Vectorizing Raw Data: N-Grams

N-Grams

Creates a document-term matrix where counts still occupy the cell but instead of the columns representing single terms, they represent all combinations of adjacent words of length n in your text.

"NLP is an interesting topic"

n	Name	Tokens
2	bigram	["nlp is", "is an", "an interesting", "interesting topic"]
3	trigram	["nlp is an", "is an interesting", "an interesting topic"]
4	four-gram	["nlp is an interesting", "is an interesting topic"]

Read in text

```
import pandas as pd
import re
import string
import nltk
pd.set_option('display.max_colwidth', 100)

stopwords = nltk.corpus.stopwords.words('english')
ps = nltk.PorterStemmer()

data = pd.read_csv("SMSSpamCollection.tsv", sep='\t')
data.columns = ['label', 'body_text']
```

Create function to remove punctuation, tokenize, remove stopwords, and stem

```
def clean_text(text):
    text = "".join([word.lower() for word in text if word not in string.punctuation])
    tokens = re.split('\W+', text)
```

```
    text = " ".join([ps.stem(word) for word in tokens if word not in stopwords])
    return text

data['cleaned_text'] = data['body_text'].apply(lambda x: clean_text(x))
data.head()
```

	label	body_text	cleaned_text
0	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	free entri 2 wkli comp win fa cup final tkt 21st may 2005 text fa 87121 receiv entri questionstd...
1	ham	Nah I don't think he goes to usf, he lives around here though	nah dont think goe usf live around though
2	ham	Even my brother is not like to speak with me. They treat me like aids patent.	even brother like speak treat like aid patent
3	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	date sunday

Apply CountVectorizer (w/ N-Grams)

```
from sklearn.feature_extraction.text import CountVectorizer

ngram_vect = CountVectorizer(ngram_range=(2,2))
X_counts = ngram_vect.fit_transform(data['cleaned_text'])
print(X_counts.shape)
print(ngram_vect.get_feature_names())
```

```
(5567, 31260)
['008704050406 sp', '0089mi last', '0121 2025050', '01223585236 xx', '01223585334 cum', '0125698789 ring', '02 user', '020603 2nd
```

Apply CountVectorizer (w/ N-Grams) to smaller sample

```
data_sample = data[0:20]

ngram_vect_sample = CountVectorizer(ngram_range=(2,2))
X_counts_sample = ngram_vect_sample.fit_transform(data_sample['cleaned_text'])
print(X_counts_sample.shape)
print(ngram_vect_sample.get_feature_names())
```

```
(20, 198)
['09061701461 claim', '100 20000', '100000 prize', '11 month', '12 hour', '150pday 6day', '16 tsandc', '20000 pound', '2005 text
```

```
X_counts_df = pd.DataFrame(X_counts_sample.toarray())
X_counts_df.columns = ngram_vect_sample.get_feature_names()
X_counts_df
```

	09061701461 claim	100 20000	100000 prize	11 month	12 hour	150pday 6day	16 tsandc	20000 pound	2005 text	21st may	...	way meet	week free	win cash	win fa	winner valu	wkli comp	word claim
0	0	0	0	0	0	0	0	0	1	1		0	0	0	1	0	1	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
5	1	0	0	0	1	0	0	0	0	0	...	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
8	0	1	0	0	0	1	1	1	0	0	...	0	0	1	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0	...	0	1	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0