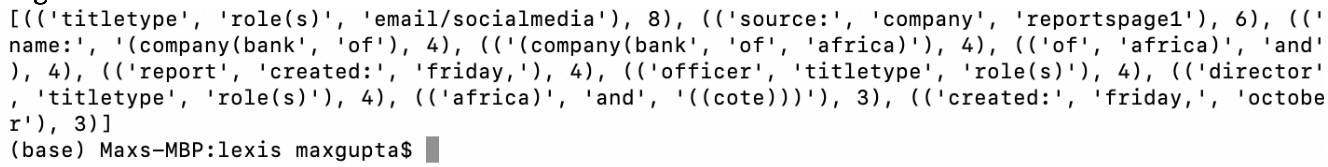
Natural Language Processing App for PDF Analysis

This app is designed to provide on-the-fly NLP analysis of financial documents ranging from Swift statements to Banker's Almanac files to Lexis Nexis and everything in between. The problem we have identified is a huge amount of manual effort flowing into the identification of key words and trends in banking documents to classify and group them based on content. At the client bank, we have identified a team of dozens of KYC (Know Your Customer) specialists devoted to the filtering of these PDF's. To save time, this application was built to instantly filter an unlimited number of files and group key words and phrases (trigrams and longer).

Use Case: Lexis Nexis Docs

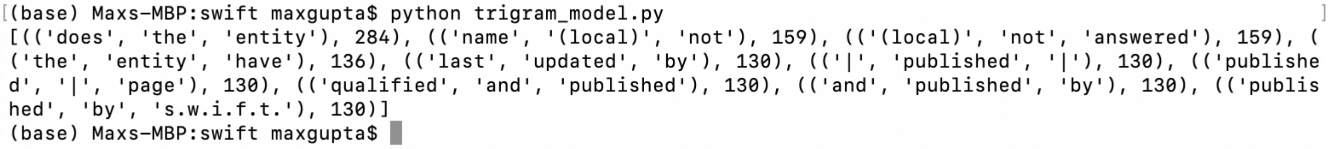
Trigrams



```
[(('titletype', 'role(s)', 'email/socialmedia'), 8), (('source:', 'company', 'reportspage1'), 6), (('
name:', '(company(bank', 'of'), 4), (('(company(bank', 'of', 'africa)'), 4), (('of', 'africa)', 'and'
), 4), (('report', 'created:', 'friday,'), 4), (('officer', 'titletype', 'role(s)'), 4), (('director'
, 'titletype', 'role(s)'), 4), (('africa)', 'and', '((cote)))'), 3), (('created:', 'friday,', 'octobe
r'), 3)]
(base) Maxs-MBP:lexis maxgupta$ 
```

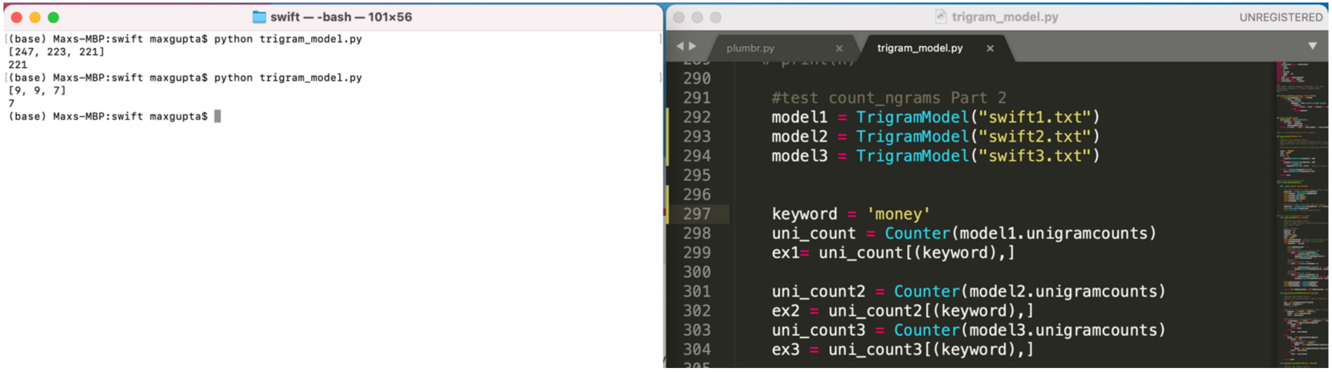
Use Case: Swift Docs

Trigrams



```
[(base) Maxs-MBP:swift maxgupta$ python trigram_model.py                                              ]
[(('does', 'the', 'entity'), 284), (('name', '(local)', 'not'), 159), (('(local)', 'not', 'answered'), 159), (
('the', 'entity', 'have'), 136), (('last', 'updated', 'by'), 130), (('|', 'published', '|'), 130), (('publishe
d', '|', 'page'), 130), (('qualified', 'and', 'published'), 130), (('and', 'published', 'by'), 130), (('publis
hed', 'by', 's.w.i.f.t.'), 130)]
(base) Maxs-MBP:swift maxgupta$ 
```

Example of lowest common occurrence with key word: "money"



```
[(base) Maxs-MBP:swift maxgupta$ python trigram_model.py                ]
[247, 223, 221]
221
[(base) Maxs-MBP:swift maxgupta$ python trigram_model.py                ]
[9, 9, 7]
7
(base) Maxs-MBP:swift maxgupta$ 
```

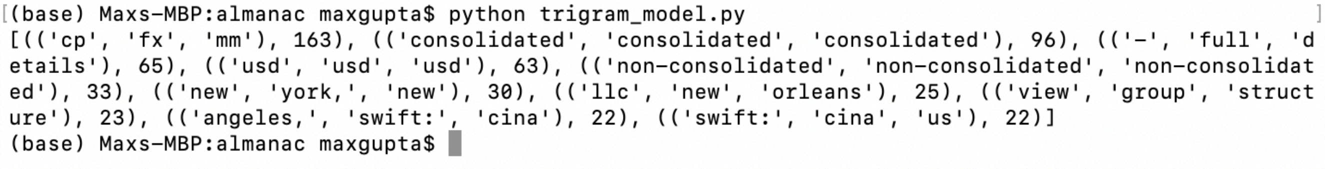
```
    #test count_ngrams Part 2
    model1 = TrigramModel("swift1.txt")
    model2 = TrigramModel("swift2.txt")
    model3 = TrigramModel("swift3.txt")


    keyword = 'money'
    uni_count = Counter(model1.unigramcounts)
    ex1= uni_count[(keyword),]

    uni_count2 = Counter(model2.unigramcounts)
    ex2 = uni_count2[(keyword),]
    uni_count3 = Counter(model3.unigramcounts)
    ex3 = uni_count3[(keyword),]
```

Use Case: Banker's Almanac Docs

Trigrams



```
[(base) Maxs-MBP:almanac maxgupta$ python trigram_model.py                                      ]
[(('cp', 'fx', 'mm'), 163), (('consolidated', 'consolidated', 'consolidated'), 96), (('-', 'full', 'd
etails'), 65), (('usd', 'usd', 'usd'), 63), (('non-consolidated', 'non-consolidated', 'non-consolidat
ed'), 33), (('new', 'york,', 'new'), 30), (('llc', 'new', 'orleans'), 25), (('view', 'group', 'struct
ure'), 23), (('angeles,', 'swift:', 'cina'), 22), (('swift:', 'cina', 'us'), 22)]
(base) Maxs-MBP:almanac maxgupta$ 
```