### USE CASE - Adverse News Media - LLM tokenizer

~~~
```python Sourced Data from various News media artciles
import pandas as pd
file1 = open(r"negativemedia_vocab.txt", "r+", encoding="utf-8")
#print("Output of Read function is ")
corpus = file1.read()
 #print(text)
 # check count of words
print("word count is -> ", len(corpus))
#print (type(corpus))
```
~~~

```
    word count is ->  22865
```

~~~
```python
#Strategy 1 : Split on white spaces
import re

text = corpus
result = re.split(r'(\s)', text)
print(result[10:100])
```
~~~

```
    ['jaypee', ' ', "group's", ' ', 'vast', ' ', 'apartment', ' ',
'complexes,', ' ', 'villas', ' ', 'golf', ' ', 'courses', ' ',
'national', ' ', 'capital', ' ', 'region,', ' ', 'making', ' ', 'grand',
' ', 'entry', ' ', "country's", ' ', 'biggest', ' ', 'property', ' ',
'market', ' ', 'insolvency', ' ', 'court.', ' ', 'according', ' ',
'people', ' ', 'familiar', ' ', 'matter,', ' ', 'adani', ' ', 'spend', '
', '$1', ' ', 'billion', ' ', 'real', ' ', 'estate', ' ', 'assets', ' ',
'jaypee,', ' ', "country's", ' ', 'biggest', ' ', 'bankruptcy', ' ',
'case', ' ', 'involving', ' ', '₹50,000', ' ', 'crore', ' ', 'bank', ' ',
'loans.', '\n', 'mumbai', ' ', 'bench', ' ', 'national', ' ', 'company',
' ']
```

~~~
```python
# Strategy 2 : Split on white space or comma or period'  ##simple
tokenizer
text = corpus
result = re.split(r'([,.:;?_!~#"()\']|--|\s)', text)
print(result)

```
~~~

```
    ['adani', ' ', 'group', ' ', 'plans', ' ', 'bid', ' ', 'bankrupt', '
', 'jaypee', ' ', 'group', "'", 's', ' ', 'vast', ' ', 'apartment', ' ',
'complexes', ',', '', ' ', 'villas', ' ', 'golf', ' ', 'courses', ' ',
'national', ' ', 'capital', ' ', 'region', ',', '', ' ', 'making', ' ',
'grand', ' ', 'entry', ' ', 'country', "'", 's', ' ', 'biggest', ' ',
'property', ' ', 'market', ' ', 'insolvency', ' ', 'court', '.', '', ' ',
'according', ' ', 'people', ' ', 'familiar', ' ', 'matter', ',', '', ' ',
```

```
'adani', ' ', 'spend', ' ', '$1', ' ', 'billion', ' ', 'real', ' ',
'estate', ' ', 'assets', ' ', 'jaypee', ',', '', ' ', 'country', "'",
's', ' ', 'biggest', ' ', 'bankruptcy', ' ', 'case', ' ', 'involving', '
', '₹50', ',', '000', ' ', 'crore', ' ', 'bank', ' ', 'loans', '.', '',
'\n', 'mumbai', ' ', 'bench', ' ', 'national', ' ', 'company', ' ',
'law', ' ', 'tribunal', ' ', '', '(', 'nclt', ')', '', ' ', 'admits', '
', 'kishore', ' ', 'biyani-promoted', ' ', 'future', ' ', 'retail', ' ',
'ltd', ' ', 'liquidation', ' ', 'absence', ' ', 'viable', ' ', 'revival',
' ', 'plan', ' ', 'company', '.', '', ' ', 'tribunal', ' ', 'appointed',
' ', 'sanjay', ' ', 'gupta', ' ', 'company’s', ' ', 'liquidator', '.',
'', ' ', 'division', ' ', 'bench', ' ', 'judicial', ' ', 'member', ' ',
'kuldip', ' ', 'kumar', ' ', 'kareer', ' ', 'technical', ' ', 'member', '
', 'anil', ' ', 'raj', ' ', 'chellan', ' ', 'allowing', ' ', 'company’s',
' ', 'resolution', ' ', 'professional', ' ', 'vijaykumar', ' ', 'v', '.',
'', ' ', 'iyer’s', ' ', 'application', ' ', 'admit', ' ', 'company', ' ',
'liquidation', ',', '', ' ', 'observed', ' ', 'maximum', ' ', 'period', '
', 'cirp', ' ', 'expired', ' ', 'resolution', ' ', 'plan', ' ',
'approved', ' ', 'coc', ' ', '', '(', 'committee', ' ', 'creditors', ')',
'', '.', '', ' ', 'company', ' ', 'admitted', ' ', 'liabilities', ' ',
'rs', ' ', '28', ',', '452', ' ', 'crore', '.', '', ' ', 'includes', ' ',
'secured', ' ', 'financial', ' ', 'creditors', "'", '', ' ', 'claims', '
', 'rs', ' ', '14', ',', '422', ' ', 'crore', '.', '', ' ', '“we', ' ',
'considered', ' ', 'opinion', ' ', 'fit', ' ', 'case', ' ',
'liquidation', ',', '”', ' ', 'said', ' ', 'tribunal', '.', '', ' ', '“',
'.', '', ' ', 'maximise', ' ', 'value', ' ', 'corporate', ' ', 'debtor',
',', '', ' ', 'liquidator', ' ', 'shall', ' ', 'endeavour', ' ', 'sale',
' ', 'corporate', ' ', 'debtor', ' ', 'going', ' ', 'concern', ' ',
'regulation', ' ', '32a', ' ', 'clause', ' ', '', '(', 'e', ')', '', ' ',
'insolvency', ' ', 'bankruptcy', ' ', 'board', ' ', 'india', ' ', '',
'(', 'liquidation', ' ', 'process', ')', '', ' ', 'regulation', ',', '',
' ', '2016', ',', '”', ' ', 'said', ' ', 'tribunal', ' ', 'order', '.',
'', ' ', 'year', ' ', 'november', ',', '', ' ', 'company’s', ' ', 'rp', '
', 'informed', ' ', 'stock', ' ', 'exchange', ' ', 'lenders', ' ',
'rejected', ' ', 'space', ' ', 'mantra', ' ', 'pvt', ' ', 'ltd’s', ' ',
'resolution', ' ', 'plan', ' ', 'future', ' ', 'retail', ' ', 'lenders',
' ', 'decided', ' ', 'admit', ' ', 'company', ' ', 'liquidation', '.',
'', ' ', 'trouble', ' ', 'future', ' ', 'group', ' ', 'initially', ' ',
'started', ' ', 'nationwide', ' ', 'lockdown', ' ', 'march', ' ', '2020',
'.', '', ' ', 'malls', ' ', 'future', ' ', 'stores', ' ', 'located', ' ',
'remained', ' ', 'shut', '.', '', ' ', 'subsequently', ',', '', ' ',
'suffered', ' ', 'setback', ' ', 'april', ' ', '2022', ' ', 'failed', '
', 'clinch', ' ', 'rs', ' ', '24', ',', '713-crore', ' ', 'deal', ' ',
'reliance', ' ', 'industries', ' ', 'sell', ' ', 'retail', ' ',
'wholesale', ' ', 'business', ' ', 'slump', ' ', 'sale', '.', 'future’s',
' ', 'deal', ' ', 'reliance', ' ', 'collapsed', ' ', 'secured', ' ',
'lenders', ' ', 'voted', ' ', 'scheme', ' ', 'arrangement', '.', '', ' ',
'lenders', ' ', 'said', ' ', 'receive', ' ', 'clarity', ' ', 'money', '
', 'reliance', ' ', 'took', ' ', 'possession', ' ', '900', ' ', 'large-
format', ' ', 'future', ' ', 'retail', ' ', 'stores', '.', '', ' ',
'reliance', ' ', 'took', ' ', 'stores', ' ', 'phased', ' ', 'manner',
',', '', ' ', 'beginning', ' ', '2022', ',', '', ' ', 'non-payment', ' ',
'rentals', ' ', 'future', ' ', 'group', '.', '', '\n', 'national', ' ',
'company', ' ', 'law', ' ', 'appellate', ' ', 'tribunal', ' ', '', '(',
'nclat', ')', '', ' ', 'judge', ' ', 'monday', ' ', 'recused', ' ',
'hearing', ' ', 'plea', ' ', 'seeking', ' ', 'interim', ' ', 'stay', ' ',
'insolvency', ' ', 'proceedings', ' ', 'think', ' ', 'learn', ',', '', '
', 'parent', ' ', 'company', ' ', 'edtech', ' ', 'firm', ' ', 'byju',
'.', 'the', ' ', 'hearing', ' ', 'plea', ' ', 'filed', ' ', 'byju', ' ',
'raveendran', ',', '', ' ', 'founder', ' ', 'company', ',', '', ' ',
'adjourned', '.', 'while', ' ', 'recusing', ' ', 'hearing', ',', '', ' ',
'justice', ' ', 'sharad', ' ', 'kumar', ' ', 'sharma', ' ', 'nclat', ' ',
'chennai', ' ', 'bench', ' ', 'said', ' ', 'appeared', ' ', 'board', ' ',
'control', ' ', 'cricket', ' ', 'india', ' ', '', '(', 'bcci', ')', '', '
', 'times', ' ', 'lawyer', ' ', 'and', ',', '', ' ', 'therefore', ',',
'', ' ', 'think', ' ', 'appropriate', ' ', 'hear', ' ', 'matter', ',',
'', ' ', 'according', ' ', 'law', ' ', 'platform', ' ', 'bar', ' ',
'bench', '.', 'a', ' ', 'recent', ' ', 'order', ' ', 'national', ' ',
'company', ' ', 'law', ' ', 'tribunal', ' ', '', '(', 'nclt', ')', '', '
', 'allowed', ' ', 'indian', ' ', 'cricket', ' ', 'board&rsquo', ';',
's', ' ', 'petition', ' ', 'initiating', ' ', 'insolvency', ' ',
'proceedings', ' ', 'edtech', ' ', 'company', ',', '', ' ', 'byju&rsquo',
';', 's', ' ', 'approached', ' ', 'nclat', '.', '&ldquo', ';', 'i', ' ',
'went', ' ', 'entire', ' ', 'case', ' ', 'papers', ' ', 'realised', ' ',
'ultimately', ',', '', ' ', 'beneficiary', ' ', 'going', ' ', 'bcci',
'.', '', ' ', 'so', ',', '', ' ', 'want', ' ', 'involved', ' ', 'this',
'.', '', ' ', 'refusing', ' ', 'hear', ' ', 'matter', '.', '', ' ',
'chairperson', ' ', 'decide', ' ', 'date', ',', '&rdquo', ';', '', ' ',
'bar', ' ', 'bench', ' ', 'report', ' ', 'cited', ' ', 'justice', ' ',
'sharma', ' ', 'saying', '.', 'after', ' ', 'matter', ' ', 'adjourned',
',', '', ' ', 'raveendran', ' ', 'mentioned', ' ', 'matter', ' ',
'nclat', ' ', 'delhi', ',', '', ' ', 'according', ' ', 'bar', ' ',
'bench', '.', 'he', ' ', 'asked', ' ', 'approach', ' ', 'nclat', ' ',
'registry', ' ', 'principal', ' ', 'bench', ' ', 'constitution', ' ',
'new', ' ', 'bench', ' ', 'date', ' ', 'hearing', '.',
'raveendran&rsquo', ';', 's', ' ', 'lawyers', ' ', 'write', ' ', 'nclat',
' ', 'registry', ' ', 'seeking', ' ', 'listing', ' ', 'matter', ' ',
'tuesday', ' ', 'bench', '.', 'raveendran&', '#', '39', ';', 's', ' ',
'counsel', ' ', 'argued', ' ', 'client', ' ', 'willing', ' ', '&ldquo',
';', 'pay', ' ', 'entire', ' ', 'riju', ' ', 'raveendran', ' ', '', '(',
'his', ' ', 'brother', ')', '&rdquo', ';', '', ',', '', ' ', 'promoter',
' ', 'think', ' ', 'learn', '.', 'however', ',', '', ' ', 'justice', ' ',
'sharma', ' ', 'said', ' ', 'raveendran', ' ', 'intended', ' ', 'argue',
' ', 'matter', ' ', 'merits', ',', '', ' ', 'inclined', ' ', 'hear', ' ',
'it', '.', 'on', ' ', 'july', ' ', '16', ',', '', ' ', 'national', ' ',
'company', ' ', 'law', ' ', 'tribunal', ' ', '', '(', 'nclt', ')', '', '
', 'admitted', ' ', 'byju&rsquo', ';', 's', ',', '', ' ', 'officially', '
', 'known', ' ', 'think', ' ', 'learn', ' ', 'pvt', ' ', 'ltd', ',', '',
' ', 'corporate', ' ', 'insolvency', ' ', 'resolution', ' ', 'process', '
', '', '(', 'cirp', ')', '', ' ', 'based', ' ', 'petition', ' ', 'filed',
' ', 'bcci', ' ', 'unpaid', ' ', 'dues', ' ', 'amounting', ' ', '', '~',
'158', '.', '90', ' ', 'crore', '.', 'due', ' ', 'nclt&', '#', '39', ';',
's', ' ', 'order', ',', '', ' ', 'raveendran', ' ', 'lost', ' ',
'immediate', ' ', 'control', ' ', 'company', '.', '', ' ', 'tribunal', '
', 'appointed', ' ', 'bankruptcy', ' ', 'professional', ' ', 'oversee', '
', 'daily', ' ', 'operations', ' ', 'proceedings', '.', 'on', ' ',
'july', ' ', '23', ',', '', ' ', 'raveendran', ' ', 'moved', ' ',
'nclat', ' ', 'seeking', ' ', 'urgent', ' ', 'hearing', ' ', 'matter',
'.', '', ' ', 'justice', ' ', 'sharma', ' ', 'questioned', ' ',
'raveendran', ' ', 'chosen', ' ', 'karnataka', ' ', 'high', ' ', 'court',
' ', 'first', ',', '', ' ', 'withdrew', ' ', 'petition', ' ',
'approached', ' ', 'nclat', ' ', 'afterthought', ',', '', ' ',
'according', ' ', 'bar', ' ', 'bench', '.', '', '\n', 'state-owned', ' ',
'oil', ' ', 'marketing', ' ', 'company', ' ', '', '(', 'omc', ')', '', '
', 'hpcl', ' ', 'monday', ' ', 'reported', ' ', 'massive', ' ', '90',
'.', '6', ' ', 'cent', ' ', 'fall', ' ', 'consolidated', ' ', 'net', ' ',
'profit', ' ', 'rs', ' ', '634', ' ', 'crore', ' ', 'quarter', ' ', '',
'(', 'april-june', ')', '', ' ', '2023-24', ' ', '', '(', 'fy25', ')',
'', '.', '', ' ', 'net', ' ', 'profit', ' ', 'fell', ' ', 'rs', ' ', '6',
',', '765', ' ', 'crore', ' ', 'q1', ' ', 'fy24', ' ', 'result', ' ',
'weak', ' ', 'gross', ' ', 'refining', ' ', 'margins', ' ', '', '(',
'grm', ')', '', ' ', 'elevated', ' ', 'costs', '.', 'on', ' ',
'sequential', ' ', 'basis', ',', '', ' ', 'net', ' ', 'profit', ' ',
'fell', ' ', '76', ' ', 'cent', ' ', 'rs', ' ', '2', ',', '709', '.',
'31', ' ', 'crore', ' ', 'registered', ' ', 'preceding', ' ', 'quarter',
'.', 'interestingly', ',', '', ' ', 'company&', '#', '39', ';', 's', ' ',
'total', ' ', 'income', ' ', 'rose', ' ', 'rs', ' ', '1', '.', '21', ' ',
'trillion', ',', '', ' ', 'rs', ' ', '1', '.', '19', ' ', 'trillion', '
', 'q1', '.', '', ' ', 'however', ',', '', ' ', 'drop', ' ', 'net', ' ',
'profits', ' ', 'total', ' ', 'expenses', ' ', 'rising', ' ', '8', '.',
'56', ' ', 'cent', ' ', 'rs', ' ', '1', '.', '21', ' ', 'trillion', ' ',
'q1', ',', '', ' ', 'rs', ' ', '1', '.', '11', ' ', 'trillion', ' ',
'q1', ' ', 'fy24', '.', '', ' ', 'cost', ' ', 'materials', ' ',
'consumed', ' ', 'rose', ' ', '18', ' ', 'cent', ' ', 'rs', ' ', '34',
',', '917', '.', '7', ' ', 'crore', ' ', 'quarter', ',', '', ' ', 'rs', '
', '29', ',', '397', '.', '5', ' ', 'crore', ' ', 'year', ' ', 'back',
'.', '', ' ', 'meanwhile', ',', '', ' ', 'company', ' ', 'purchased', '
', 'higher', ' ', 'stock-in-trade', ' ', 'worth', ' ', 'rs', ' ', '69',
',', '016', ' ', 'crore', ',', '', ' ', '9', '.', '34', ' ', 'cent', ' ',
'higher', ' ', 'quarter', ' ', 'previous', ' ', 'financial', ' ', 'year',
'.', 'the', ' ', 'company&', '#', '39', ';', 's', ' ', 'average', ' ',
'grm', ' ', 'q1', ' ', '$5', '.', '03', ' ', 'barrel', ' ', '', '(',
'bbl', ')', '', ',', '', ' ', '$7', ' ', 'bbl', ' ', 'q1', ' ', 'fy24',
'.', '', ' ', 'grm', ' ', 'refiners', ' ', 'earn', ' ', 'turning', ' ',
'barrel', ' ', 'crude', ' ', 'oil', ' ', 'refined', ' ', 'fuel', ' ',
'products', '.', 'hpcl', ' ', 'recorded', ' ', 'highest-ever', ' ',
'quarterly', ' ', 'sales', ' ', 'volume', ' ', '12', '.', '63', ' ',
'million', ' ', 'metric', ' ', 'tonnes', ' ', '', '(', 'mmt', ')', '', '
', '', '(', 'including', ' ', 'exports', ')', '', ' ', 'q1', ',', '', '
', '6', '.', '6', ' ', 'cent', ' ', 'higher', ' ', '11', '.', '85', ' ',
'mmt', ' ', 'year-ago', ' ', 'period', ',', '', ' ', 'company', ' ',
'said', '.', '', ' ', 'logged', ' ', 'quarterly', ' ', 'domestic', ' ',
'sales', ' ', '12', '.', '07', ' ', 'mmt', ' ', 'quarter', ',', '', ' ',
'11', '.', '43', ' ', 'mmt', ' ', 'year', '.', '', ' ', 'exports', ' ',
'quarter', ' ', 'came', ' ', '0', '.', '56', ' ', 'mmt', '.', '', ' ',
'throughput', ' ', 'hpcl', ' ', 'refineries', ' ', '5', '.', '76', ' ',
'mmt', ' ', 'quarter', ',', '', ' ', 'compared', ' ', '5', '.', '40', '
', 'mmt', ' ', 'year', '.', 'on', ' ', 'hand', ',', '', ' ', 'company', '
', 'recorded', ' ', 'highest-ever', ' ', 'petrochemical', ' ', 'sales', '
', '30', '.', '3', ' ', 'thousand', ' ', 'metric', ' ', 'tonnes', ' ',
'', '(', 'tmt', ')', '', ',', '', ' ', 'aviation', ' ', 'business', ' ',
'recorded', ' ', 'robust', ' ', 'growth', ' ', '31', ' ', 'cent', '.',
'hpcl', ' ', 'commissioned', ' ', '126', ' ', 'retail', ' ', 'outlets', '
', 'country', ',', '', ' ', 'taking', ' ', 'total', ' ', 'number', ' ',
'outlets', ' ', '22', ',', '148', '.', '', ' ', 'company', ' ',
'commissioned', ' ', '9', ' ', 'new', ' ', 'lpg', ' ',
'distributorships', ' ', 'period', ',', '', ' ', 'taking', ' ', 'total',
' ', 'count', ' ', 'lpg', ' ', 'distributorships', ' ', '6', ',', '358',
'.', 'hpcl', ' ', 'shares', ' ', 'closed', ' ', 'rs', ' ', '381', '.',
'20', ' ', 'bse', ',', '', ' ', '1', '.', '26', ' ', 'cent', ' ',
'higher', ' ', 'previous', ' ', 'day&', '#', '39', ';', 's', ' ',
'close', '.', '', '\n', 'adani', ' ', 'wilmar', ' ', 'reported', ' ',
'net', ' ', 'profit', ' ', 'rs', ' ', '313', ' ', 'crore', ' ', 'june', '
', 'quarter', ',', '', ' ', 'compared', ' ', 'loss', ' ', 'rs', ' ',
'79', ' ', 'crore', ' ', 'year-ago', ' ', 'period', ',', '', ' ',
'volume', ' ', 'sales', ' ', 'grew', ' ', 'q1fy25', '.', 'the', ' ',
'edible', ' ', 'oil', ' ', 'major', ' ', 'saw', ' ', 'net', ' ', 'sales',
' ', 'increase', ' ', '9', '.', '6', ' ', 'cent', ' ', 'rs', ' ', '14',
',', '169', ' ', 'crore', ' ', 'q1fy25', ' ', 'volume', ' ', 'growth', '
', 'quarter', ' ', 'stood', ' ', '12', ' ', 'cent', '.', '', ' ',
'adani', ' ', 'group&rsquo', ';', 's', ' ', 'pbidt', ' ', '', '(',
'profit', ' ', 'interest', ',', '', ' ', 'depreciation', ' ', 'tax', ')',
'', ' ', 'stood', ' ', 'rs', ' ', '680', ',', '', ' ', '246', '.', '1', '
', 'cent', ' ', 'compared', ' ', 'q1fy24', '.', 'edible', ' ', 'oil', '
', 'registered', ' ', 'strong', ' ', 'volume', ' ', 'growth', ' ', '12',
' ', 'cent', ' ', 'year-on-year', ' ', '', '(', 'y-o-y', ')', '', ' ',
'surpassed', ' ', '1', ' ', 'million', ' ', 'tonnes', ' ', 'quarter',
'.', '', ' ', 'food', ' ', 'fmcg', ' ', 'sales', ' ', 'crossed', ' ',
'rs', ' ', '1', ',', '500', ' ', 'crore', ' ', 'quarter', ',', '', ' ',
'underlying', ' ', 'volume', ' ', 'growth', ' ', '42', ' ', 'cent', ' ',
'y-o-y', ',', '', ' ', 'company', ' ', 'said', ' ', 'release', '.',
'&ldquo', ';', 'strong', ' ', 'business', ' ', 'momentum', ' ', 'led', '
', 'increased', ' ', 'market', ' ', 'share', ' ', 'key', ' ', 'product',
' ', 'categories', '.', '', ' ', 'edible', ' ', 'oils', ',', '', ' ',
'rocp', ' ', '', '(', 'refined', ' ', 'oil', ' ', 'consumer', ' ',
'pack', ')', '', ' ', 'market', ' ', 'share', ' ', 'adani', ' ',
'wilmar', ' ', 'increased', ' ', '60', ' ', 'bps', ' ', 'y-o-y', ' ',
'19', ' ', 'cent', ' ', 'moving', ' ', 'annual', ' ', 'total', ' ', '',
'(', 'mat', ')', '', ' ', 'basis', ',', '', ' ', 'wheat', ' ', 'flour',
',', '', ' ', 'market', ' ', 'share', ' ', 'increased', ' ', '90', ' ',
'bps', ' ', 'y-o-y', ' ', '5', '.', '9', ' ', 'cent', '.', '', ' ',
'additionally', ',', '', ' ', 'branded', ' ', 'exports', ' ', 'volume', '
', 'surged', ' ', '36', ' ', 'cent', ' ', 'y-o-y', ',', '&rdquo', ';',
'', ' ', 'company', ' ', 'release', ' ', 'said', '.', '&ldquo', ';',
'the', ' ', 'company&rsquo', ';', 's', ' ', 'revenue', ' ', 'grew', ' ',
'10', ' ', 'cent', ' ', 'y-o-y', ' ', 'rs', ' ', '14', ',', '169', ' ',
'crore', '.', '', ' ', 'consumer', ' ', 'shift', ' ', 'branded', ' ',
'staples', ' ', 'benefiting', ' ', 'significantly', '.', '', ' ',
'delivered', ' ', 'strong', ' ', 'quarter', ',', '', ' ', 'double-digit',
' ', 'growth', ' ', 'edible', ' ', 'oils', ' ', 'food', ' ', 'fmcg', ' ',
'segments', ',', '&rdquo', ';', '', ' ', 'said', ' ', 'angshu', ' ',
'mallick', ',', '', ' ', 'md', ' ', 'ceo', ',', '', ' ', 'adani', ' ',
'wilmar', '.', 'he', ' ', 'added', ',', '', ' ', '&ldquo', ';', 'with', '
', 'trusted', ' ', 'brand', ' ', 'fortune', ',', '', ' ', 'expect', ' ',
'continued', ' ', 'market', ' ', 'share', ' ', 'gains&hellip', ';', '', '
', 'food', ' ', 'products', ' ', 'making', ' ', 'significant', ' ',
'inroads', ' ', 'indian', ' ', 'households', ',', '', ' ', 'plan', ' ',
'meet', ' ', 'large', ' ', 'demand', ' ', 'enhancing', ' ', 'food', ' ',
'distribution', ' ', 'edible', ' ', 'oil', ' ', 'network', '.', '', ' ',
'years', ' ', 'launching', ' ', 'dedicated', ' ', 'horeca', ' ',
'distribution', ' ', 'channel', ',', '', ' ', 'surpassed', ' ', 'rs', '
', '500', ' ', 'crore', ' ', 'revenue', ' ', '12-month', ' ', 'basis', '
', 'achieved', ' ', '90', ' ', 'cent', ' ', 'y-o-y', ' ', 'volume', ' ',
'increase', ' ', 'q1', '.', '&rdquo', ';', '', '\n', 'securities', ' ',
'exchange', ' ', 'board', ' ', 'india', ' ', '', '(', 'sebi', ')', '', '
', 'friday', ' ', 'barred', ' ', 'fugitive', ' ', 'businessman', ' ',
'vijay', ' ', 'mallya', ' ', 'accessing', ' ', 'securities', ' ',
'markets', ' ', 'years', '.', 'the', ' ', 'liquor', ' ', 'baron', ' ',
'barred', ' ', 'associating', ' ', 'listed', ' ', 'company', ' ',
'years', '.', '', ' ', 'directed', ' ', 'freezing', ' ', 'securities', '
', 'holdings', ',', '', ' ', 'including', ' ', 'mutual', ' ', 'fund', '
', 'units', '.', 'mallya', ',', '', ' ', 'fii', ' ', 'entity']
```

```python
#Data cleaning - remove white space characters from a chunk
result = [item for item in result if item.strip()]
print(result[10:200])
```

```
    ['apartment', 'complexes', ',', 'villas', 'golf', 'courses',
'national', 'capital', 'region', ',', 'making', 'grand', 'entry',
'country', "'", 's', 'biggest', 'property', 'market', 'insolvency',
'court', '.', 'according', 'people', 'familiar', 'matter', ',', 'adani',
'spend', '$1', 'billion', 'real', 'estate', 'assets', 'jaypee', ',',
'country', "'", 's', 'biggest', 'bankruptcy', 'case', 'involving', '₹50',
',', '000', 'crore', 'bank', 'loans', '.', 'mumbai', 'bench', 'national',
'company', 'law', 'tribunal', '(', 'nclt', ')', 'admits', 'kishore',
'biyani-promoted', 'future', 'retail', 'ltd', 'liquidation', 'absence',
'viable', 'revival', 'plan', 'company', '.', 'tribunal', 'appointed',
'sanjay', 'gupta', 'company’s', 'liquidator', '.', 'division', 'bench',
'judicial', 'member', 'kuldip', 'kumar', 'kareer', 'technical', 'member',
'anil', 'raj', 'chellan', 'allowing', 'company’s', 'resolution',
'professional', 'vijaykumar', 'v', '.', 'iyer’s', 'application', 'admit',
'company', 'liquidation', ',', 'observed', 'maximum', 'period', 'cirp',
'expired', 'resolution', 'plan', 'approved', 'coc', '(', 'committee',
'creditors', ')', '.', 'company', 'admitted', 'liabilities', 'rs', '28',
',', '452', 'crore', '.', 'includes', 'secured', 'financial',
'creditors', "'", 'claims', 'rs', '14', ',', '422', 'crore', '.', '“we',
'considered', 'opinion', 'fit', 'case', 'liquidation', ',', '”', 'said',
'tribunal', '.', '“', '.', 'maximise', 'value', 'corporate', 'debtor',
',', 'liquidator', 'shall', 'endeavour', 'sale', 'corporate', 'debtor',
'going', 'concern', 'regulation', '32a', 'clause', '(', 'e', ')',
'insolvency', 'bankruptcy', 'board', 'india', '(', 'liquidation',
'process', ')', 'regulation', ',', '2016', ',', '”', 'said', 'tribunal',
'order', '.', 'year', 'november']
```

```python
#Create an array of tokens from a sample chunked input text
from typing import List
import re
def text_to_tokens(text: str) -> List[str]:
    """Create an array of tokens from a given input text data. White
spaces are
    Split takes care of special characters , which are treated as tokens
als
    Parameters:
    tokens (text: str ): A text string which needs to be tokenized
    Returns:
    List[str]: an list of tokens"""

    # split text into tokens
    result = re.split(r'([,.:;?_!~#"()\']|--|\s)', text)
    # remove white spaces
    result = [item for item in result if item.strip()]
    return result
```

```python
# # check the  function called -sample chunk output
tokenized = text_to_tokens(text)
 # check length of original text
print("The toekn count is -> ",len(text))
 # check length after tokenization and removal of whitespace characters
print("length after tokenization and removal of whitespace characters ->
",len(tokenized))
 # display 1st ten tokens
print(tokenized[10:200])
```

```
    The token count is ->  107
    length after tokenization and removal of whitespace characters ->  21
    ["'", 's', 'biggest', 'bankruptcy', 'case', 'involving', '₹50', ',',
'000', 'crore', '.']
```

```python
# Build News Media Vocabulary # "create_vocab"
from typing import List, Dict
def create_vocab(tokens: List[str], ) -> List [int]:
    """Creates a Dictionary which maps a token to its token ID. The token
input
    removed before a dictionary is mapped
    Parameters:
    tokens (tokens: List[str]): A list of tokens
    Returns:
    Dict[str, int]: a vocabulary dictionary which maps a token to a
unique t"""
    # remove duplicates
    unq_tokens = list(set(tokens))
    # sort
    srt_tokens = sorted(unq_tokens)
    # create vocabulary
    vocabulary = {token:tokenid for tokenid,token in
enumerate(srt_tokens)}
    return vocabulary
```

```python
# Create the Encoder
##Consider a variable-length sequence as input, and the leftwards context
of the target sequence and predicting
#the subsequent token in the target sequence.
import re
from typing import List, Dict

def encode(text: str, vocabulary: Dict[str, int]) -> List[int]:
    """
    Encode the input text into a list of token IDs using the given
vocabular
    Parameters:
    text (str): The input text string of tokens.
```

~~~
    vocabulary (Dict[str, int]): A dictionary mapping tokens to integer
valu
    Returns:
    List[int]: A list of integers representing the token IDs.
    """

    # Split the input text into tokens
    result = re.split(r'([,.:;?_!~#"()\']|--|\s)', text)

    # remove white spaces
    tokens = [item for item in result if item.strip()]


    # Generate the list of token IDs using the vocabulary
    token_ids = []
    for token in tokens:
        if token.strip() and token in vocabulary:
            token_ids.append(vocabulary[token])
        else:
            # Handle unknown tokens if necessary (e.g., append a special
tok
            # For example, let's append -1 for unknown tokens
            token_ids.append(-99)

    return token_ids

```
~~~

~~~
```python
# call function
vocab = create_vocab(tokenized)
 # Check - print 1st ten items of the vocabulary
for i, item in enumerate(vocab.items()):
    print(item)
    if i > 10:
        break
```
~~~

```
    ('!', 0)
    ('"', 1)
    ('#', 2)
    ('$1', 3)
    ('$10', 4)
    ('$12', 5)
    ('$15', 6)
    ('$150', 7)
    ('$2', 8)
    ('$20', 9)
    ('$225', 10)
    ('$243', 11)
```

~~~
```python
# Example usage
vocabulary = {
"adani":    1,
~~~

~~~~
"spend":    2,
"$1": 3,
"billion":  4,
"real":     5,
"estate":6,
"assets":7,
"jaypee":8,
"country's":9,
"biggest":10,
"bankruptcy":11,
"case":     12,
"involving":13,
"₹50,000":14,
"crore":15
}
text = "adani spend $1 billion real estate assets jaypee,country's
biggest bankruptcy case involving ₹50,000 crore."
encoded_text = encode(text, vocabulary)
print(encoded_text)

## If an unknown token is passed for encoding - return a -99 for the same
```

    [1, 2, 3, 4, 5, 6, 7, 8, -99, -99, -99, -99, 10, 11, 12, 13, -99, -
99, -99, 15, -99]



```python
# We develop the Decoder
from typing import List, Dict
def decode(vocabulary: Dict[str, int], token_ids: List[int]) ->
List[str]:
    """
    Decode the input list of token IDs into a list of string tokens using
th
    Parameters:
    vocabulary (Dict[str, int]): A dictionary mapping tokens to integer
valu
    token_ids (List[int]): A list of integers representing the token IDs.
    Returns:
    List[str]: A list of string tokens.
    """

    # Create a reverse dictionary from the vocabulary
    int_to_str = {v: k for k, v in vocabulary.items()}

    # Generate the list of string tokens using the reverse dictionary
    tokens = []
    for token_id in token_ids:
        if token_id in int_to_str:
            tokens.append(int_to_str[token_id])
        else:
            # Handle unknown token IDs if necessary (e.g., append a
special
            # For example, let's append -99 for unknown token IDs
            tokens.append(-99)
~~~~

~~~
    return tokens
```


```python
 # Example usage
 # define a test vocab
vocabulary = {
"adani":   1,
"spend":   2,
"$1": 3,
"billion":  4,
"real":     5,
"estate":6,
"assets":7,
"jaypee":8,
"country's":9,
"biggest":10,
"bankruptcy":11,
"case":    12,
"involving":13,
"₹50,000":14,
"crore":15
}
 token_ids = [1, 2, 3, 4, 5, 6, 7,8,9,10,11,12,13,14]
 decoded_tokens = decode(vocabulary, token_ids)
 print(decoded_tokens)
```

    ['adani', 'spend', '$1', 'billion', 'real', 'estate', 'assets',
'jaypee', "country's", 'biggest', 'bankruptcy', 'case', 'involving',
'₹50,000']



```python
# Build a final modified vocabulary function
from typing import List, Dict
def create_vocab(rawtext: List[str], ) -> List [int]:
    """
    Creates a Dictionary which maps a token to its token ID.
    Takes a list of raw strings
    tokenizes them and removes white spaces
    sorts the list
    adds a special token for unknown token
    adds a special token to mark end of text of a particular text source.
    generates token id list as output
    Parameters:
    rawtext (rawtext: List[str]): A list of raw text strings
    Returns:
    Dict[str, int]: a vocabulary dictionary which maps a token to a
unique t
    """

    # tokenize input text string
    tokens = re.split(r'([,.?_!~#"()\']|--|\s)', rawtext)

    # remove white space
~~~

~~~~
    tokens = [item.strip() for item in tokens if item.strip()]

    # remove duplicates
    unq_tokens = list(set(tokens))

    # sorted tokens
    srt_tokens = sorted(unq_tokens)

    # add special tokens for unknown strings and end of text segment
    srt_tokens.extend(["<|endoftext|>", "<|unk|>"])

    # create vocabulary
    vocabulary = {token:tokenid for tokenid,token in
enumerate(srt_tokens)}

    return vocabulary
```


```python
# check News media vocab text
print(text)
```

    adani spend $1 billion real estate assets jaypee,country's biggest
bankruptcy case involving ₹50,000 crore.



```python
 # Create Vocabulary from  text
vocab = create_vocab(text)
 # check length
lenvocab = len(vocab)
 print(lenvocab)
```

    22



```python
# # print the Negative media vocab dict
print(text)
```

    adani spend $1 billion real estate assets jaypee,country's biggest
bankruptcy case involving ₹50,000 crore.



```python
## Sample Adverse News Media keywords

words=["fraud","conspiracy", "cheating", "forgery", "false evidence",
"fake evidence", "breach of trust", "anti","money laundering",
      "corrupt", "abetment","scam", "helped in crime", "helped in
criminal", "extortion", "warrants pending", "warrant pending",
~~~~

```
       "sexual harassment", "fraudulent", "corruption","delayed payment",
       "loan due", "taxes due", "liquidation", "winding up","bankruptcy",
"bankrupt", "debarred", "blacklisted","labor issues",
       "labour issues", "forced labour", "drt", "debt recovery tribunal",
"debt recovery","forensic audit","financial misreporting",

'resigned','resignation','stake','acquired','offloaded','stakes','acquire
s','acquire','acquisition','appointed','appointments','insolvency','insol
vent',
      'show cause notice','loan default','loan
defaults','settlement','court proceedings','take over',
      'contempt notice','liquidator','winding up','penalises','levied
penalties','penalties','steps down','imposed penalty',
      'penalty','partnership','tax evasion','step
away','buyout','violated provisions','levied penalty',
      'demand notice','demand notices','show cause
notices','arrest','fined','refuses','ban','income tax','judicial
custody']
```

```python
from collections import Counter
def negative_media_word():
    i=0;
for  i in create_vocab(text):
    if i in words:
        print(f'The Negative media word /s is: {i}\n')
        print("The counts of negative media words are: \n",i.count(i))

```

```
    The Negative media word /s is: bankruptcy

    The counts of negative media words are: 
     1
```

```python
# Store teh negative media keywords in dictionary for future reference
import os
with open("negative_media_Vocab.txt", "w+") as output:
    output.write(i)
   # print(os.getcwd())
```

Note : Consolidated list of negative media dictionary available on demand for specific use case.