done 232,233,234,235,236,237,238,239,240,241

June 27, 2024

1 REMEMBER TO CHECK: HARUS 2023 TERBARU, dan lengkap 10 tahun ke belakang. Kalau engga, do not proceed that ticker/number to the 2nd snippet.

Future improvement:

Yang masuk sample dari 186 ke bawah, wajib di double-confirm dgn antconc karena no|none nya salah algoritma :(

nanti, setelah populasi muncul semua, bagi tiga kategori, outliers, first half, second half -> dimitris cuma bagi tiga rata aja

AAAAAAA SALAH ALGORITMA NYAAAAAAAAA, harusnya run through ulang dari nol

[1]:
```
import os
import pandas as pd
from sec_downloader import Downloader
from sec_downloader.types import RequestedFilings

def read_tickers_from_csv(csv_filename):
    df = pd.read_csv(csv_filename)
    return df.set_index('number')['ticker'].to_dict()

def validate_filing_dates(metadatas, start_year=2023, end_year=2014):
    if len(metadatas) != (start_year - end_year + 1):
        return False
    expected_years = {str(year) for year in range(end_year, start_year + 1)}
    actual_years = {metadata.report_date[:4] for metadata in metadatas}
    return expected_years == actual_years

# Initialize API connection
dl = Downloader("Afi Testing", "afiinvesting@gmail.com")
print("Connection successful\n")

# Load tickers from CSV
ticker_mapping = read_tickers_from_csv('numbered_from_smallest_modified.csv')
print("Tickers loaded from CSV:", ticker_mapping)
print("\n")
```

```
# Input numbers and get corresponding tickers
numbers = [int(num.strip()) for num in input("Enter the number(s) corresponding␣
 ↪to the ticker(s), separated by comma: ").split(",")]
tickers = [(num, ticker_mapping.get(num)) for num in numbers]

# Check for missing tickers
missing_tickers = [num for num, ticker in tickers if not ticker]
if missing_tickers:
    print(f"Ticker(s) not found for number(s): {', '.join(map(str,␣
 ↪missing_tickers))}\n")

# Process each ticker
tickers_not_covering_years = []
for num, ticker in tickers:
    if ticker:
        metadatas = dl.get_filing_metadatas(
            RequestedFilings(ticker_or_cik=ticker, form_type="10-K", limit=10)
        )

        if validate_filing_dates(metadatas):
            output_dir = f"downloaded_filings_{ticker}"
            os.makedirs(output_dir, exist_ok=True)

            for metadata in metadatas:
                html = dl.download_filing(url=metadata.primary_doc_url).decode()
                report_date = metadata.report_date
                file_name = f"{ticker}_{report_date}.txt"
                file_path = os.path.join(output_dir, file_name)

                with open(file_path, 'w', encoding='utf-8') as file:
                    file.write(html)

                print(f"Downloaded {file_name}")
            print()
        else:
            print(f"The filings for ticker {ticker} (number {num}) do not cover␣
 ↪the required range of years from 2023 to 2014.\n")
            tickers_not_covering_years.append((num, ticker))

if tickers_not_covering_years:
    print("The following tickers do not cover the required range of years from␣
 ↪2023 to 2014:")
    for num, ticker in tickers_not_covering_years:
        print(f"Ticker {ticker} (number {num})\n")

print("All filings downloaded and saved.\n")
```

```
Connection successful

Tickers loaded from CSV: {1: 'NIDB', 2: 'CRSB', 3: 'BKSC', 4: 'CARV', 5: 'VBFC',
6: 'QNTO', 7: 'PFBX', 8: 'NWPP', 9: 'IROQ', 10: 'FSRL', 11: 'AUBN', 12: 'SFBC',
13: 'UBOH', 14: 'UWHR', 15: 'FUSB', 16: 'CMTV', 17: 'HMNF', 18: 'PFLC', 19:
'CSBB', 20: 'UBFO', 21: 'FMBM', 22: 'PROV', 23: 'SBFG', 24: 'OVBC', 25: 'ASRV',
26: 'CIZN', 27: 'FKYS', 28: 'FXNC', 29: 'OPOF', 30: 'UNB', 31: 'BFIN', 32:
'FFNW', 33: 'RVSB', 34: 'SFDL', 35: 'LARK', 36: 'PLBC', 37: 'PEBK', 38: 'VABK',
39: 'NKSH', 40: 'EMYB', 41: 'QNBC', 42: 'MBCN', 43: 'EFSI', 44: 'FCCO', 45:
'FRAF', 46: 'TSBK', 47: 'OVLY', 48: 'CZWI', 49: 'FNRN', 50: 'HWBK', 51: 'FNCB',
52: 'FUNC', 53: 'BPRN', 54: 'PKBK', 55: 'ISBA', 56: 'EBMT', 57: 'EVBN', 58:
'ATLO', 59: 'CVLY', 60: 'NWFL', 61: 'PWOD', 62: 'WSBF', 63: 'TBNK', 64: 'FSFG',
65: 'LCNB', 66: 'ESSA', 67: 'ACNB', 68: 'CWBC', 69: 'CFFI', 70: 'FDBC', 71:
'CZNC', 72: 'BCML', 73: 'WNEB', 74: 'UNTY', 75: 'CHMG', 76: 'MCBC', 77: 'NRIM',
78: 'ISTR', 79: 'NBN', 80: 'FNLC', 81: 'FSBW', 82: 'CZFS', 83: 'CBAN', 84:
'ORRF', 85: 'FCBC', 86: 'FMAO', 87: 'MVBF', 88: 'HBCP', 89: 'FBIZ', 90: 'FGBI',
91: 'FRBA', 92: 'BSRR', 93: 'PFIS', 94: 'BMRC', 95: 'WTBA', 96: 'BCBP', 97:
'CIVB', 98: 'FRST', 99: 'AMBZ', 100: 'BHB', 101: 'SFST', 102: 'AROW', 103:
'FLIC', 104: 'CCBG', 105: 'HBIA', 106: 'SMBC', 107: 'EBTC', 108: 'HIFS', 109:
'HTBI', 110: 'SMBK', 111: 'THFF', 112: 'CNND', 113: 'EQBK', 114: 'FMNB', 115:
'INBK', 116: 'HTBK', 117: 'IBCP', 118: 'MPB', 119: 'FMCB', 120: 'TFIN', 121:
'MBWM', 122: 'CATC', 123: 'NFBK', 124: 'CAC', 125: 'OSBC', 126: 'FBAK', 127:
'CCNE', 128: 'CTBI', 129: 'GSBC', 130: 'SHBI', 131: 'FRBK', 132: 'GABC', 133:
'FISI', 134: 'CHCO', 135: 'TRST', 136: 'WABC', 137: 'MOFG', 138: 'PGC', 139:
'LKFN', 140: 'RBCAA', 141: 'PFBC', 142: 'HFWA', 143: 'WASH', 144: 'CASH', 145:
'HAFC', 146: 'FMBH', 147: 'CPF', 148: 'TBBK', 149: 'UVSP', 150: 'TMP', 151:
'HBNC', 152: 'FBMS', 153: 'SYBT', 154: 'SBSI', 155: 'NIC', 156: 'FFIC', 157:
'QCRH', 158: 'PFC', 159: 'SRCE', 160: 'PEBO', 161: 'HMST', 162: 'STBA', 163:
'PRK', 164: 'CNOB', 165: 'TCBK', 166: 'NBHC', 167: 'CFFN', 168: 'LOB', 169:
'BRKL', 170: 'FCF', 171: 'EGBN', 172: 'FBNC', 173: 'BUSE', 174: 'BANF', 175:
'VBTX', 176: 'BHLB', 177: 'FBK', 178: 'FFIN', 179: 'NBTB', 180: 'OCFC', 181:
'DCOM', 182: 'SASR', 183: 'PFS', 184: 'NWBI', 185: 'EFSC', 186: 'SBCF', 187:
'IBOC', 188: 'CBU', 189: 'BANR', 190: 'CVBF', 191: 'SFBS', 192: 'HTH', 193:
'TOWN', 194: 'RNST', 195: 'FFBC', 196: 'WSBC', 197: 'FRME', 198: 'TRMK', 199:
'PPBI', 200: 'IBTX', 201: 'HOPE', 202: 'INDB', 203: 'HTLF', 204: 'AX', 205:
'WSFS', 206: 'AUB', 207: 'CUBI', 208: 'WAFD', 209: 'HOMB', 210: 'CATY', 211:
'BOH', 212: 'ABCB', 213: 'UCBI', 214: 'SFNC', 215: 'FULT', 216: 'GBCI', 217:
'TCBI', 218: 'UBSI', 219: 'FIBK', 220: 'CBSH', 221: 'HWC', 222: 'BKU', 223:
'BANC', 224: 'PB', 225: 'ASB', 226: 'UMBF', 227: 'SSB', 228: 'FNB', 229: 'PNFP',
230: 'CADE', 231: 'ONB', 232: 'BOKF', 233: 'CFR', 234: 'COLB', 235: 'WTFC', 236:
'SNV', 237: 'VLY', 238: 'EWBC', 239: 'WAL', 240: 'WBS', 241: 'FHN', 242: 'CMA',
243: 'SBNY', 244: 'NYCB', 245: 'RF', 246: 'KEY', 247: 'HBAN', 248: 'MTB', 249:
'SIVBQ', 250: 'FRCB', 251: 'FCNCA', 252: 'FITB', 253: 'CFG', 254: 'TFC', 255:
'PNC', 256: 'USB', 257: 'WFC', 258: 'C', 259: 'BAC', 260: 'JPM'}


Enter the number(s) corresponding to the ticker(s), separated by comma:
```

```
232,233,234,235,236,237,238,239,240,241

Downloaded BOKF_2023-12-31.txt
Downloaded BOKF_2022-12-31.txt
Downloaded BOKF_2021-12-31.txt
Downloaded BOKF_2020-12-31.txt
Downloaded BOKF_2019-12-31.txt
Downloaded BOKF_2018-12-31.txt
Downloaded BOKF_2017-12-31.txt
Downloaded BOKF_2016-12-31.txt
Downloaded BOKF_2015-12-31.txt
Downloaded BOKF_2014-12-31.txt

Downloaded CFR_2023-12-31.txt
Downloaded CFR_2022-12-31.txt
Downloaded CFR_2021-12-31.txt
Downloaded CFR_2020-12-31.txt
Downloaded CFR_2019-12-31.txt
Downloaded CFR_2018-12-31.txt
Downloaded CFR_2017-12-31.txt
Downloaded CFR_2016-12-31.txt
Downloaded CFR_2015-12-31.txt
Downloaded CFR_2014-12-31.txt

Downloaded COLB_2023-12-31.txt
Downloaded COLB_2022-12-31.txt
Downloaded COLB_2021-12-31.txt
Downloaded COLB_2020-12-31.txt
Downloaded COLB_2019-12-31.txt
Downloaded COLB_2018-12-31.txt
Downloaded COLB_2017-12-31.txt
Downloaded COLB_2016-12-31.txt
Downloaded COLB_2015-12-31.txt
Downloaded COLB_2014-12-31.txt

Downloaded WTFC_2023-12-31.txt
Downloaded WTFC_2022-12-31.txt
Downloaded WTFC_2021-12-31.txt
Downloaded WTFC_2020-12-31.txt
Downloaded WTFC_2019-12-31.txt
Downloaded WTFC_2018-12-31.txt
Downloaded WTFC_2017-12-31.txt
Downloaded WTFC_2016-12-31.txt
Downloaded WTFC_2015-12-31.txt
Downloaded WTFC_2014-12-31.txt

Downloaded SNV_2023-12-31.txt
Downloaded SNV_2022-12-31.txt
```

```
Downloaded SNV_2021-12-31.txt
Downloaded SNV_2020-12-31.txt
Downloaded SNV_2019-12-31.txt
Downloaded SNV_2018-12-31.txt
Downloaded SNV_2017-12-31.txt
Downloaded SNV_2016-12-31.txt
Downloaded SNV_2015-12-31.txt
Downloaded SNV_2014-12-31.txt

Downloaded VLY_2023-12-31.txt
Downloaded VLY_2022-12-31.txt
Downloaded VLY_2021-12-31.txt
Downloaded VLY_2020-12-31.txt
Downloaded VLY_2019-12-31.txt
Downloaded VLY_2018-12-31.txt
Downloaded VLY_2017-12-31.txt
Downloaded VLY_2016-12-31.txt
Downloaded VLY_2015-12-31.txt
Downloaded VLY_2014-12-31.txt

Downloaded EWBC_2023-12-31.txt
Downloaded EWBC_2022-12-31.txt
Downloaded EWBC_2021-12-31.txt
Downloaded EWBC_2020-12-31.txt
Downloaded EWBC_2019-12-31.txt
Downloaded EWBC_2018-12-31.txt
Downloaded EWBC_2017-12-31.txt
Downloaded EWBC_2016-12-31.txt
Downloaded EWBC_2015-12-31.txt
Downloaded EWBC_2014-12-31.txt

Downloaded WAL_2023-12-31.txt
Downloaded WAL_2022-12-31.txt
Downloaded WAL_2021-12-31.txt
Downloaded WAL_2020-12-31.txt
Downloaded WAL_2019-12-31.txt
Downloaded WAL_2018-12-31.txt
Downloaded WAL_2017-12-31.txt
Downloaded WAL_2016-12-31.txt
Downloaded WAL_2015-12-31.txt
Downloaded WAL_2014-12-31.txt

Downloaded WBS_2023-12-31.txt
Downloaded WBS_2022-12-31.txt
Downloaded WBS_2021-12-31.txt
Downloaded WBS_2020-12-31.txt
Downloaded WBS_2019-12-31.txt
Downloaded WBS_2018-12-31.txt
```

```
Downloaded WBS_2017-12-31.txt
Downloaded WBS_2016-12-31.txt
Downloaded WBS_2015-12-31.txt
Downloaded WBS_2014-12-31.txt

Downloaded FHN_2023-12-31.txt
Downloaded FHN_2022-12-31.txt
Downloaded FHN_2021-12-31.txt
Downloaded FHN_2020-12-31.txt
Downloaded FHN_2019-12-31.txt
Downloaded FHN_2018-12-31.txt
Downloaded FHN_2017-12-31.txt
Downloaded FHN_2016-12-31.txt
Downloaded FHN_2015-12-31.txt
Downloaded FHN_2014-12-31.txt

All filings downloaded and saved.
```

2 JANGAN LUPA CEK 2023 HARUS TERBARU!! Plus lengkap 10 tahun yaaaa

[2]:
```python
import os
import re
import pandas as pd
from collections import Counter
from sec_downloader import Downloader
from sec_downloader.types import RequestedFilings

# Function to read tickers from CSV
def read_tickers_from_csv(csv_filename):
    df = pd.read_csv(csv_filename)
    return df.set_index('number')['ticker'].to_dict()

# Read tickers from the CSV file
ticker_mapping = read_tickers_from_csv('numbered_from_smallest_modified.csv') ␣
 ↪# Change this to your actual file path
print("Tickers loaded from CSV:", ticker_mapping)

# Input the number corresponding to the ticker
numbers = input("Enter the number(s) corresponding to the ticker(s), separated␣
 ↪by comma: ")
numbers = [int(num.strip()) for num in numbers.split(",")]

# Get the ticker corresponding to the input number
tickers = [ticker_mapping.get(num) for num in numbers]
```

```
# Check if the ticker exists
for ticker in tickers:
    if not ticker:
        print("Ticker not found.")
    else:
        # Define Keywords and Patterns
        keywords = [
            "Technology", "Digital Banking", "Network", "Internet Banking",
            "Online Services", "FinTech", "AI", "Blockchain", "E-payment",␣
 ↪"Mobile Banks"
        ]

        # Define Negative Patterns
        negative_patterns = re.compile(r'\b(no|none)\b', re.IGNORECASE)

        print("Step 1: Keywords and patterns defined.")

        def load_text(file_path):
            with open(file_path, 'r', encoding='utf-8') as file:
                return file.read()

        print("Step 2: Function to load text defined.")

        def find_keyword_context(text, keyword, context_length=20):
            pattern = re.compile(r'(.{0,%d}\b%s\b.{0,%d})' % (context_length,␣
 ↪re.escape(keyword), context_length), re.IGNORECASE)
            matches = pattern.findall(text)
            return matches

        print("Step 3: Function to capture keyword context defined.")

        # Function to count keywords and context
        def count_keywords_and_context(text, keywords, context_length=20):
            word_counts = Counter()
            keyword_contexts = {keyword: [] for keyword in keywords}

            for keyword in keywords:
                matches = find_keyword_context(text, keyword, context_length)
                word_counts[keyword] = len(matches)
                keyword_contexts[keyword] = matches

            return word_counts, keyword_contexts

        print("Step 4: Function to count keywords and capture context defined.")

        def clean_negative_occurrences(text):
```

```
            return re.sub(negative_patterns, '', text)

        print("Step 5: Function to clean negative occurrences defined.")

        # Processing files and capturing context
        def process_files(directory, keywords):
            results = []
            print("Step 6: Processing files and capturing context.")

            # Check if directory exists
            if not os.path.exists(directory):
                print(f"Error: The directory '{directory}' does not exist.")
                return results

            for filename in os.listdir(directory):
                if filename.endswith(".txt"):
                    file_path = os.path.join(directory, filename)
                    print(f"Processing file: {file_path}")
                    text = load_text(file_path)
                    text_cleaned = clean_negative_occurrences(text)
                    keyword_counts, keyword_contexts =␣
↪count_keywords_and_context(text_cleaned, keywords)

                    # Extract the company name and report date from the filename
                    base_filename = os.path.splitext(filename)[0]
                    current_ticker, report_date = base_filename.split('_')

                    if current_ticker == ticker:
                        # Calculate the fintech index as the sum of keyword␣
↪frequencies
                        fintech_index = sum(keyword_counts.values())

                        # Store the results
                        result = {
                            'Company': current_ticker,
                            'Report Date': report_date,
                            'Fintech Index': fintech_index,
                            'Keyword Contexts': keyword_contexts
                        }
                        result.update(keyword_counts)
                        results.append(result)
                        print("Processing completed.")

            return results

        print("Step 6: Function to process files and capturing context defined")
```

```
        # Function to display results with contexts
        def display_results(results, keywords, ticker):
            file_name = f"context_{ticker}.txt"
            with open(file_name, 'w') as file:
                file.write("Step 7: Displaying results with contexts.\n")
                for result in results:
                    file.write("Company: " + result['Company'] + "\n")
                    file.write("Report Date: " + result['Report Date'] + "\n")
                    file.write("Fintech Index: " + str(result['Fintech Index'])␣
↪+ "\n")  # Convert to string
                    for keyword in keywords:
                        file.write(f"{keyword}: {result[keyword]}\n")
                        for context in result['Keyword Contexts'][keyword]:
                                file.write(f"  Context: ...{context}...\n")
                    file.write("-" * 40 + "\n")  # Separator for each company's␣
↪results
                file.write("Results displayed.\n")
            print(f"Results saved to {file_name}")

        print("Step 7: Function to display results with contexts defined")

        # Function to save results to CSV
        def save_to_csv(results, ticker, output_filename):
            df = pd.DataFrame(results)
            df.to_csv(output_filename, index=False)
            print(f"Data successfully saved to {output_filename}")

        print("Step 8: Function to save results to CSV defined")
        print("\n")

        # Set the directory dynamically based on the ticker
        directory = f"downloaded_filings_{ticker}"

        # Process the text files
        results = process_files(directory, keywords)

        if results:
            # Display the results
            display_results(results, keywords, ticker)

            # Save the results to a CSV file with a dynamic name based on the␣
↪ticker symbol
            csv_filename = f"fintechindex_{ticker}.csv"
            save_to_csv(results, ticker, csv_filename)
```

```
Tickers loaded from CSV: {1: 'NIDB', 2: 'CRSB', 3: 'BKSC', 4: 'CARV', 5: 'VBFC',
6: 'QNTO', 7: 'PFBX', 8: 'NWPP', 9: 'IROQ', 10: 'FSRL', 11: 'AUBN', 12: 'SFBC',
```

```
13: 'UBOH', 14: 'UWHR', 15: 'FUSB', 16: 'CMTV', 17: 'HMNF', 18: 'PFLC', 19:
'CSBB', 20: 'UBFO', 21: 'FMBM', 22: 'PROV', 23: 'SBFG', 24: 'OVBC', 25: 'ASRV',
26: 'CIZN', 27: 'FKYS', 28: 'FXNC', 29: 'OPOF', 30: 'UNB', 31: 'BFIN', 32:
'FFNW', 33: 'RVSB', 34: 'SFDL', 35: 'LARK', 36: 'PLBC', 37: 'PEBK', 38: 'VABK',
39: 'NKSH', 40: 'EMYB', 41: 'QNBC', 42: 'MBCN', 43: 'EFSI', 44: 'FCCO', 45:
'FRAF', 46: 'TSBK', 47: 'OVLY', 48: 'CZWI', 49: 'FNRN', 50: 'HWBK', 51: 'FNCB',
52: 'FUNC', 53: 'BPRN', 54: 'PKBK', 55: 'ISBA', 56: 'EBMT', 57: 'EVBN', 58:
'ATLO', 59: 'CVLY', 60: 'NWFL', 61: 'PWOD', 62: 'WSBF', 63: 'TBNK', 64: 'FSFG',
65: 'LCNB', 66: 'ESSA', 67: 'ACNB', 68: 'CWBC', 69: 'CFFI', 70: 'FDBC', 71:
'CZNC', 72: 'BCML', 73: 'WNEB', 74: 'UNTY', 75: 'CHMG', 76: 'MCBC', 77: 'NRIM',
78: 'ISTR', 79: 'NBN', 80: 'FNLC', 81: 'FSBW', 82: 'CZFS', 83: 'CBAN', 84:
'ORRF', 85: 'FCBC', 86: 'FMAO', 87: 'MVBF', 88: 'HBCP', 89: 'FBIZ', 90: 'FGBI',
91: 'FRBA', 92: 'BSRR', 93: 'PFIS', 94: 'BMRC', 95: 'WTBA', 96: 'BCBP', 97:
'CIVB', 98: 'FRST', 99: 'AMBZ', 100: 'BHB', 101: 'SFST', 102: 'AROW', 103:
'FLIC', 104: 'CCBG', 105: 'HBIA', 106: 'SMBC', 107: 'EBTC', 108: 'HIFS', 109:
'HTBI', 110: 'SMBK', 111: 'THFF', 112: 'CNND', 113: 'EQBK', 114: 'FMNB', 115:
'INBK', 116: 'HTBK', 117: 'IBCP', 118: 'MPB', 119: 'FMCB', 120: 'TFIN', 121:
'MBWM', 122: 'CATC', 123: 'NFBK', 124: 'CAC', 125: 'OSBC', 126: 'FBAK', 127:
'CCNE', 128: 'CTBI', 129: 'GSBC', 130: 'SHBI', 131: 'FRBK', 132: 'GABC', 133:
'FISI', 134: 'CHCO', 135: 'TRST', 136: 'WABC', 137: 'MOFG', 138: 'PGC', 139:
'LKFN', 140: 'RBCAA', 141: 'PFBC', 142: 'HFWA', 143: 'WASH', 144: 'CASH', 145:
'HAFC', 146: 'FMBH', 147: 'CPF', 148: 'TBBK', 149: 'UVSP', 150: 'TMP', 151:
'HBNC', 152: 'FBMS', 153: 'SYBT', 154: 'SBSI', 155: 'NIC', 156: 'FFIC', 157:
'QCRH', 158: 'PFC', 159: 'SRCE', 160: 'PEBO', 161: 'HMST', 162: 'STBA', 163:
'PRK', 164: 'CNOB', 165: 'TCBK', 166: 'NBHC', 167: 'CFFN', 168: 'LOB', 169:
'BRKL', 170: 'FCF', 171: 'EGBN', 172: 'FBNC', 173: 'BUSE', 174: 'BANF', 175:
'VBTX', 176: 'BHLB', 177: 'FBK', 178: 'FFIN', 179: 'NBTB', 180: 'OCFC', 181:
'DCOM', 182: 'SASR', 183: 'PFS', 184: 'NWBI', 185: 'EFSC', 186: 'SBCF', 187:
'IBOC', 188: 'CBU', 189: 'BANR', 190: 'CVBF', 191: 'SFBS', 192: 'HTH', 193:
'TOWN', 194: 'RNST', 195: 'FFBC', 196: 'WSBC', 197: 'FRME', 198: 'TRMK', 199:
'PPBI', 200: 'IBTX', 201: 'HOPE', 202: 'INDB', 203: 'HTLF', 204: 'AX', 205:
'WSFS', 206: 'AUB', 207: 'CUBI', 208: 'WAFD', 209: 'HOMB', 210: 'CATY', 211:
'BOH', 212: 'ABCB', 213: 'UCBI', 214: 'SFNC', 215: 'FULT', 216: 'GBCI', 217:
'TCBI', 218: 'UBSI', 219: 'FIBK', 220: 'CBSH', 221: 'HWC', 222: 'BKU', 223:
'BANC', 224: 'PB', 225: 'ASB', 226: 'UMBF', 227: 'SSB', 228: 'FNB', 229: 'PNFP',
230: 'CADE', 231: 'ONB', 232: 'BOKF', 233: 'CFR', 234: 'COLB', 235: 'WTFC', 236:
'SNV', 237: 'VLY', 238: 'EWBC', 239: 'WAL', 240: 'WBS', 241: 'FHN', 242: 'CMA',
243: 'SBNY', 244: 'NYCB', 245: 'RF', 246: 'KEY', 247: 'HBAN', 248: 'MTB', 249:
'SIVBQ', 250: 'FRCB', 251: 'FCNCA', 252: 'FITB', 253: 'CFG', 254: 'TFC', 255:
'PNC', 256: 'USB', 257: 'WFC', 258: 'C', 259: 'BAC', 260: 'JPM'}

Enter the number(s) corresponding to the ticker(s), separated by comma:
232,233,234,235,236,237,238,239,240,241

Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
```

```
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_BOKF\BOKF_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_BOKF\BOKF_2023-12-31.txt
Processing completed.
Results saved to context_BOKF.txt
Data successfully saved to fintechindex_BOKF.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_CFR\CFR_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2018-12-31.txt
```

```
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_CFR\CFR_2023-12-31.txt
Processing completed.
Results saved to context_CFR.txt
Data successfully saved to fintechindex_CFR.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_COLB\COLB_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_COLB\COLB_2023-12-31.txt
Processing completed.
Results saved to context_COLB.txt
Data successfully saved to fintechindex_COLB.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
```

```
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_WTFC\WTFC_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_WTFC\WTFC_2023-12-31.txt
Processing completed.
Results saved to context_WTFC.txt
Data successfully saved to fintechindex_WTFC.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_SNV\SNV_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2016-12-31.txt
Processing completed.
```

```
Processing file: downloaded_filings_SNV\SNV_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_SNV\SNV_2023-12-31.txt
Processing completed.
Results saved to context_SNV.txt
Data successfully saved to fintechindex_SNV.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_VLY\VLY_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_VLY\VLY_2023-12-31.txt
Processing completed.
Results saved to context_VLY.txt
```

```
Data successfully saved to fintechindex_VLY.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_EWBC\EWBC_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_EWBC\EWBC_2023-12-31.txt
Processing completed.
Results saved to context_EWBC.txt
Data successfully saved to fintechindex_EWBC.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_WAL\WAL_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2015-12-31.txt
```

```
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_WAL\WAL_2023-12-31.txt
Processing completed.
Results saved to context_WAL.txt
Data successfully saved to fintechindex_WAL.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_WBS\WBS_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_WBS\WBS_2022-12-31.txt
Processing completed.
```

```
Processing file: downloaded_filings_WBS\WBS_2023-12-31.txt
Processing completed.
Results saved to context_WBS.txt
Data successfully saved to fintechindex_WBS.csv
Step 1: Keywords and patterns defined.
Step 2: Function to load text defined.
Step 3: Function to capture keyword context defined.
Step 4: Function to count keywords and capture context defined.
Step 5: Function to clean negative occurrences defined.
Step 6: Function to process files and capturing context defined
Step 7: Function to display results with contexts defined
Step 8: Function to save results to CSV defined


Step 6: Processing files and capturing context.
Processing file: downloaded_filings_FHN\FHN_2014-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2015-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2016-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2017-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2018-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2019-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2020-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2021-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2022-12-31.txt
Processing completed.
Processing file: downloaded_filings_FHN\FHN_2023-12-31.txt
Processing completed.
Results saved to context_FHN.txt
Data successfully saved to fintechindex_FHN.csv
```

[ ]:

[ ]: