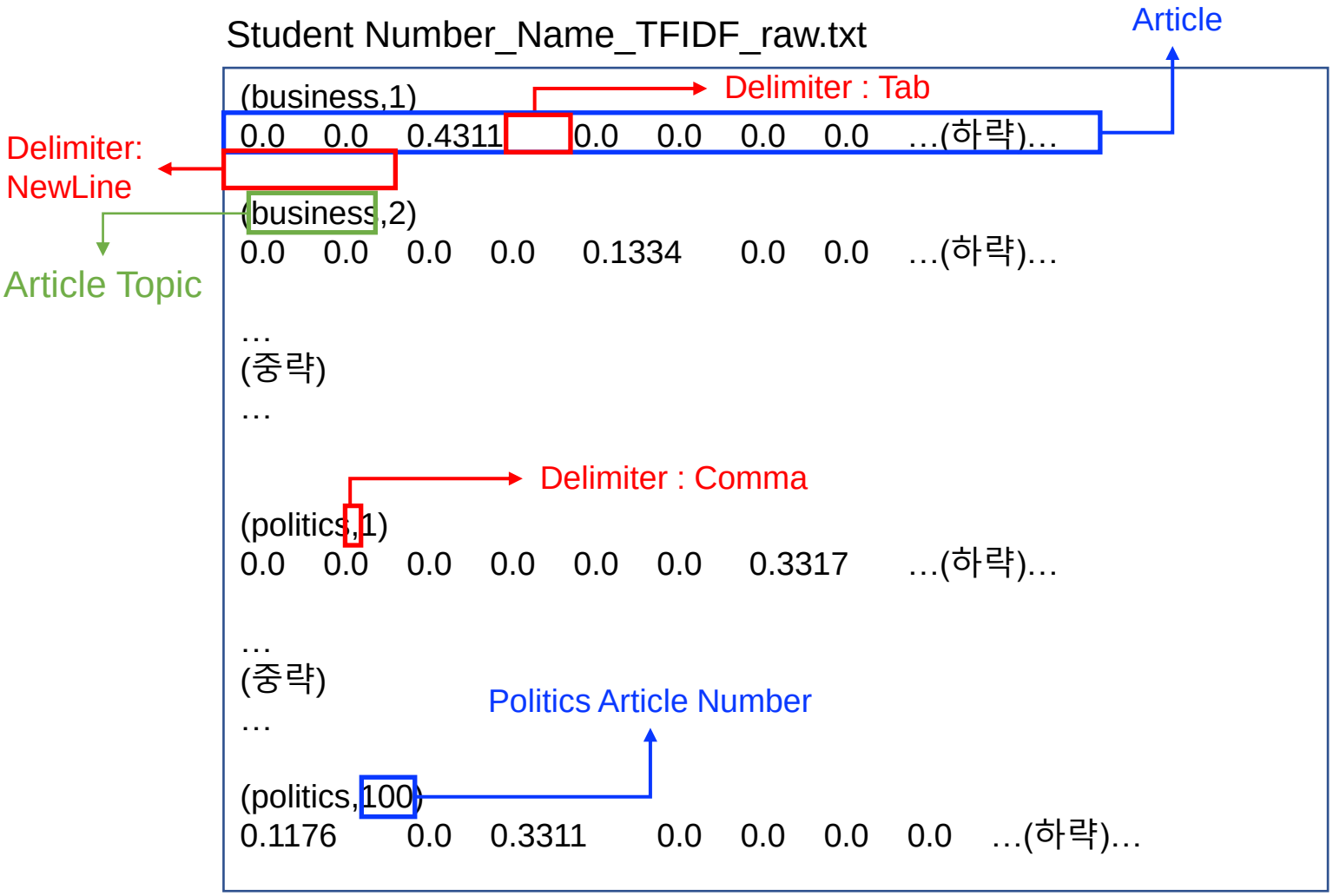
Ex)
```
fw = open('./Student-Number_NAME.txt', 'w', encodig='UTF-8')
fw.write(result)
fw.close()
```

JSON Input File



- JSON input file includes 300 articles on business, politics, and tech topics.

An Example of Output File



- This is an example.
- This could be different with the actual results.