Text Preprocessing

(do not ask for whom the bell tolls)

MSDS 7337 - Natural Language Processing - Homework 03

Patrick McDevitt

28-Sep-2018

PreProcessing : Edit distances, Stop words, and Stemming

For this project we are requested to :

1. Compare your given name with your nickname (if you don't have a nickname, invent one for this assignment) by answering the following questions:
   a. What is the edit distance between your nickname and your given name?
   b. What is the percentage string match between your nickname and your given name ?
   Show your work for both calculations.
2. Find a friend (or family member or classmate) who you know has read a certain book. Without your friend knowing, copy the first two sentences of that book. Now rewrite the words from those sentences, excluding stop words. Now tell your friend to guess which book the words are from by reading them just that list of words. Did you friend correctly guess the book on the first try? What did he or she guess? Explain why you think you friend either was or was not able to guess the book from hearing the list of words.
3. Run one of the stemmers available in Python. Run the same two sentences from question 2 above through the stemmer and show the results. How many of the outputted stems are valid morphological roots of the corresponding words? Express this answer as a percentage.

---

1 - Edit Distances

Given name : Patrick
Nickname : Pat

a. What is the edit distance between your nickname and your given name ?

Action	letter	additional distance
delete	r	(+1)
delete	i	(+1)
delete	c	(+1)
delete	k	(+1)
edit distance		4

b. What is the percentage string match between your nickname and your given name ?

1	2	3	4	5	6	7
P	a	t	r	i	c	k
P	a	t	-	-	-	-
—	—	—	—	—	—	—
1	1	1	0	0	0	0
—	—	—	—	—	—	—

Percentage match = 3 / 7 = 42.9%

2 - Stop word elimination

Find a friend who you know has read a certain book. Without your friend knowing, copy the first two sentences of that book. Now rewrite the words from those sentences, excluding stop words. Now tell your friend to guess which book the words are from by reading them just that list of words.

> "He lay flat on the brown, pine-needled floor of the forest, his chin on his folded arms, and high overhead the wind blew in the tops of the pine trees. The mountainside sloped gently where he lay; but below it was steep and he could see the dark of the oiled road winding through the pass."
>
> — *For Whom the Bell Tolls*, Ernest Hemingway, 1940

After removing the stop words [using stop words as defined in NLTK package : stopwords.words('english')], the remaining tokens are :

```
##            lay |            flat |           brown |    pine-needled |
##          floor |          forest |            chin |          folded |
##           arms |            high |        overhead |            wind |
##           blew |            tops |            pine |           trees |
##   mountainside |          sloped |          gently |             lay |
##          steep |           could |             see |            dark |
##          oiled |            road |         winding |            pass |
```

Did you friend correctly guess the book on the first try? What did he or she guess ?

Well, the book from which these two entences came was not recognized. Forced to name a book title from which these words came, my collaborator stated : "*Last of the Mohicans*" by James Fenimore Cooper, which I consider to be not such a bad guess, considering the text content.

Explain why you think you friend either was or was not able to guess the book from hearing the list of words.

Several contributors to why the book title was not guessed :

1. there is no proper noun included to identify place or person
2. the only verbs are "blew" (ostensibly associated to the wind) and "see" and "lay" which are potentially associated to a person ("chin" and "folded" "arms"), so there is no uniquely discernible action described that places this in a specific context.
3. the literary quality and style of the writer is removed when the stop words are extracted. Even if the title of the book is not recalled, it might have been possible to recognize the stylistic way in which Hemingway initiates a novel with subtle yet tangible tension and drama even while describing an

otherwise characteristically banal setting. That writing is achieved by the interaction among all the words - function and content words.
4. this is not a book that my collaborator had read recently
5. my collaborator is not as big a fan of Hemingway as am I ;-)>

3 - Stemming

Run one of the stemmers available in Python. Run the same two sentences from question 2 above through the stemmer and show the results.

Porter Stemmer

```
##              lay |            flat |           brown |      pine-needl |           floor |
##           forest |            chin |            fold |             arm |            high |
##         overhead |            wind |            blew |             top |            pine |
##             tree |     mountainsid |           slope |          gentli |             lay |
##            steep |           could |             see |            dark |             oil |
##             road |            wind |            pass |
```

Snowball Stemmer

```
##              lay |            flat |           brown |      pine-needl |           floor |
##           forest |            chin |            fold |             arm |            high |
##         overhead |            wind |            blew |             top |            pine |
##             tree |     mountainsid |           slope |           gentl |             lay |
##            steep |           could |             see |            dark |             oil |
##             road |            wind |            pass |
```

Lemmatization

```
##              lay |            flat |           brown |    pine-needled |           floor |
##           forest |            chin |          folded |             arm |            high |
##         overhead |            wind |            blew |             top |            pine |
##             tree |    mountainside |          sloped |          gently |             lay |
##            steep |           could |             see |            dark |           oiled |
##             road |         winding |             pas |
```

How many of the outputted stems are valid morphological roots of the corresponding words? Express this answer as a percentage.

- **Porter Stem**
  - 3 of 28 stems are not valid morphological roots : pine-needl, mountainsid, gentli
  - –> 89.3% are valid morphological roots
- **Snowball Stem**
  - 3 of 28 stems are not valid morphological roots : pine-needl, mountainsid, gentl
  - –> 89.3% are valid morphological roots
- **Lemmatization**
  - 1 of 28 lemmas are not valid morphological roots : pas
  - –> 96.4% are valid morphological roots

The python code to produce the above are included in Appendices A and B.

References

[1] - http://www.nltk.org/book/
[2] - https://courses.cs.ut.ee/LTAT.01.001/2017_fall/uploads/Main/Lecture6.pdf

Appendix A - Remove stop words python script

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

def remove_stopwords(phrase) :

    stop_words = set(stopwords.words('english'))

    word_tokens = word_tokenize(phrase)

    filtered_sentence = [w for w in word_tokens if not w in stop_words]
    filtered_sentence = []

    for w in word_tokens:
        if w not in stop_words:
            filtered_sentence.append(w)

    return(filtered_sentence)
```

Appendix B - Stemming python script

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

def remove_stopwords(phrase) :

    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(phrase)
    filtered_sentence = [w for w in word_tokens if not w in stop_words]
    filtered_sentence = []
    for w in word_tokens:
        if w not in stop_words:
            filtered_sentence.append(w)

    return(filtered_sentence)


def get_stems(phrase, method) :

    prtr = nltk.stem.PorterStemmer()
    snob = nltk.stem.SnowballStemmer('english')
    lema = nltk.wordnet.WordNetLemmatizer()

    words_to_stem = remove_stopwords(phrase)
    stems = [w for w in words_to_stem]
    stems = []

    if method == 'porter' :
        for w in words_to_stem:
            stems.append(prtr.stem(w))

    elif method == 'snowball':
        for w in words_to_stem:
            stems.append(snob.stem(w))

    elif method == 'lemmatize':
        for w in words_to_stem:
            stems.append(lema.lemmatize(w))

    return (stems)
```