

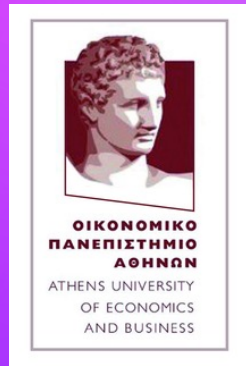
# Natural Language Processing for Business Documents

Lefteris Loukas

**Advisors:** Ion Androutsopoulos, George Paliouras, George Leledakis

**Mentors:** Prodromos Malakasiotis, Stavros Vassos





# Chapter #1

Research Question #1: “How can we use open-access documents for financial NLP?”

**Open-Source Software (OSS) and Resources**  
(EDGAR-CORPUS)


# Overview of Chapter #1

## EDGAR-CORPUS (ECONLP @ EMNLP 2021)

- Largest financial NLP corpus in the literature
- SOTA Word2Vec Embeddings (EDGAR-W2V)
- **Paper** at ECONLP Workshop @



## EDGAR-CRAWLER **edgar-crawler** Turn unstructured financial documents into clean JSON files.

- The go-to NLP toolkit for business/financial data/text preprocessing from the SEC
- 420+ stars on Github 
- Details:

- **Converts unstructured documents** of 100+ pages to a structured **JSON**
- Supports **multiple filters** like company, years, stock ticker
- Supports **multiple filings** (10-K, 10-Q, 8-K)
- Clean, normalize and **remove tables** with 2 clicks

- Lightning Talk at NLP-OSS @ 

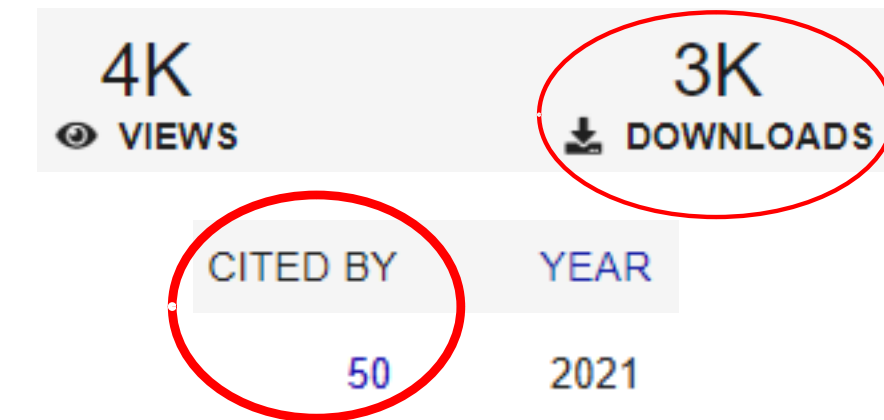
- Started in 2020, continuing until now

- Earned **grant** from  Google Summer of Code

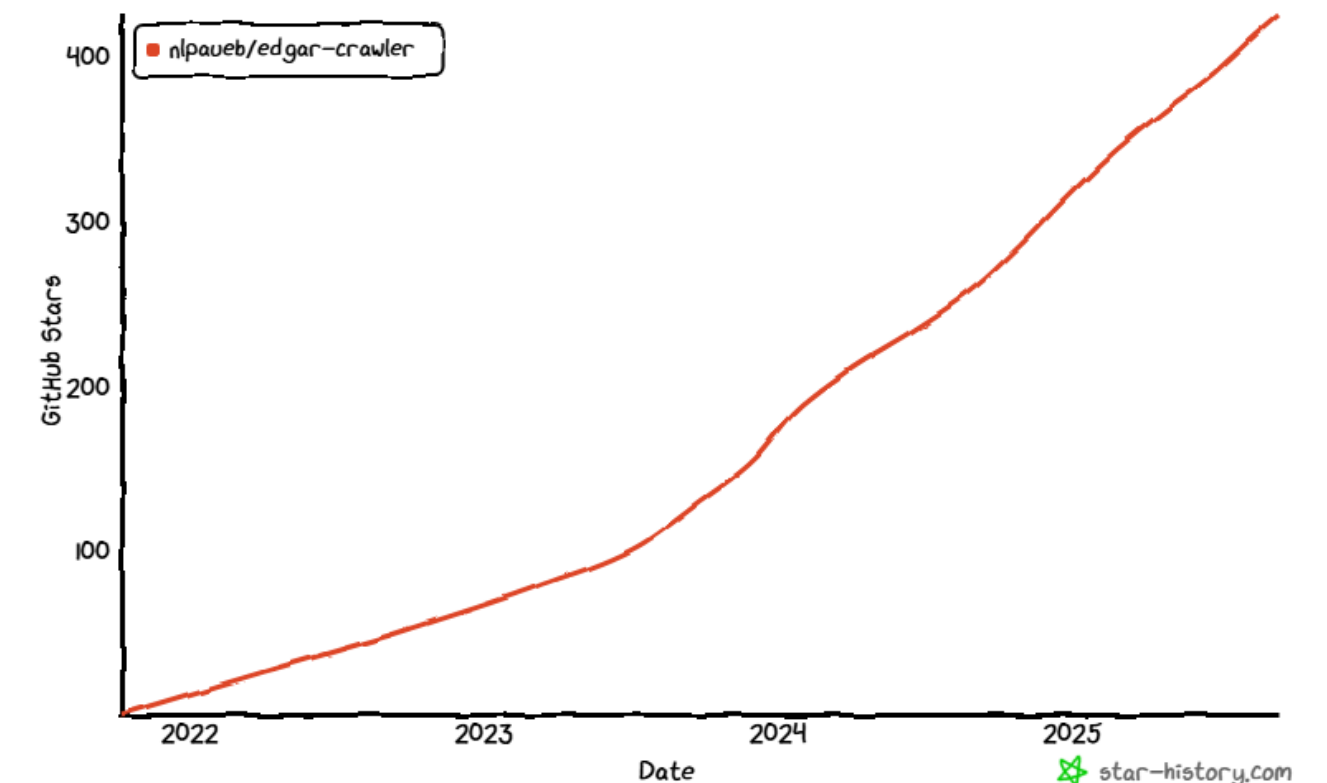
- **Paper at WWW 2025 (A\* CORE ranking)** 



Hugging Face



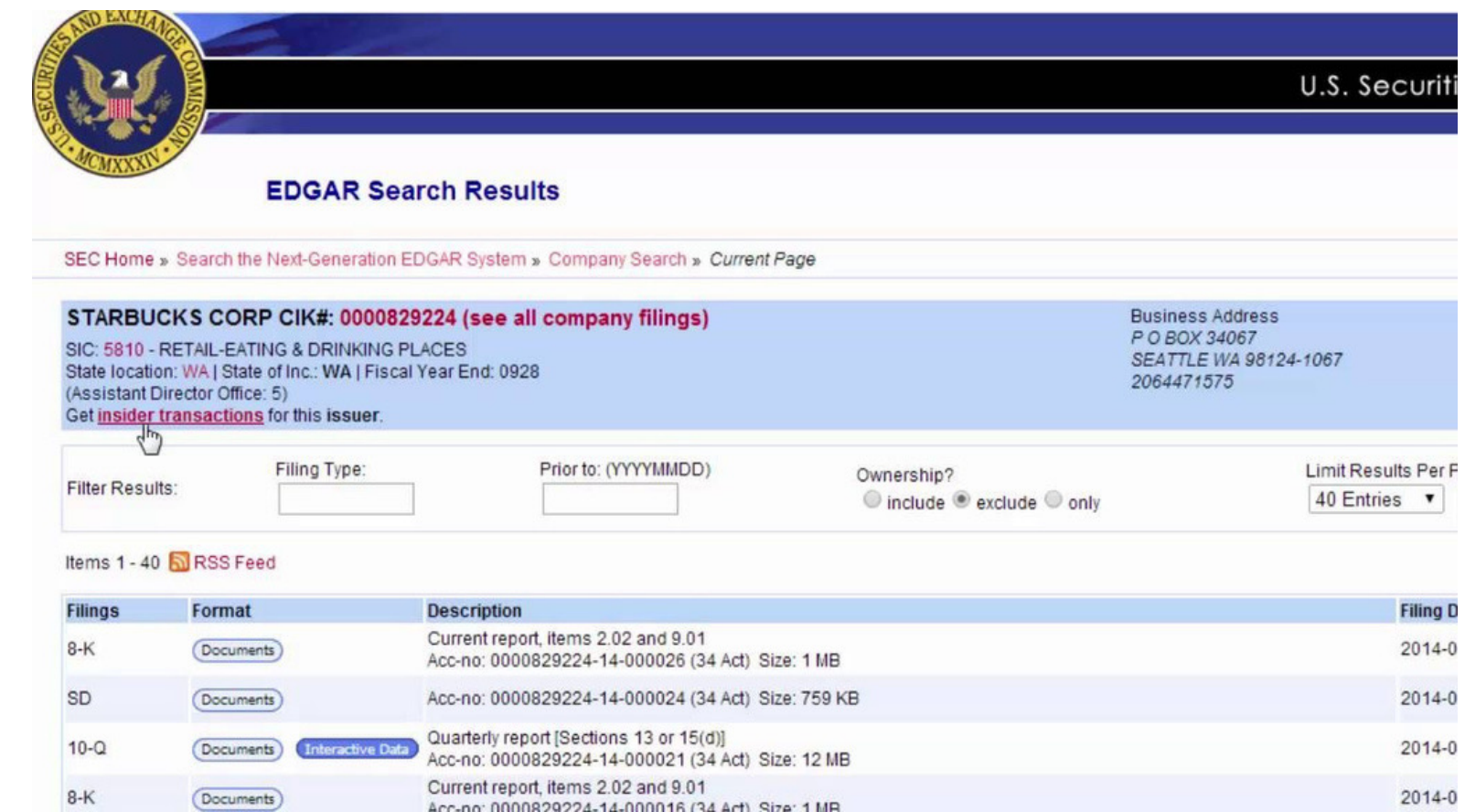
Star History





# Motivation

- Want to research financial NLP? You need data. 🤔
- **Problem: Domain is data limited..** Practitioners currently rely on heavily-paid data sources like paid APIs from SeekingAlpha/Bloomberg API, or they build their own web crawlers..
- **Idea:** Publicly-traded U.S. businesses store documents in **EDGAR**, a web repo from the **Securities & Exchange Commision (SEC)**
  - *EDGAR = Electronic Data Gathering, Analysis, and Retrieval*
  - So.. let's write some code for EDGAR and download batches of documents..
- We are not alone – lots of financial NLP applications use data from EDGAR:
  - *Stock price prediction (Lee et al., 2014)*
  - *Merger participants detection (Katsafados et al., 2021)*



The screenshot shows the SEC EDGAR Search Results page for Starbucks Corp. The page header includes the SEC logo and the text "U.S. Securities and Exchange Commission". The main heading is "EDGAR Search Results". Below this, there is a navigation bar with links: "SEC Home", "Search the Next-Generation EDGAR System", "Company Search", and "Current Page".

The search results for Starbucks Corp. (CIK#: 0000829224) are displayed. The company information includes: SIC: 5810 - RETAIL-EATING & DRINKING PLACES, State location: WA, State of Inc.: WA, Fiscal Year End: 0928, (Assistant Director Office: 5), and Business Address: P O BOX 34067, SEATTLE WA 98124-1067, 2064471575. A link to "insider transactions" is provided.

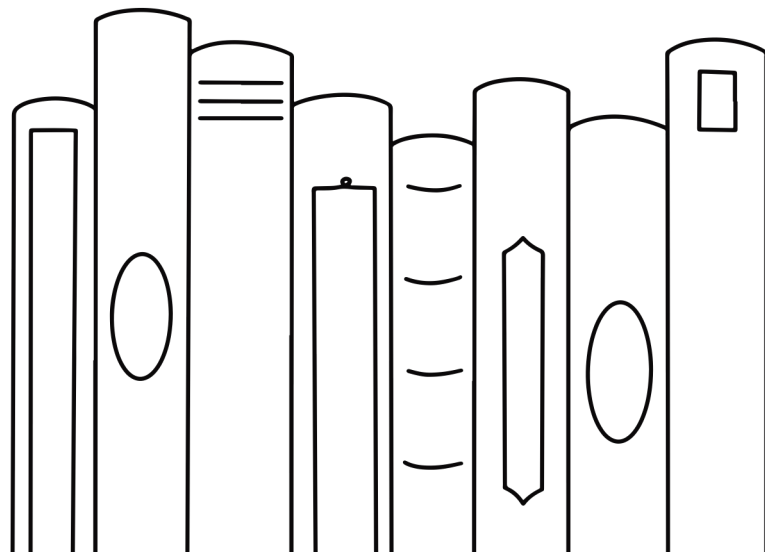
Below the company information, there are filters for "Filter Results:", "Filing Type:", "Prior to: (YYYYMMDD)", "Ownership?" (include, exclude, only), and "Limit Results Per Page" (40 Entries).

The search results table shows the following filings:

Filings	Format	Description	Filing Date
8-K	Documents	Current report, Items 2.02 and 9.01 Acc-no: 0000829224-14-000026 (34 Act) Size: 1 MB	2014-0
SD	Documents	Acc-no: 0000829224-14-000024 (34 Act) Size: 759 KB	2014-0
10-Q	Documents	Quarterly report [Sections 13 or 15(d)] Acc-no: 0000829224-14-000021 (34 Act) Size: 12 MB	2014-0
8-K	Documents	Current report, Items 2.02 and 9.01 Acc-no: 0000829224-14-000016 (34 Act) Size: 1 MB	2014-0

# Contribution

- Using our code, we gathered lots of documents and created **EDGAR-CORPUS**
- Novel resource for financial NLP 💡
- Contains annual reports (10-K filings) from all the publicly traded companies for a period of >25 years
- Largest financial NLP corpus available up to date!



# Related Work

- Few textual financial resources
- Certain limitations! 🛑
- Kogan et al. considered only 1 item (out of 20!) from the annual reports (10-K filings)
- Tsai et al. updated Kogan’s corpus to the year 2013

## EDGAR-CORPUS (ours)

- Contains **all 20 items** of the **annual reports from 1993 to 2020**
- Annual reports (10-K filings) describe the company’s activities; most notable text resource for business/economic NLP
- Total of **6.5B tokens inside!**

Corpora	Filings	Tokens	Companies	Years
<a href="#">Händschke et al. (2018)</a>	Various	242M	270	2000-2015
<a href="#">Daudert and Ahmadi (2019)</a>	Various	188M	60	1995-2018
<a href="#">Lee et al. (2014)</a>	8-K	27.9M	500	2002-2012
<a href="#">Kogan et al. (2009)</a>	10-K	247.7M	10,492	1996-2006
<a href="#">Tsai et al. (2016)</a>	10-K	359M	7,341	1996-2013
EDGAR-CORPUS (ours)	10-K	<b>6.5B</b>	<b>38,009</b>	<b>1993-2020</b>

Table 1: Financial corpora derived from SEC (lower part) and other sources (upper part).



# EDGAR documents have hundreds of pages

- An annual report (10-K report) is organized in 4 parts and **20 different items**, with **each item having specific points of interests**
- For each specific problem you want to solve (like stock prediction), you need to focus in a different item/subsection like Item 7 or Item 8
- Extracting specific items from documents with hundreds of pages requires extensive, manual work 🤔
- Inability to use those documents directly 🤔



	Item	Section Name
Part I	Item 1	Business
	Item 1A	Risk Factors
	Item 1B	Unresolved Staff Comments
	Item 2	Properties
	Item 3	Legal Proceedings
	Item 4	Mine Safety Disclosures
Part II	Item 5	Market
	Item 6	Consolidated Financial Data
	Item 7	Management's Discussion and Analysis
	Item 7A	Quantitative and Qualitative Disclosures about Market Risks
	Item 8	Financial Statements
	Item 9	Changes in and Disagreements With Accountants
Part III	Item 9A	Controls and Procedures
	Item 9B	Other Information
	Item 10	Directors, Executive Officers and Corporate Governance
	Item 11	Executive Compensation
	Item 12	Security Ownership of Certain Beneficial Owners
	Item 13	Certain Relationships and Related Transactions
Part IV	Item 14	Principal Accounting Fees and Services
	Item 15	Exhibits and Financial Statement Schedules Signatures

Table 2: The 20 different items of a 10-K report.

UNITED STATES  
SECURITIES AND EXCHANGE COMMISSION  
Washington, D.C. 20549

Form 10-K

(Mark One)

☒ ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

For the fiscal year ended September 28, 2013

Or

☐ TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

For the transition period from \_\_\_\_\_ to \_\_\_\_\_

Commission file number: 000-10030

**APPLE INC.**

(Exact name of registrant as specified in its charter)

California

(State or other jurisdiction of incorporation or organization)

1 Infinite Loop

Cupertino, California

(Address of principal executive offices)

94-2404110

(I.R.S. Employer Identification No.)

95014

(Zip Code)

Registrant's telephone number, including area code: (408) 996-1010

Securities registered pursuant to Section 12(b) of the Act:

Common Stock, no par value  
(Title of class)

The NASDAQ Stock Market LLC  
(Name of exchange on which registered)

Securities registered pursuant to Section 12(g) of the Act: None

AMAZON.COM, INC.  
CONSOLIDATED BALANCE SHEETS  
(in millions, except per share data)

	December 31,	
	2022	2023
<b>ASSETS</b>		
Current assets:		
Cash and cash equivalents	\$ 53,888	\$ 73,387
Marketable securities	16,138	13,393
Inventories	34,405	33,318
Accounts receivable, net and other	42,360	52,253
Total current assets	146,791	172,351
Property and equipment, net	186,715	204,177
Operating leases	66,123	72,513
Goodwill	20,288	22,789
Other assets	42,758	56,024
Total assets	\$ 462,675	\$ 527,854
<b>LIABILITIES AND STOCKHOLDERS' EQUITY</b>		
Current liabilities:		
Accounts payable	\$ 79,600	\$ 84,981
Accrued expenses and other	62,566	64,709
Unearned revenue	13,227	15,227
Total current liabilities	155,393	164,917
Long-term lease liabilities	72,968	77,297
Long-term debt	67,150	58,314
Other long-term liabilities	21,121	25,451
Commitments and contingencies (Note 7)		
Stockholders' equity:		
Preferred stock (\$0.01 par value; 500 shares authorized; no shares issued or outstanding)	—	—
Common stock (\$0.01 par value; 100,000 shares authorized; 10,757 and 10,898 shares issued; 10,242 and 10,383 shares outstanding)	108	109
Treasury stock, at cost	(7,837)	(7,837)
Additional paid-in capital	75,066	99,025
Accumulated other comprehensive income (loss)	(4,487)	(3,040)
Retained earnings	83,193	113,618
Total stockholders' equity	146,043	201,875
Total liabilities and stockholders' equity	\$ 462,675	\$ 527,854

Facebook, Inc.  
Form 10-K

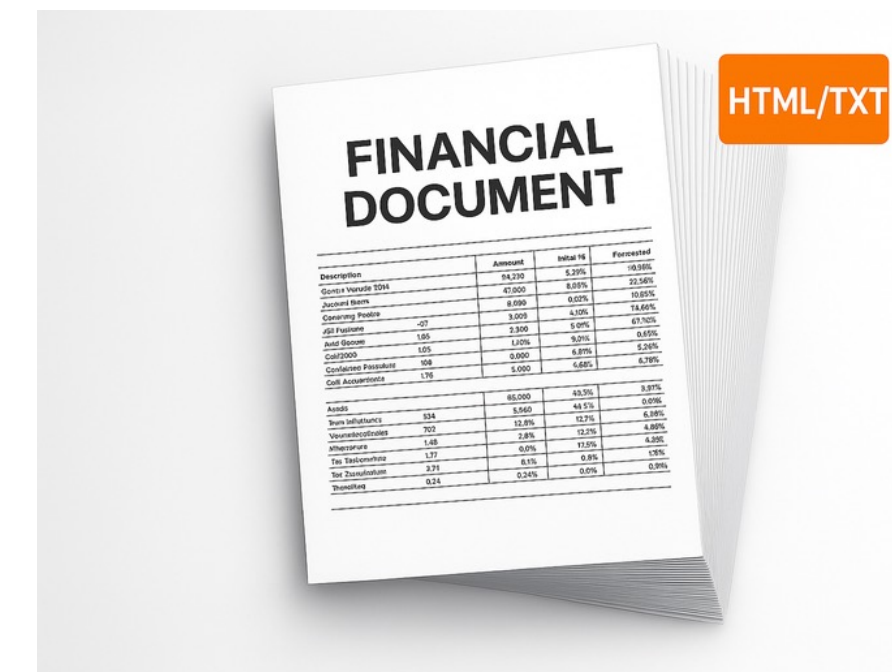
TABLE OF CONTENTS

	Page
<a href="#">Note About Forward-Looking Statements</a>	<a href="#">3</a>
<a href="#">Limitations of Key Metrics and Other Data</a>	<a href="#">4</a>
<b>PART I</b>	
<a href="#">Item 1. Business</a>	<a href="#">7</a>
<a href="#">Item 1A. Risk Factors</a>	<a href="#">12</a>
<a href="#">Item 1B. Unresolved Staff Comments</a>	<a href="#">45</a>
<a href="#">Item 2. Properties</a>	<a href="#">45</a>
<a href="#">Item 3. Legal Proceedings</a>	<a href="#">45</a>
<a href="#">Item 4. Mine Safety Disclosures</a>	<a href="#">47</a>
<b>PART II</b>	
<a href="#">Item 5. Market for Registrant's Common Equity, Related Stockholder Matters and Issuer Purchases of Equity Securities</a>	<a href="#">48</a>
<a href="#">Item 6. Selected Financial Data</a>	<a href="#">50</a>
<a href="#">Item 7. Management's Discussion and Analysis of Financial Condition and Results of Operations</a>	<a href="#">52</a>
<a href="#">Item 7A. Quantitative and Qualitative Disclosures About Market Risk</a>	<a href="#">76</a>
<a href="#">Item 8. Financial Statements and Supplementary Data</a>	<a href="#">77</a>
<a href="#">Item 9. Changes in and Disagreements with Accountants on Accounting and Financial Disclosure</a>	<a href="#">112</a>
<a href="#">Item 9A. Controls and Procedures</a>	<a href="#">112</a>
<a href="#">Item 9B. Other Information</a>	<a href="#">112</a>
<b>PART III</b>	
<a href="#">Item 10. Directors, Executive Officers and Corporate Governance</a>	<a href="#">113</a>
<a href="#">Item 11. Executive Compensation</a>	<a href="#">113</a>
<a href="#">Item 12. Security Ownership of Certain Beneficial Owners and Management and Related Stockholder Matters</a>	<a href="#">113</a>
<a href="#">Item 13. Certain Relationships and Related Transactions, and Director Independence</a>	<a href="#">113</a>
<a href="#">Item 14. Principal Accounting Fees and Services</a>	<a href="#">113</a>
<b>PART IV</b>	
<a href="#">Item 15. Exhibits, Financial Statement Schedules</a>	<a href="#">114</a>
<a href="#">Item 16. Form 10-K Summary</a>	<a href="#">116</a>
<a href="#">Signatures</a>	



# Let's structure (the unstructured)

- Used a variety of regular expressions – **trial & error** (many hours of tears and pains..)
- the item extraction algorithm in a high level:
  - for each document item, we scan for its item header and collect its text until the next item header, using regular expressions
  - Lots of false positives inside: table of contents, inline references..
    - lots of domain-specific rules to deal with those
- Finally, we have a way to convert unstructured documents with hundreds of pages to some structured JSON files with clean text data 😊
- Annual reports of every public US company + more than 20 years of data => we have **EDGAR-CORPUS!**



```
JSON
{
  filename : "881790_10K_2012.htm"
  cik : "881790"
  year : "2012"
  section_1 : "Item 1. Business ..."
  section_1A : "Item 1A. Risk Factors: The company identifies multiple risks ..."
  section_1B : "Item 1B. Unresolved Staff Comments: None yet."
  .. : ".."
  section_7 : "Item 7. Management's Discussion and Analysis: This section should ..."
  section_7A : "Item 7A. Quantitative and Qualitative Disclosures ..."
  section_8 : "Item 8. Financial Statements and Supplementary Data ..."
  ... : "..."
  section_15 : "Item 15. Exhibits and Financial Statement Schedules ..."
```

Figure 1: An example of a 10-K report in JSON format as downloaded and extracted by EDGAR-CRAWLER.

## **EDGAR-CORPUS characteristics**

- Contains all 20 items of the annual reports (**10-K filings**)
- Covers a time period from 1993 to 2020
- Each 10-K describes the company's activities comprehensively
- The documents provide a full outline of risks, liabilities and financial operations
- Total of 6.5B tokens inside!



# Word embeddings

- We used EDGAR-CORPUS to train and provide state-of-the-art Word2Vec **embeddings (EDGAR-W2V)**
- Helpful for downstream financial tasks
- Skip-gram algorithm
- 200-dimensional
- Vocabulary of 100k tokens

*EDGAR-W2V*

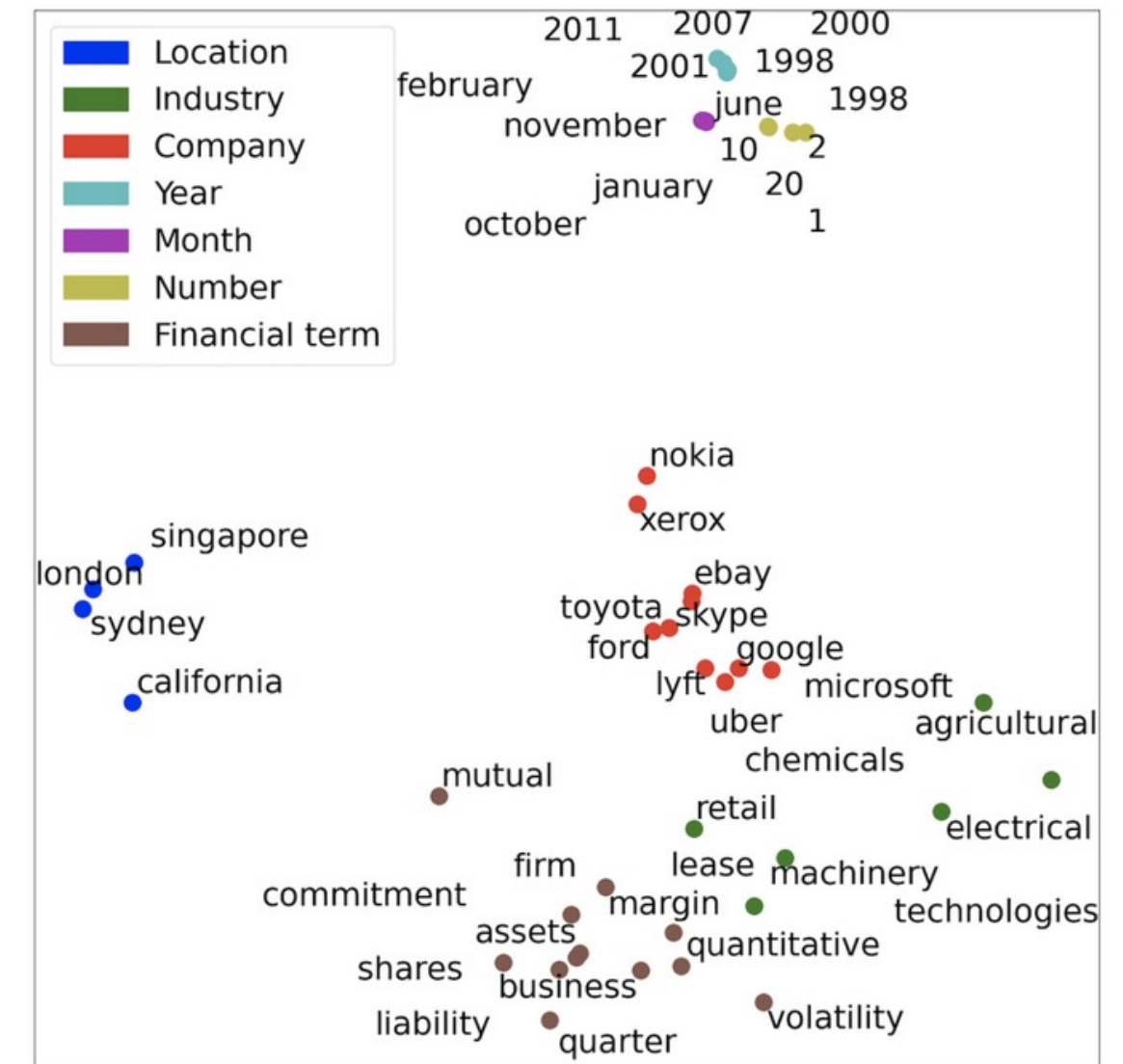


Figure 2: Visualization of the EDGAR-W2V embeddings. Different colors indicate different entity types.



# Experiments

- EDGAR-W2V outperforms generic GloVe embeddings and the financial embeddings of Tsai et al.
- 3 financial NLP tasks
  - **FinSim**: Business Hypernym Classification
  - **FiNER**: Sequence Labeling with word-level Annotations
  - **FiQA**: Financial Sentiment Analysis
- We also compare them with jina-embeddings-v4 and find that EDGAR-W2V outperforms them at 2 of 3 tasks

	FinSim-3		FiNER	FiQA	
	Acc. $\uparrow$	Rank $\downarrow$	F1 $\uparrow$	MSE $\downarrow$	$R^2$ $\uparrow$
GloVe	85.3	1.26	75.8	0.151	0.119
Tsai et al. (2016)	84.9	1.27	75.3	0.142	0.169
EDGAR-W2V (ours)	<b>87.9</b>	<b>1.21</b>	<b>77.3</b>	<b>0.141</b>	<b>0.176</b>
jina-embeddings-v4	83.9	1.32	72.3	<b>0.110</b>	<b>0.302</b>

Table 2.9: Results across financial NLP tasks, with different static word embeddings, as well as some LLM-derived embeddings. We report averages over 3 runs with different random seeds.

# Summary

- Introduced a novel NLP corpus for the financial domain, comprising textual data for all the US public companies, covering more than 25 years
- All the reports in the corpus are cleaned and split to an easy-to-use structured JSON format
- We trained new financial w2v embeddings, called EDGAR-W2V, which outperformed generic-domain and other financial embeddings in various financial NLP tasks
- Publication to 3rd ECONLP (EMNLP 2021)

1. EDGAR-CORPUS is available at: <https://huggingface.co/datasets/eloukas/edgar-corpus>
2. EDGAR-CRAWLER is available at: <https://github.com/nlpauieb/edgar-crawler>
3. The EDGAR-W2V embeddings are available at: <https://zenodo.org/record/5524358>

[EDGAR-CORPUS: Billions of Tokens Make The World Go Round](#). Lefteris Loukas, Manos Fergadiotis, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 13–18, Punta Cana, Dominican Republic. Association for Computational Linguistics.

aclanthology.org/2021.econlp-1.2/

ACL Anthology News FAQ Corrections Submissions Github Search...

## EDGAR-CORPUS: Billions of Tokens Make The World Go Round

Lefteris Loukas, Manos Fergadiotis, Ion Androutsopoulos, Prodromos Malakasiotis

**Abstract**

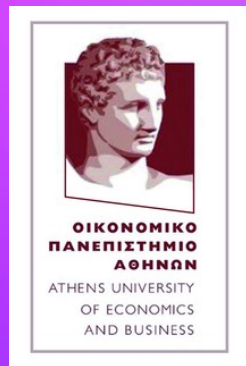
We release EDGAR-CORPUS, a novel corpus comprising annual reports from all the publicly traded companies in the US spanning a period of more than 25 years. To the best of our knowledge, EDGAR-CORPUS is the largest financial NLP corpus available to date. All the reports are downloaded, split into their corresponding items (sections), and provided in a clean, easy-to-use JSON format. We use EDGAR-CORPUS to train and release EDGAR-W2V, which are WORD2VEC embeddings for the financial domain. We employ these embeddings in a battery of financial NLP tasks and showcase their superiority over generic GloVe embeddings and other existing financial word embeddings. We also open-source EDGAR-CRAWLER, a toolkit that facilitates downloading and extracting future annual reports.

**Anthology ID:** 2021.econlp-1.2  
**Volume:** Proceedings of the Third Workshop on Economics and Natural Language Processing  
**Month:** November  
**Year:** 2021  
**Address:** Punta Cana, Dominican Republic  
**Editors:** Udo Hahn, Veronique Hoste, Amanda Stent  
**Venue:** ECONLP  
**SIG:** –  
**Publisher:** Association for Computational Linguistics  
**Note:** –  
**Pages:** 13–18  
**Language:** –  
**URL:** <https://aclanthology.org/2021.econlp-1.2>  
**DOI:** 10.18653/v1/2021.econlp-1.2

PDF Cite Search Video

Find all relevant  
links here!





# Chapter #1

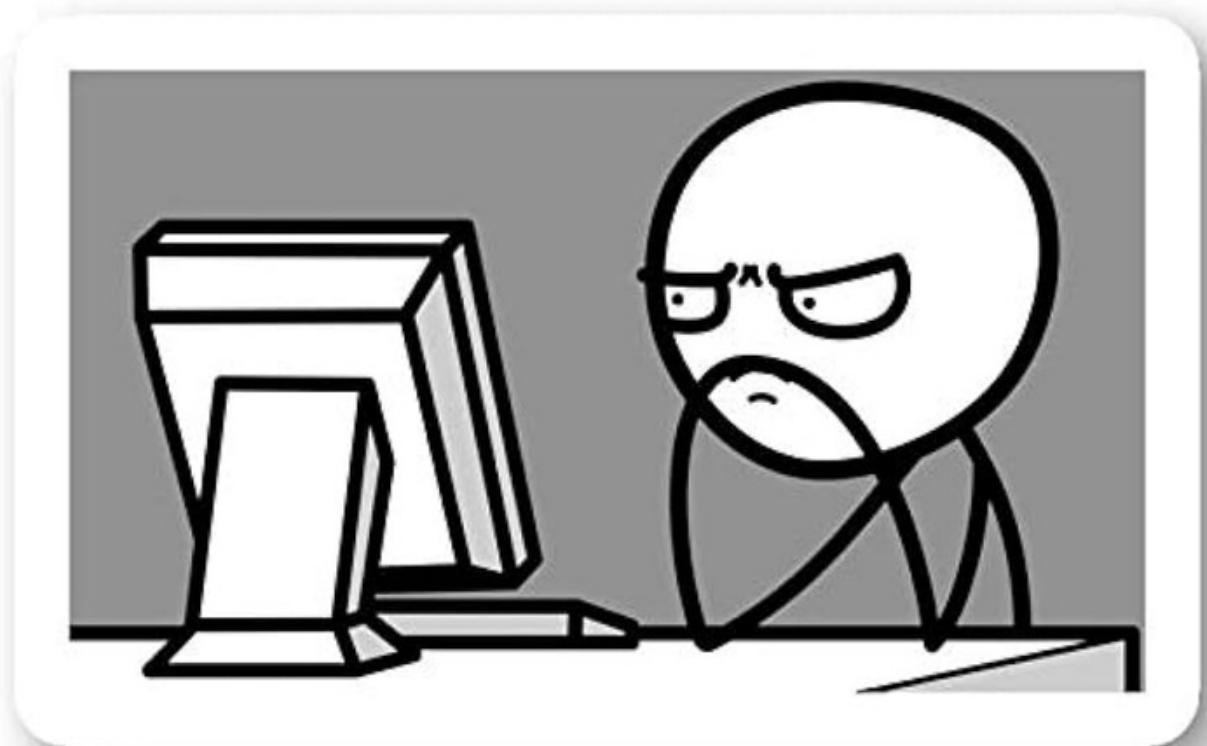
Research Question #1: “How can we use open-access documents for financial NLP?”

**Open-Source Software (OSS) and Resources**  
(EDGAR-CRAWLER)



## EDGAR-CORPUS was great, but..

- People were asking for the code of our crawler
- They **wanted** to preprocess **recent documents** (> 2020)
- They **wanted more filings** to be supported (not only 10-Ks)



🤔 Seems like there is a need for a stable and robust **open-source toolkit** to preprocess such documents -> **EDGAR-CRAWLER**

# EDGAR-CRAWLER: From Raw Web Documents to Structured Financial NLP Datasets

<https://github.com/nlpaueb/edgar-crawler>

## 💰 What's the **problem?** 🤔

Most **NLP datasets** are often behind APIs and paywalls. **EDGAR**, however, is a prominent free resource, offering financial filings from US publicly traded companies. Yet these reports come as **complex PDF, HTML, or TXT files**, filled with **multiple sections and pages**, making them challenging to work with. **Extracting specific data** often **means downloading countless reports and manually sifting through them**—an **impractical** and time-intensive task for researchers.

## 💰 Our **Solution:**

EDGAR-CRAWLER, a free, open-source toolkit that downloads and extracts information from SEC/EDGAR filings into an easy-to-manage JSON format. Unit-tested and fully documented.  
Supports 10-K, 10-Q, 8-K filings.

Our software, **EDGAR-CRAWLER**, is made up of two modules:

```
1. python download_filings.py
2. python extract_items.py
```

1. Responsible for crawling and downloading financial reports.  
Supports multiple input arguments.

2. Cleans and extracts the text of all or particular items from downloaded filings and saves them as JSON files.


## 💰 Scientific Contributions in **ML & NLP:**


- Trusted by the community (**420+ stars** on Github!)
- **Multiple citations** in relevant literature.



## 💰 **Future Work:**

*Looking for contributors for these, send us a message if interested. 😊*

  
Support more types of documents like those for insider trading.

  
Create a GUI for more user-friendly configuration.




Turn unstructured financial documents into clean JSON files.

**edgar-**  
**crawler**



Hundreds of pages from unstructured company filings?

Structured JSON to bootstrap your research!

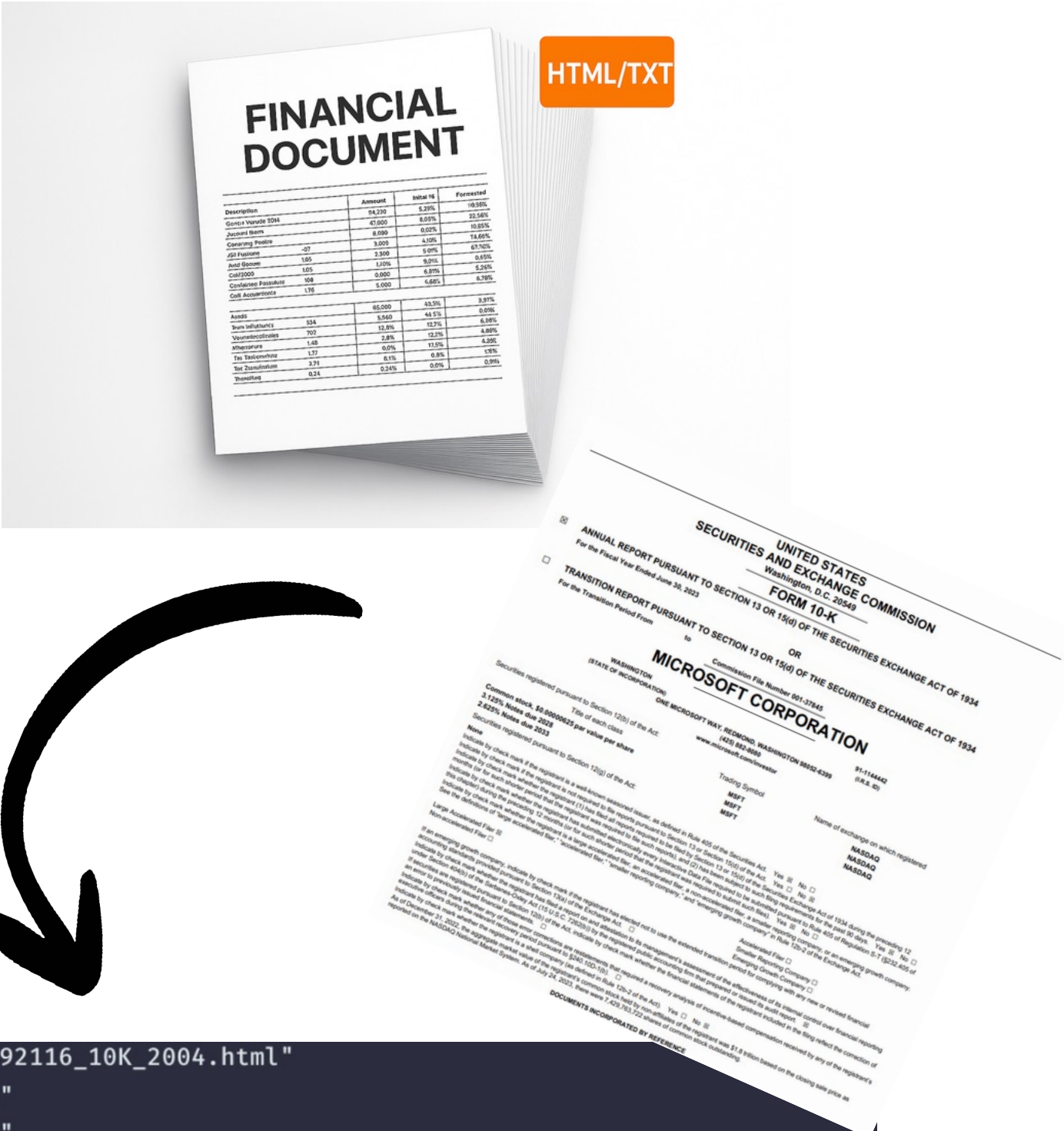


edgar-crawler

Turn unstructured financial documents into clean JSON files.

Fork110

Star426



```
filename: "92116_10K_2004.html"
cik: "92116"
year: "2004"
section_1: "Item 1. Business ..."
section_1A: "Item 1A. Risk Factors: ..."
section_1B: "Item 1B. Unresolved Staff Comments: None."
...: "..."
section_7: "Item 7. Management's Discussion and Analysis: ..."
section_7A: "Item 7A. Quantitative and Qualitative Disclosures: ..."
...: "..."
section_15: "Item 15. Exhibits and Financial Statement Schedules: ..."
```







## Let's structure (the unstructured)

- Used a variety of domain-specific regular expressions
- lots of **trial & error** (and many hours of tears..) to find what it works
- the item extraction algorithm in a high level:
  - for each document item, we scan for its item header and collect its text until the next item header, using regular expressions
  - .. while we make sure we filter out false positives like table of contents, inline references, missed sections, etc. (hardest part)
- Continuous development if something arises
  - Coding agents like Github Copilot are used to update for new document structures (rare, but might happen once in a while)
    - “issues” can be delegated to Agent
    - human-in-the-loop only for the review of the regular expressions

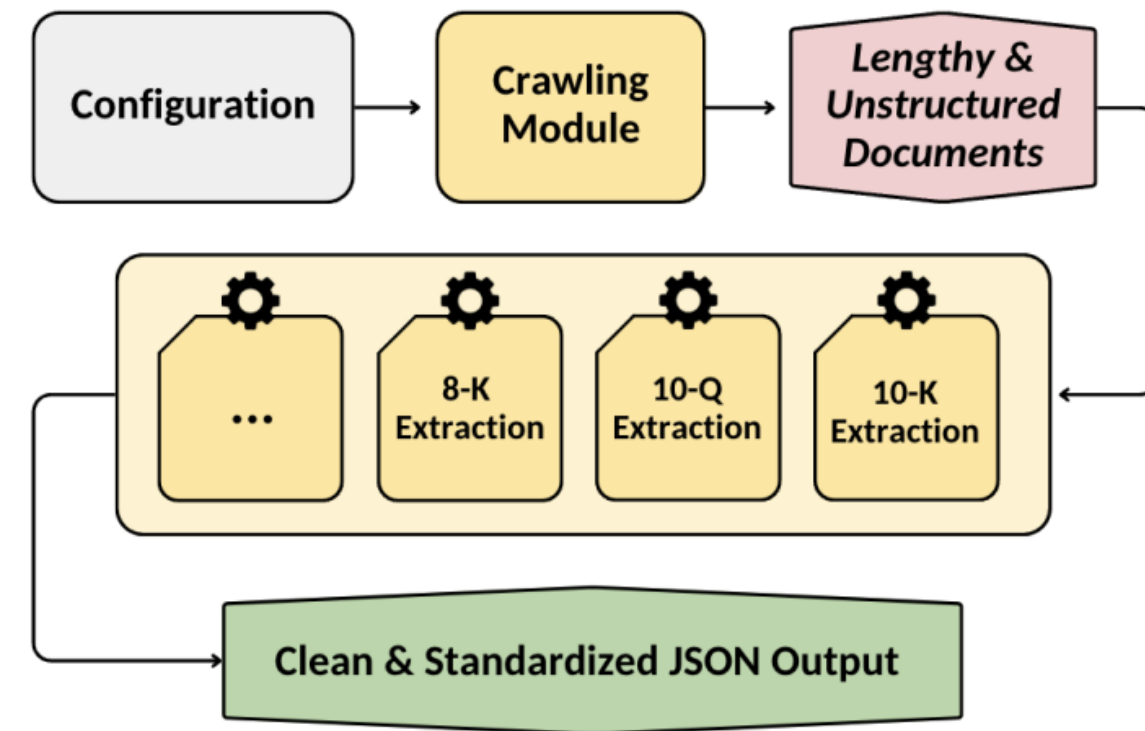


Figure 2.4: EDGAR-CRAWLER's architecture. First, the user specifies (Configuration) what data they want (companies, years, filing types) and the software downloads them. Then, the item extraction pipeline extracts the item-specific sections and converts them to a standardized JSON format.

```
JSON
{
  filename : "881790_10K_2012.htm"
  cik : "881790"
  year : "2012"
  section_1 : "Item 1. Business ..."
  section_1A : "Item 1A. Risk Factors: The company identifies multiple risks ..."
  section_1B : "Item 1B. Unresolved Staff Comments: None yet."
  .. : ".."
  section_7 : "Item 7. Management's Discussion and Analysis: This section should ..."
  section_7A : "Item 7A. Quantitative and Qualitative Disclosures ..."
  section_8 : "Item 8. Financial Statements and Supplementary Data ..."
  ... : "..."
  section_15 : "Item 15. Exhibits and Financial Statement Schedules ..."
```

Figure 1: An example of a 10-K report in JSON format as downloaded and extracted by EDGAR-CRAWLER.

## Nop, LLMs can not process SEC documents easily

- Hundreds of pages inside financial documents → SEC documents don't even fit inside the context window of most LLMs
- What about chunk-based processing for LLMs? still, it would be really expensive to do that
- **EDGAR-CRAWLER** can produce structured JSON output faster & easier



<https://github.com/nlpauieb/edgar-crawler>

lefterisloukas / **edgar-crawler** Public

[Code](#) [Issues](#) 3 [Pull requests 1 \[Actions\]\(#\) \[Projects\]\(#\) \[Security\]\(#\) \[Insights\]\(#\)](#)

main 2 Branches 0 Tags

Go to file

Code

About

lefterisloukas	Update README.md	84a8d0c · 2 months ago	95 Commits
datasets	- Update companies_info metadata	3 years ago	
images	Update documentation	2 years ago	
logs	- New JSON metadata	4 years ago	
tests	Allow item pattern to match 'Items'. When multiple items ar...	8 months ago	
.gitignore	Add Visual Studio Code configuration to .gitignore	9 months ago	
LICENSE	Initial commit	4 years ago	
README.md	Update README.md	2 months ago	
__init__.py	Create folders for datasets and logs on startup	4 years ago	
config.json	Rename edgar_crawler to download_filings and update confi...	9 months ago	
download_filings.py	Rename edgar_crawler to download_filings and update confi...	9 months ago	
extract_items.py	Allow item pattern to match 'Items'. When multiple items ar...	8 months ago	
item_lists.py	solve merge conflicts for 10-Q branch.	11 months ago	
logger.py	Rename edgar_crawler to download_filings and update confi...	9 months ago	
requirements.txt	Add extract items test	2 years ago	

README GPL-3.0 license

## EDGAR-CRAWLER: Extract Key Financial Data from SEC Filings Effortlessly 🚀

The only open-source toolkit that can download SEC EDGAR financial reports and extract textual data from specific item sections into nice & clean structured JSON files. Presented at WWW 2025 @ Sydney, Australia (<https://dl.acm.org/doi/10.1145/3701716.3715289>)

python nlp finance natural-language-processing business data-mining web-crawler sec edgar edgar-crawler

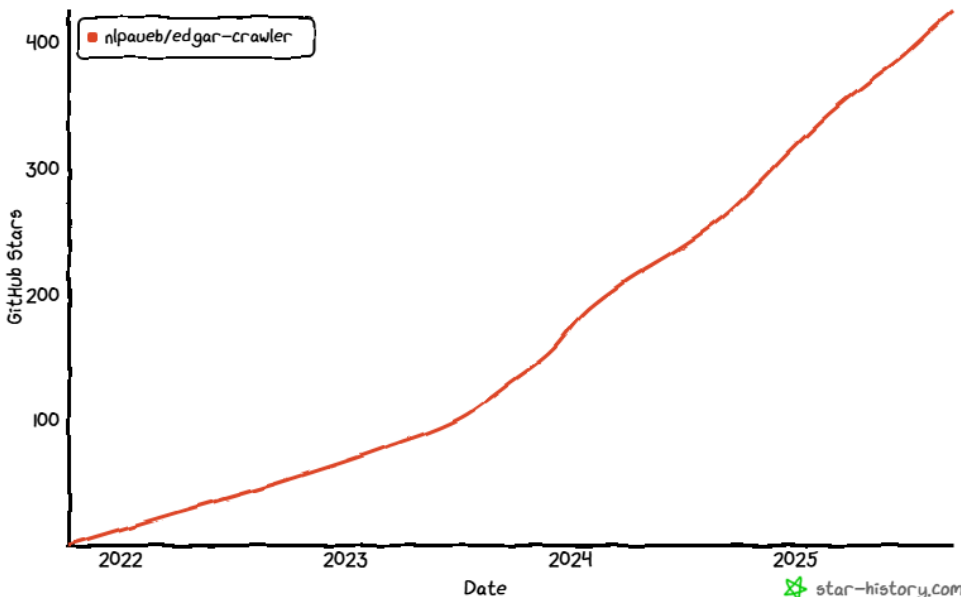
Readme  
GPL-3.0 license  
Activity  
426 stars  
21 watching  
110 forks  
Report repository

Releases  
No releases published

Contributors 4

lefterisloukas Lefteris Loukas  
Bailefan Fabian Billert  
manosfer Manos Fergadiotis

Star History



## EDGAR-CRAWLER: Extract Key Financial Data from SEC Filings Effortlessly 🚀



EDGAR-CRAWLER simplifies access to financial text data by downloading SEC EDGAR filings and transforming these complex, unstructured documents into structured, standardized JSON files, making it easier to use them for downstream NLP tasks and financial analysis.

EDGAR-CRAWLER has 2 core functionalities:

- Seamless downloading:** Retrieve and download financial filings from all US publicly-traded companies based on your specified filters, like year, quarters, filing type, etc.
- Structured output:** Extract and parse key sections from 10-K, 10-Q, and 8-K filings into a nice-and-easy standardized JSON format. (filings supported: 10-K, 10-Q, 8-K)

### News

- 2024-10-14: We added support for JSON parsing of 10-Q filings. (@Bailefan)
- 2024-10-05: We added support for JSON parsing of 8-K filings. (@Bailefan)
- 2023-12-06: We had a Lightning Talk about EDGAR-CRAWLER at the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS), hosted at EMNLP 2023, in Singapore.
- 2023-01-16: EDGAR-CORPUS, the biggest financial NLP corpus (generated from EDGAR-CRAWLER), is available as a HuggingFace dataset card. See Accompanying Resources for more details.
- 2022-10-13: Updated documentation and fixed a minor import bug.
- 2022-04-03: EDGAR-CRAWLER is available for Windows systems too.
- 2021-11-11: We presented EDGAR-CORPUS, our sister work that started it all, at ECONLP 2021 (EMNLP Workshop) at the Dominican Republic. See Accompanying Resources for more details.

### Table of Contents

- [Example Outputs](#)
- [Install](#)
- [Usage](#)
- [Citation](#)
- [Accompanying Resources](#)
- [Contributing](#)
- [License](#)

eloukas Lefteris Loukas  
Bailefan Fabian Billert  
manosfer Manos Fergadiotis  
dependabot[bot]

### Languages

Python 100.0%

### Suggested workflows

Based on your tech stack

**Python Package using Anaconda** [Configure](#)

Create and test a Python package on multiple Python versions using Anaconda for package management.

**SLSA Generic generator** [Configure](#)

Generate SLSA3 provenance for your existing release workflows

**Python application** [Configure](#)

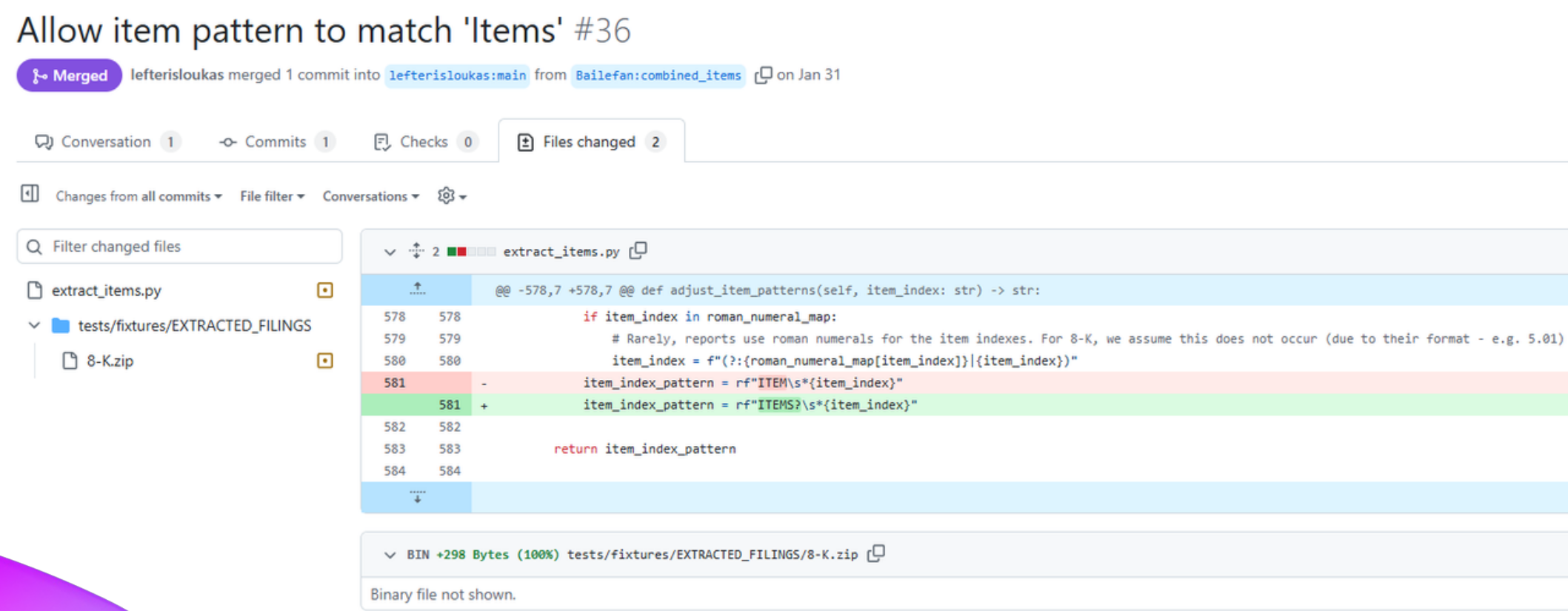
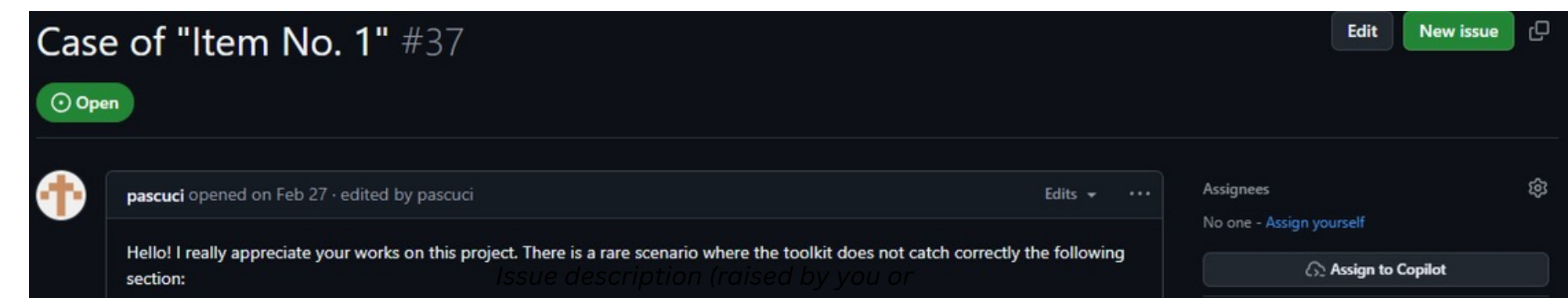
Create and test a Python application.

[More workflows](#)

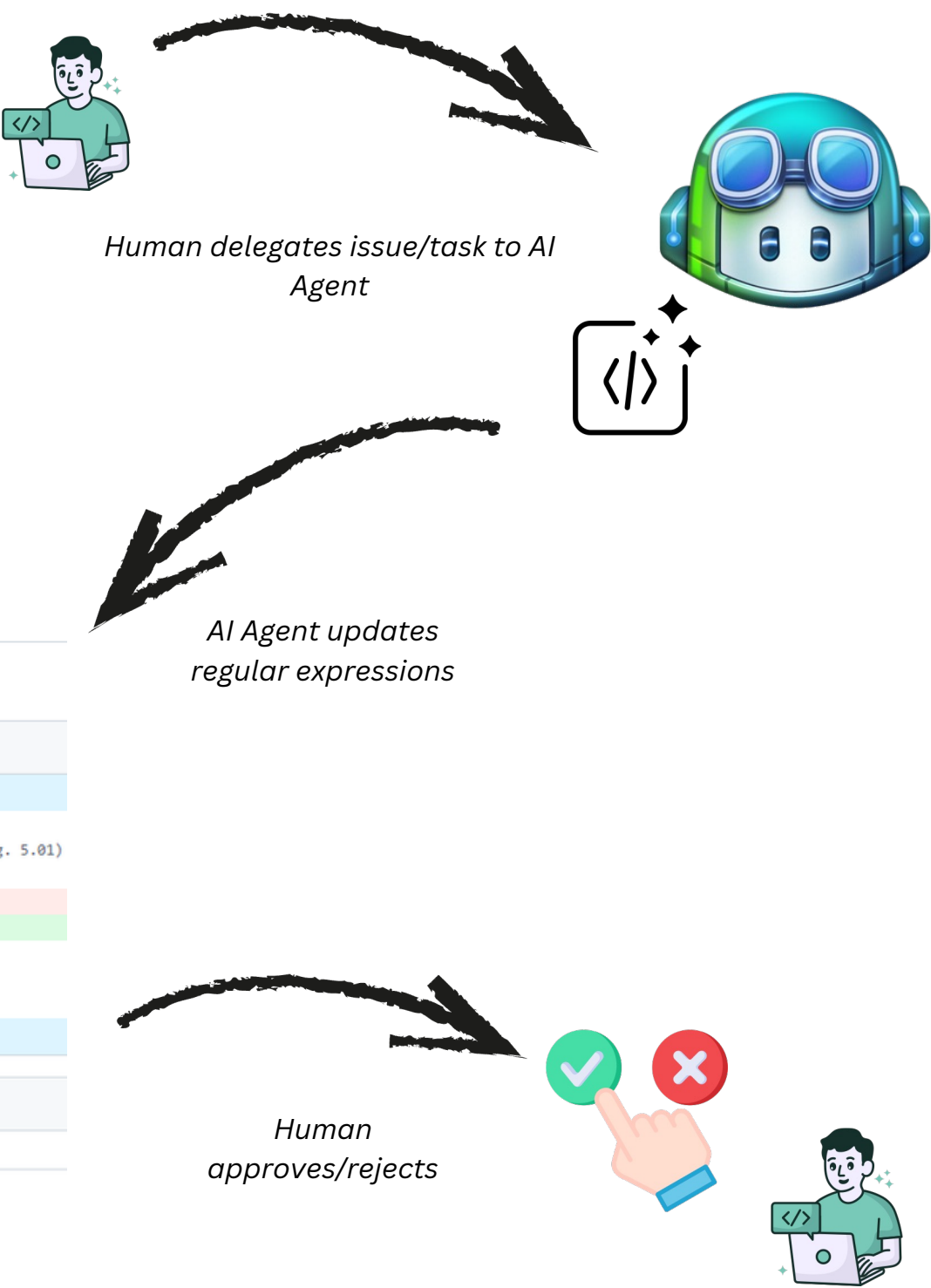
[Dismiss suggestions](#)



# Updating the software's rules is easy when you use AI (Coding) Agents



Code changes implemented by Copilot / Agent



EDGAR-CRAWLER is widely used by researchers to utilize open-access business documents for financial NLP

Google Scholar

"EDGAR-CRAWLER"

Articles11 results (0.06 sec)

Any time

Since 2024

Since 2023

Since 2020

Custom range...

Sort by relevance

Sort by date

Any type

Review articles

☐ include patents

☐ include citations

☒ Create alert

From Numbers to Words: Multi-Modal Bankruptcy Prediction Using the ECL Dataset

[H Arno](#), [K Mulier](#), [J Baeck](#), [T Demeester](#) - arXiv preprint arXiv:2401.12652, 2024 - arxiv.org

... Using the **EDGAR-crawler** tool,<sup>3</sup> we have collected the textual data (and corresponding ...  
github.com/nlpau**edgar-crawler** <sup>4</sup>This is the starting point of the EDGAR-corpus as well. <sup>5</sup>Our ...

☆ Save ↀ Cite Cited by 2 Related articles All 8 versions ↀ

Financial misstatement detection: a realistic evaluation

[E Zavitsanos](#), [D Mavroeidis](#), [K Bougiatiotis](#)... - Proceedings of the ..., 2021 - dl.acm.org

... We downloaded the 10-K annual filings from EDGAR using the **edgarcrawler** in [31]. We  
extracted the text from the MD&A section and performed basic segmentation and cleaning to ...

☆ Save ↀ Cite Cited by 7 Related articles All 5 versions

Characterizing Multimodal Long-form Summarization: A Case Study on Financial Reports

[T Cao](#), [N Raman](#), [D Dervovic](#), [C Tan](#) - arXiv preprint arXiv:2404.06162, 2024 - arxiv.org

As large language models (LLMs) expand the power of natural language processing to handle  
long inputs, rigorous and systematic analyses are necessary to understand their abilities ...

☆ Save ↀ Cite Cited by 1 Related articles All 2 versions ↀ

Tracking Real Time Layoffs with SEC Filings: A Preliminary Investigation

[LD Crane](#), [E Green](#), [M Harnish](#), [W McClennan](#), [PE Soto](#)... - 2024 - papers.ssrn.com

We explore a new source of data on layoffs: timely 8-K filings with the Securities and and  
Exchange Commission. We develop measures of both the number of reported layoff events and ...

☆ Save ↀ Cite Cited by 1 Related articles All 6 versions ↀ

Identifying going concern issues in auditor opinions: link to bankruptcy events

[K Bougiatiotis](#), [E Zavitsanos](#)... - 2023 IEEE International ..., 2023 - ieeexplore.ieee.org

... We downloaded the 10-K annual filings from EDGAR using the **edgar-crawler** in [32] and  
extracted the corresponding sections that provide the auditors' opinions. We filter the reports by ...

☆ Save ↀ Cite Cited by 1 Related articles All 2 versions

Hidden neighbours: extracting industry momentum from stock networks

[JCJ Ahn](#), [D Gorduza](#), [S Park](#) - Financial Markets and Portfolio ..., 2024 - Springer

... The historical 10-X disclosures are gathered from the **Edgar Crawler** developed in Loukas  
et al. (2021) and The Notre Dame Software Repository for Accounting and Finance. Footnote ...

☆ Save ↀ Cite Related articles

[PDF] Predicting companies' ESG rating from their 10-K filings using a text mining approach

[B Roufousse](#) - 2024 - matheo.uliege.be

... crawler, named **edgar-crawler** (details about the crawler can be found in the related Github...  
1https://github.com/nlpau**edgar-crawler** ...

☆ Save ↀ Cite Related articles ↀ





# Overview of Chapter #1

## EDGAR-CORPUS (ECONLP @ EMNLP 2021)

- **Largest** financial NLP corpus in the literature
- SOTA Word2Vec **Embeddings** (EDGAR-W2V)
- **Paper** at ECONLP Workshop @

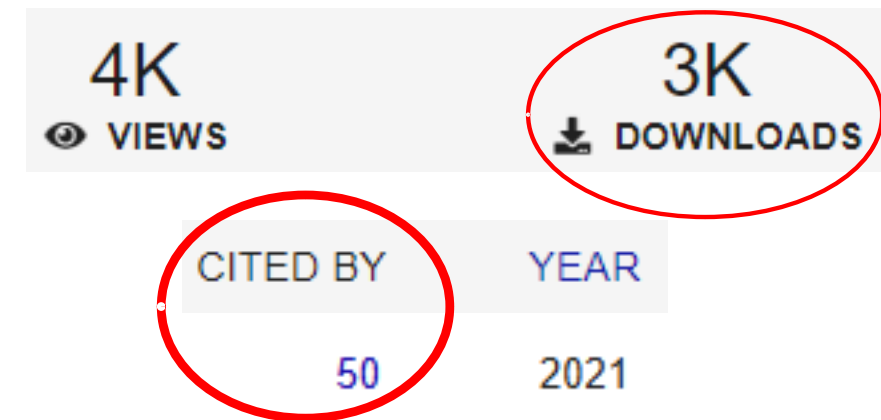


## EDGAR-CRAWLER **edgar-crawler** Turn unstructured financial documents into clean JSON files.

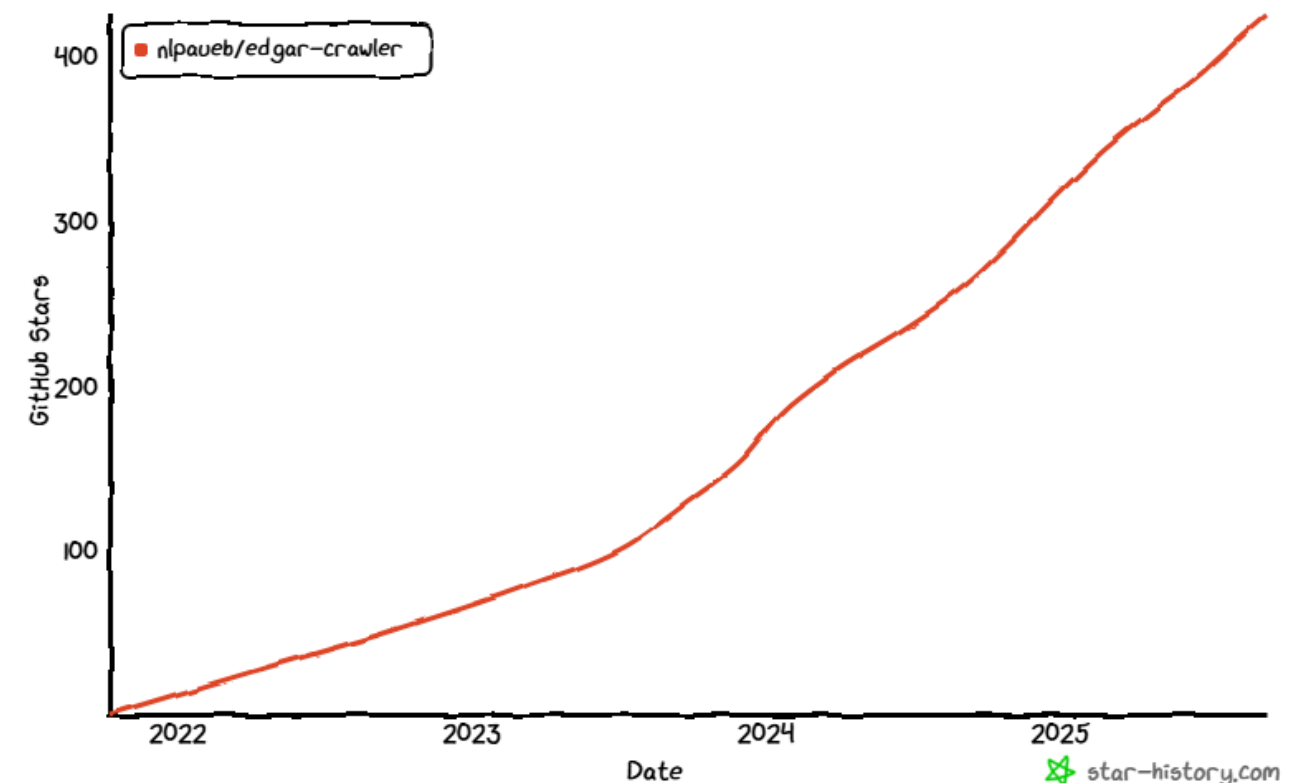
- The go-to NLP toolkit for business/financial data/text preprocessing from the SEC
- 420+ stars on Github 
- Details:
  - Turns unstructured documents into structured data to be used in financial NLP
  - **Converts documents** of 100+ pages to an easy-to-digest **JSON**
  - Supports **multiple filters** like company, years, stock ticker
  - Supports **multiple filings** (10-K, 10-Q, 8-K)
- Lightning Talk at NLP-OSS @ 
- Started in 2020, continuing until now
- Earned **grant** from  Google Summer of Code
- **Paper at WWW 2025 (A\* CORE ranking)** 



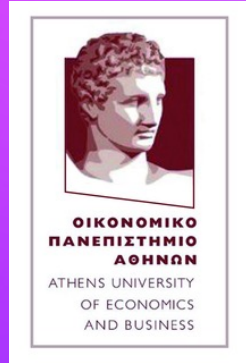
Hugging Face



Star History







# Chapter #2

Research Question #2: “How can NLP/DL methods create business value in automatic document tagging, and how can current methods be improved?”

**Numeric Entity Recognition for XBRL Tagging**

FiNER: Financial Numeric Entity Recognition for XBRL Tagging

- Published at **ACL 2022** (A\* CORE ranking conference)
- Problem:** new task + token overfragmentation of Transformer models in numbers
- Solution:** new tokenization method for pre-training and fine-tuning so transformers learn better number representations
- Created new BERT models & dataset (<https://huggingface.co/nlpaueb/sec-bert-base> & <https://huggingface.co/datasets/nlpaueb/finer-139>)
  - downloaded around 7,000 times
- Solves a real-life business problem task of XBRL Tagging
  - US SEC requires publicly traded companies to tag their documents with XBRL tags



1 granted US patent based on this methodology/task

- 1 granted US patent
- Patent is also submitted to the EU / World Patent Office
- Assigned to **Ernst and Young (EY) & NCSR Demokritos** for commercial use



CITED BY	YEAR
98	2022

Debt Carrying Value is net of \$ 5.2 million and \$ 6.4 million of deferred financing fees at March 31, 2019, and December 2018, respectively.

Deferred Finance Costs Net

Debt Carrying Value is net of \$ [NUM] million and \$ [NUM] million of deferred financing fees at March [NUM], [NUM], and December [NUM], respectively.

Deferred Finance Costs Net

Deferred Finance Costs Net

Debt Carrying Value is net of \$ [X.X] million and \$ [X.X] million of deferred financing fees at March [XX], [XXXX], and December [XXXX], respectively.

(12) <b>United States Patent</b>		(10) <b>Patent No.: US 12,333,236 B2</b>	
Loukas et al.		(45) <b>Date of Patent: Jun. 17, 2025</b>	
(54) <b>SYSTEM AND METHOD FOR AUTOMATICALLY TAGGING DOCUMENTS</b>		(58) <b>Field of Classification Search</b> None See application file for complete search history.	
(71) Applicant: <b>National Centre for Scientific Research "Demokritos"</b> , Agia Paraskevi (GR)		(56) <b>References Cited</b>  U.S. PATENT DOCUMENTS 10,817,619 B1 * 10/2020 Kolli ..... G06F 21/552 10,997,369 B1 * 5/2021 Frazier ..... G06F 40/284 (Continued)  FOREIGN PATENT DOCUMENTS CN 112257442 1/2021 EP 4124988 2/2023 WO WO 2023/006773 2/2023	
(72) Inventors: <b>Eleftherios Panagiotis Loukas</b> , Agia Paraskevi (GR); <b>Eirini Spyropoulou</b> , Agia Paraskevi (GR); <b>Prodromos Malakasiotis</b> , Agia Paraskevi (GR); <b>Emmanouil Fergadiotis</b> , Agia Paraskevi (GR); <b>Ilias Chalkidis</b> , Agia Paraskevi (GR); <b>Ioannis Androutsopoulos</b> , Agia Paraskevi (GR); <b>Georgios Paliouras</b> , Agia Paraskevi (GR)			



# Motivation / Problem

- **Publicly-traded companies** in the U.S. are required to file periodic financial reports to EDGAR
- **Filings** must be **tagged with XBRL (Extensive Business Reporting Language)** to indicate financial entities
  - XBRL helps with document analytics and processing
- **XBRL Tagging is costly, manual** & intensive for businesses -> **need for automation**
- XBRL Tagging also became a requirement for E.U. filings in 2020

U.S. Securities and Exchange Commission

## EDGAR Search Results

SEC Home » Search the Next-Generation EDGAR System » Company Search » Current Page

**STARBUCKS CORP CIK#: 0000829224 (see all company filings)**

SIC: 5810 - RETAIL-EATING & DRINKING PLACES  
State location: WA | State of Inc.: WA | Fiscal Year End: 0928  
(Assistant Director Office: 5)  
Get [insider transactions](#) for this issuer.

Business Address  
P O BOX 34067  
SEATTLE WA 98124-1067  
2064471575

Filter Results: Filing Type: Prior to: (YYYYMMDD) Ownership? ☐ include ☒ exclude ☐ only Limit Results Per Page: 40 Entries

Items 1 - 40 [RSS Feed](#)

Filings	Format	Description	Filing Date
8-K	<a href="#">Documents</a>	Current report, Items 2.02 and 9.01 Acc-no: 0000829224-14-000026 (34 Act) Size: 1 MB	2014-0
SD	<a href="#">Documents</a>	Acc-no: 0000829224-14-000024 (34 Act) Size: 759 KB	2014-0
10-Q	<a href="#">Documents</a> <a href="#">Interactive Data</a>	Quarterly report [Sections 13 or 15(d)] Acc-no: 0000829224-14-000021 (34 Act) Size: 12 MB	2014-0
8-K	<a href="#">Documents</a>	Current report, Items 2.02 and 9.01 Acc-no: 0000829224-14-000016 (34 Act) Size: 1 MB	2014-0

Note 5 - [Income Taxes](#)

**U.S. Tax Cuts and Jobs Act**

On December 22, 2017, the U.S. enacted the Tax Cuts and Jobs Act (the "Act"), which significantly changed U.S. tax law. The Act lowered the Company's U.S. statutory federal income tax rate from 35% to 21% while also imposing a deemed repatriation tax on previously deferred foreign income. The Act also created a new minimum tax on certain foreign earnings, for which the Company is liable. The Company completed its accounting for the income tax effects of the Act during 2019, in accordance with the U.S. Securities and Exchange Commission Staff Accounting Bulletin 118.

**Provision for Income Taxes and Effective Tax Rate**

The provision for income taxes for 2019, 2018 and 2017, consisted of the following (in millions):

	2019	2018	2017
Federal:			
Current	\$ 6,384	\$ 5,384	\$ 5,384
Deferred	(2,939)	(2,939)	(2,939)
Total	3,445	2,445	2,445
State:			
Current	475	475	475
Deferred	(67)	(67)	(67)
Total	408	408	408
Foreign:			
Current	3,962	3,962	3,962
Deferred	2,666	2,666	2,666
Total	6,628	6,628	6,628
Provision for income taxes	\$ 10,481	\$ 9,481	\$ 9,481

The foreign provision for income taxes is based on foreign pre-tax earnings of \$44.3 billion, \$48.0 billion and \$44.7 billion in 2019, 2018 and 2017, respectively.



# The Problem (from an NLP POV)

- We focus on text (since tables are mostly static)
- Viewed as a **sequence labeling task**
- Given a document, recognize the XBRL tags in its text

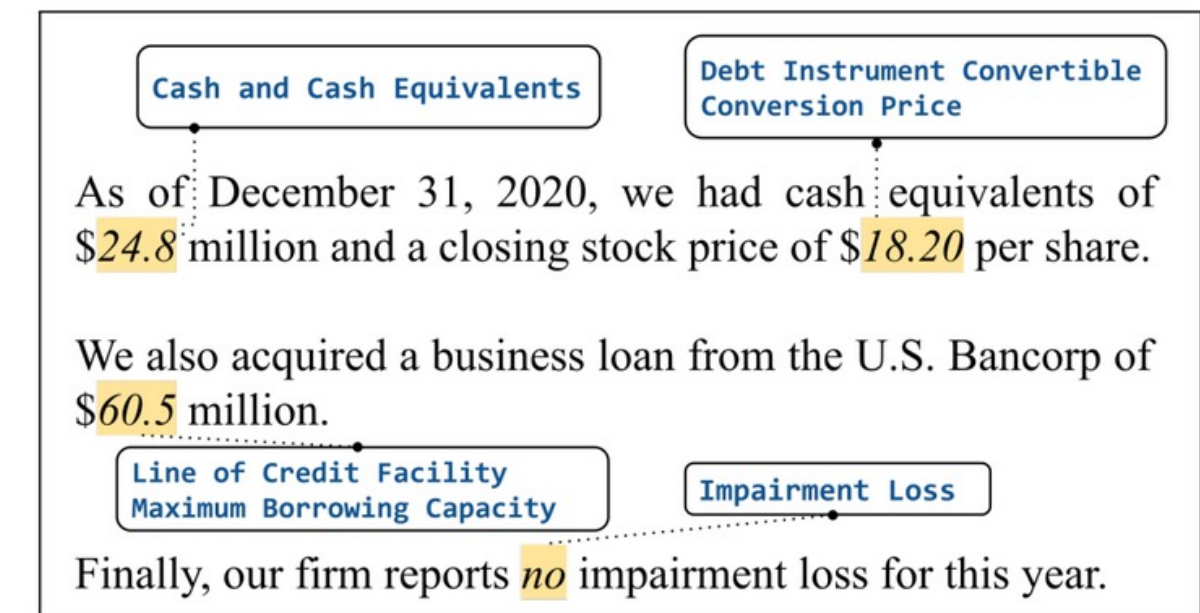
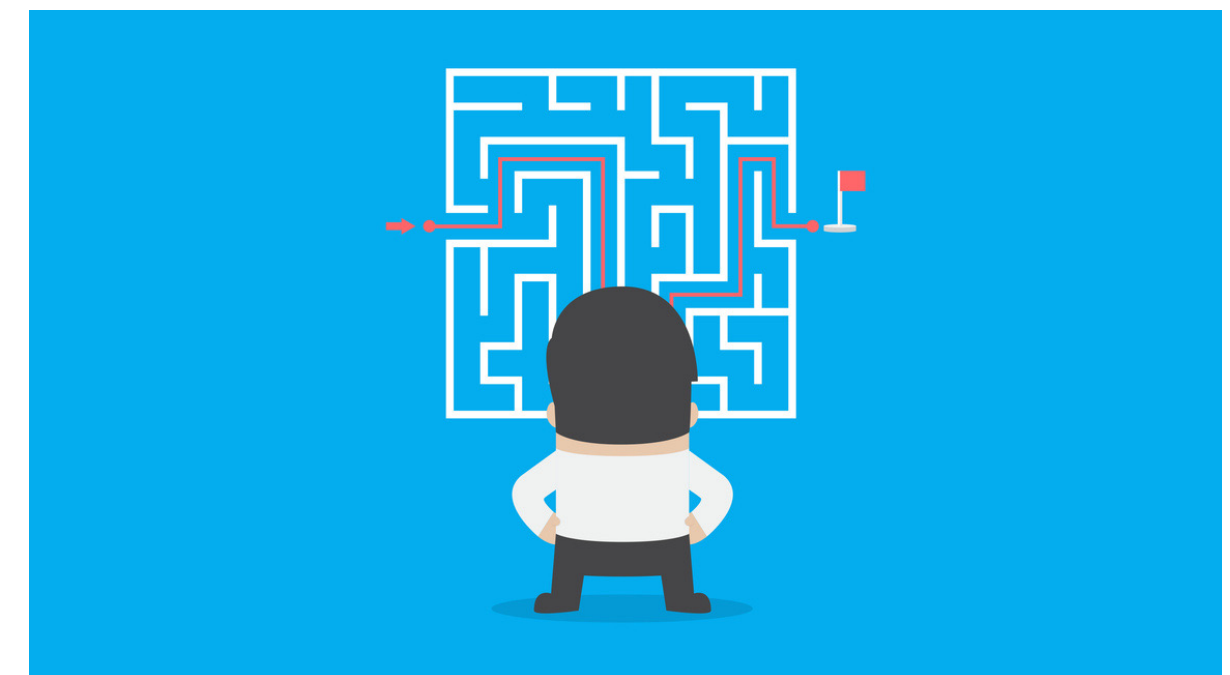


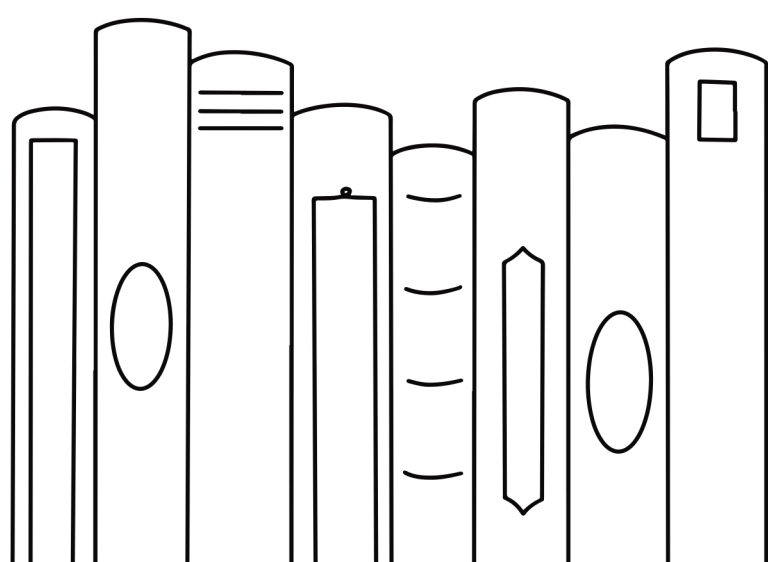
Figure 1: Sentences from FinER-139, with XBRL tags on numeric and non-numeric tokens. XBRL tags are actually XML-based and most tagged tokens are numeric.



# Related Work

## Entity Extraction

- XBRL tagging differs from typical entity extraction tasks
- There is a **much larger set of entity types (139)**
- Most tagged **tokens are numeric**
- The correct **tag depends** mostly **on context**; not the token itself



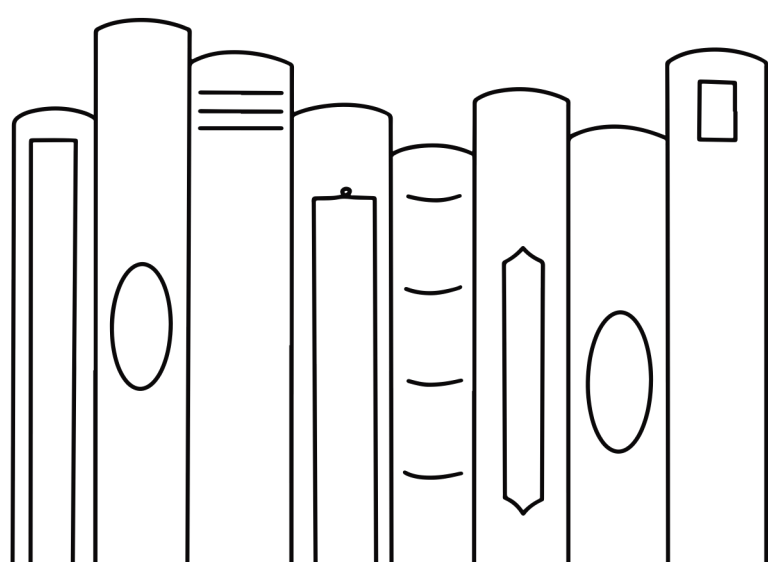
Dataset	Domain	Entity Types
CONLL-2003	Generic	4
ONTONOTES-V5	Generic	18
ACE-2005	Generic	7
GENIA	Biomedical	36
<a href="#">Chalkidis et al. (2019)</a>	Legal	14
<a href="#">Francis et al. (2019)</a>	Financial	9
FiNER-139 (ours)	Financial	<b>139</b>

Table 1: Examples of previous entity extraction datasets. Information about the first four from [Tjong Kim Sang and De Meulder \(2003\)](#); [Pradhan et al. \(2012\)](#); [Doddington et al. \(2004\)](#); [Kim et al. \(2003\)](#).

# Related Work

## Financial NER

- Small-scale datasets collected manually
- No deep learning methods
- **Classic named entity recognition extending the basic entity types**
- CRFs + Rules perform well on their tasks
- **We focus on a more detailed and fine-grained label set consisting of 139 actual XBRL tags, using several neural classifiers**



Paper	Method	Dataset Size	Labels
Kumar et al. (2016)	CRFs + Rules	10.000 sentences	4 (DATE, VALUE, ECONOMIC TERMS)
Hampton et al. (2016)	CRFs + Rules	-	10 (PERSON, ORG, TIME, PERCENT, TEMPORAL, etc)
Hampton et al. (2015)	Max. Entropy + Rules	-	10 (PERSON, ORG, TIME, PERCENT, TEMPORAL, etc)
Ours	Several Neural Classifiers	1.000.000 sentences	139 financial entities from XBRL taxonomies (Depreciation, LongTermDebt, OperatingLeaseCost, etc)



# Dataset (“FiNER-139”)

Subset	Sentences (S)	Avg. Tokens/S	Avg. Tags/S
Train	900,384	$44.7 \pm 33.9$	$1.8 \pm 1.2$
Dev	112,494	$45.4 \pm 35.9$	$1.7 \pm 1.2$
Test	108,378	$46.5 \pm 38.9$	$1.7 \pm 1.1$

Table 2: FiNER-139 statistics, using SPACY’s tokenizer and the 139 tags of this work ( $\pm$  standard deviation).

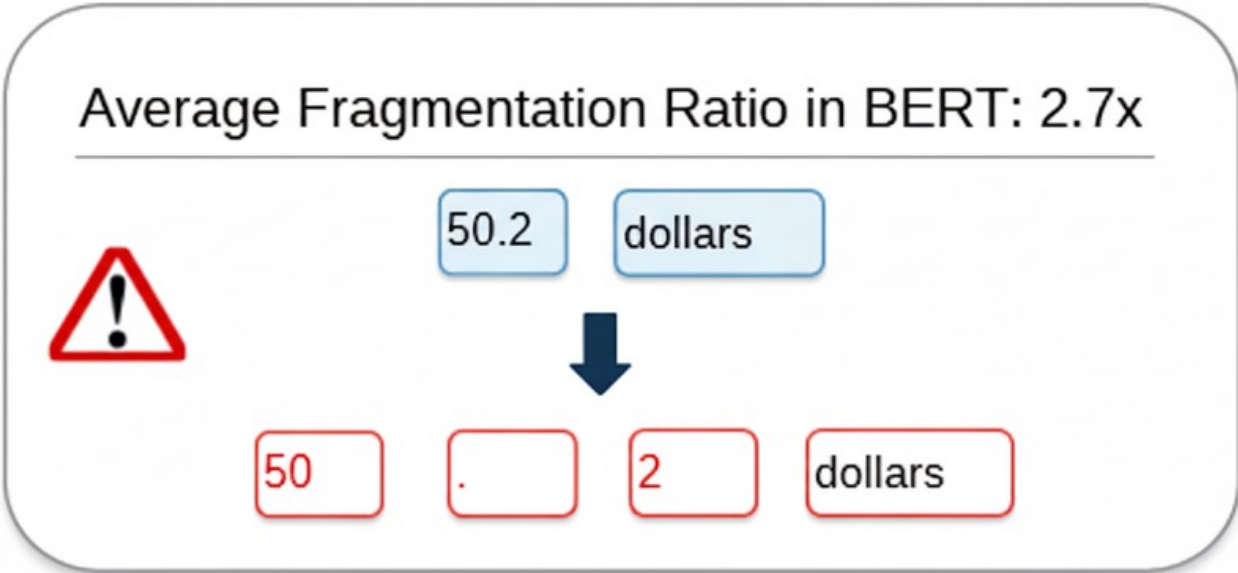
## FiNER-139 characteristics

- ➔ Downloaded **~10.000** quarterly/annual reports using the *edgar-crawler* toolkit from Chapter #1
- ➔ We currently focus on **recognizing** the top 139 **frequent XBRL financial tags**
- ➔ We use the Text Notes from Financial Statements Item Sections
- ➔ Each Text Notes Section has ~15 pages of ~50 XBRL tags scattered throughout it

# Experimental studies

## Baselines

- spaCy performs poorly, possibly due to the differences from typical entity extraction datasets
- Initially, **BERT** performs **worse than BILSTM** (words)
- **Why? BERT** produces **extreme fragmentation in \*numeric\* tokens -> meaningless subword units** (see example!)
- Controversial effect of CRF Layer; **CRF** Layer **helps** only in fixing misclassifications in **subword models**



Baseline methods	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
SPACY (words)	48.6 $\pm$ 0.4	37.6 $\pm$ 0.2
BILSTM (words)	<u>77.3</u> $\pm$ 0.6	<u>73.8</u> $\pm$ 1.8
BILSTM (subwords)	71.3 $\pm$ 0.2	68.6 $\pm$ 0.2
BERT (subwords)	75.1 $\pm$ 1.1	72.6 $\pm$ 1.4
BILSTM (words) + CRF	69.4 $\pm$ 1.2	67.3 $\pm$ 1.6
BILSTM (subwords) + CRF	76.2 $\pm$ 0.2	73.4 $\pm$ 0.3
BERT (subwords) + CRF	<b>78.0</b> $\pm$ 0.5	<b>75.2</b> $\pm$ 0.6



Table 3: Entity-level  $\mu$ -F<sub>1</sub> and m-F<sub>1</sub> (% , avg. of 3 runs with different random seeds,  $\pm$  std. dev.) on test data.

For the BILSTMs , we use 200-dimensional word2vec embeddings produced from ~200K financial documents downloaded from EDGAR. All models are tuned in a held-out dev dataset.

# Numbers + Transformers

remember this when ChatGPT got released? (2023)



 **Andrej Karpathy**   
@karpathy



I was given early access to Grok 3 earlier today, making me I think one of the first few who could run a quick vibe check.

## Random LLM "gotcha"s

I tried a few more fun / random LLM gotcha queries I like to try now and then. Gotchas are queries that specifically on the easy side for humans but on the hard side for LLMs, so I was curious which of them Grok 3 makes progress on.

- ✓ Grok 3 knows there are 3 "r" in "strawberry", but then it also told me there are only 3 "L" in LOLAPALOOZA. Turning on Thinking solves this.
- ✓ Grok 3 told me  $9.11 > 9.9$ . (common with other LLMs too), but again, turning on Thinking solves it.

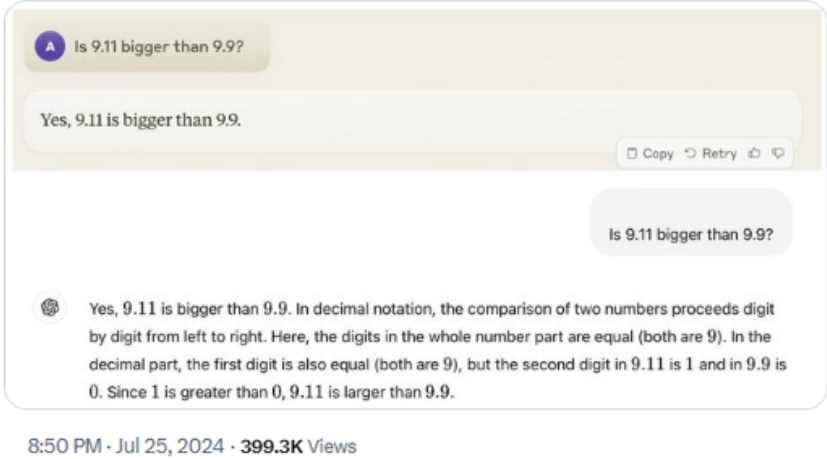
7:25 AM · Feb 18, 2025 · 3.6M Views

 **Andrej Karpathy**   
@karpathy

**Jagged Intelligence**

The word I came up with to describe the (strange, unintuitive) fact that state of the art LLMs can both perform extremely impressive tasks (e.g. solve complex math problems) while simultaneously struggle with some very dumb problems.

E.g. example from two days ago - which number is bigger, 9.11 or 9.9?  
Wrong.  
[x.com/karpathy/statu...](https://x.com/karpathy/status...)

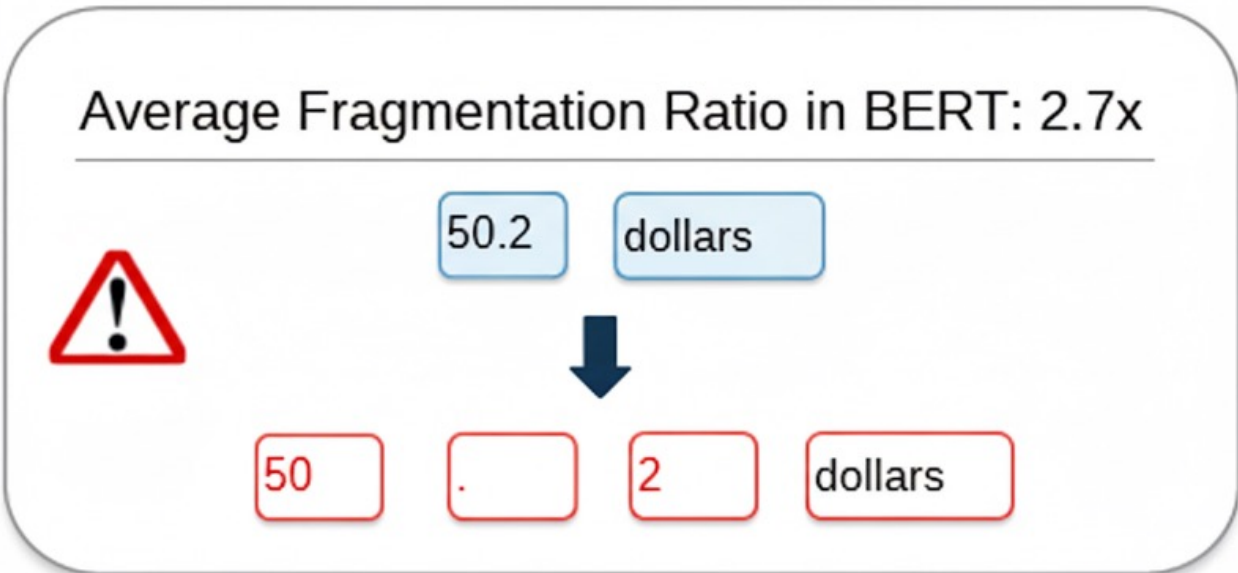




# Experimental studies

## Improving BERT with numbers: masking with [NUM]

- 💡 **BERT + [NUM]**: Replace all numbers with a special token [NUM]
- Tokens 50.2 and 40,233.12 will be mapped to [NUM]**
- Solves fragmentation issues ✓
- Better than vanilla BERT!**
  - Comes on par with BERT+CRF
- Limitation: No semantic representation for different shapes/magnitudes ✗
  - Intuition: Tokens that represent stocks (XX.X%) are expressed different than those that represent revenue (XX,XXX.XX)



BERT-based methods	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
BERT	75.1 ± 1.1	72.6 ± 1.4
BERT + CRF	<u>78.0</u> ± 0.5	<u>75.2</u> ± 0.6
BERT + [NUM]	78.3 ± 0.7	75.7 ± 0.9
BERT + [SHAPE]	<u>79.4</u> ± 0.2	<u>77.2</u> ± 0.2

Table 4: Entity-level Micro-F1 ( $\mu$ -F<sub>1</sub>) and Macro-F1 (m-F<sub>1</sub>) Score ± std (3 runs) on the test data for BERT-based models.

# Experimental studies

## Improving BERT with numbers: masking with [SHAPE]

- 💡 **BERT + [SHAPE]:** Normalize different magnitudes to different special tokens
- [SHAPE] methodology example:
  - 50.2 -> [XX.X]
  - 40,233.12 -> [XX,XXX.XX]
- Solves fragmentation issues ✓
- Semantic representation for different shapes/magnitudes ✓
- Better than all other methodologies!**

BERT-based methods	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
BERT	75.1 ± 1.1	72.6 ± 1.4
BERT + CRF	<u>78.0</u> ± 0.5	<u>75.2</u> ± 0.6
BERT + [NUM]	78.3 ± 0.7	75.7 ± 0.9
BERT + [SHAPE]	<u>79.4</u> ± 0.2	<u>77.2</u> ± 0.2

Table 4: Entity-level Micro-F1 ( $\mu$ -F<sub>1</sub>) and Macro-F1 (m-F<sub>1</sub>) Score ± std (3 runs) on the test data for BERT-based models.

Debt Carrying Value is net of \$ 5.2 million and \$ 6.4 million of deferred financing fees at March 31, 2019, and December 2018, respectively.

Deferred Finance Costs Net

Debt Carrying Value is net of \$ [NUM] million and \$ [NUM] million of deferred financing fees at March [NUM], [NUM], and December [NUM], respectively.

Deferred Finance Costs Net

Deferred Finance Costs Net

Debt Carrying Value is net of \$ [X.X] million and \$ [X.X] million of deferred financing fees at March [XX], [XXXX], and December [XXXX], respectively.

# Experimental studies

## In-domain knowledge

- 💡 **Does in-domain pre-training help BERT?**
- Not always! FIN-BERT (Yang et al., 2020) is worse than BERT!
- The better the representation of the numeric tokens ([NUM]/[SHAPE] tokens), the bigger the boost from the in-domain knowledge
- FIN-BERT + [SHAPE] > BERT + [SHAPE]

BERT-based methods	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
BERT	75.1 $\pm$ 1.1	72.6 $\pm$ 1.4
BERT + CRF	<u>78.0</u> $\pm$ 0.5	<u>75.2</u> $\pm$ 0.6
BERT + [NUM]	78.3 $\pm$ 0.7	75.7 $\pm$ 0.9
BERT + [SHAPE]	<u>79.4</u> $\pm$ 0.2	<u>77.2</u> $\pm$ 0.2
FIN-BERT	74.0 $\pm$ 1.1	71.3 $\pm$ 1.2
FIN-BERT + [NUM]	78.8 $\pm$ 0.3	76.3 $\pm$ 0.5
FIN-BERT + [SHAPE]	<u>80.1</u> $\pm$ 1.4	<u>77.8</u> $\pm$ 2.0

Table 4: Entity-level Micro-F1 ( $\mu$ -F<sub>1</sub>) and Macro-F1 (m-F<sub>1</sub>) Score  $\pm$  std (3 runs) on the test data for BERT-based models.



# Experimental studies

## In-domain knowledge

- 💡 **Does in-domain pre-training help BERT?**
- Not always! FIN-BERT (Yang et al., 2020) is worse than BERT!
- The better the representation of the numeric tokens ([NUM]/[SHAPE] tokens), the bigger the boost from the in-domain knowledge
- FIN-BERT + [SHAPE] > BERT + [SHAPE]
- SEC-BERT (ours) is pre-trained on 200K annual reports from SEC (EDGAR-CORPUS, Loukas et al., 2021)

BERT-based methods	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
BERT	75.1 $\pm$ 1.1	72.6 $\pm$ 1.4
BERT + CRF	<u>78.0</u> $\pm$ 0.5	<u>75.2</u> $\pm$ 0.6
BERT + [NUM]	78.3 $\pm$ 0.7	75.7 $\pm$ 0.9
BERT + [SHAPE]	<u>79.4</u> $\pm$ 0.2	<u>77.2</u> $\pm$ 0.2
FIN-BERT	74.0 $\pm$ 1.1	71.3 $\pm$ 1.2
FIN-BERT + [NUM]	78.8 $\pm$ 0.3	76.3 $\pm$ 0.5
FIN-BERT + [SHAPE]	<u>80.1</u> $\pm$ 1.4	<u>77.8</u> $\pm$ 2.0
SEC-BERT (ours)	75.7 $\pm$ 0.1	72.6 $\pm$ 0.4

Table 4: Entity-level Micro-F1 ( $\mu$ -F<sub>1</sub>) and Macro-F1 (m-F<sub>1</sub>) Score  $\pm$  std (3 runs) on the test data for BERT-based models.

# Experimental studies

## In-domain knowledge

- 💡 **Does in-domain pre-training help BERT?**
- Not always! FIN-BERT (Yang et al., 2020) is worse than BERT!
- The better the representation of the numeric tokens ([NUM]/[SHAPE] tokens), the bigger the boost from the in-domain knowledge
- $\text{FIN-BERT} + [\text{SHAPE}] > \text{BERT} + [\text{SHAPE}]$
- SEC-BERT (ours) is pre-trained on 200K annual reports from SEC (EDGAR-CORPUS, Loukas et al., 2021)
- Pre-training SEC-BERT using special numeric tokens is a better strategy than trying to acquire this knowledge only during fine-tuning
- SEC-BERT-SHAPE > all other methods

BERT-based methods	$\mu\text{-F}_1$	m-F <sub>1</sub>
BERT	$75.1 \pm 1.1$	$72.6 \pm 1.4$
BERT + CRF	<u><math>78.0 \pm 0.5</math></u>	<u><math>75.2 \pm 0.6</math></u>
BERT + [NUM]	$78.3 \pm 0.7$	$75.7 \pm 0.9$
BERT + [SHAPE]	<u><math>79.4 \pm 0.2</math></u>	<u><math>77.2 \pm 0.2</math></u>
FIN-BERT	$74.0 \pm 1.1$	$71.3 \pm 1.2$
FIN-BERT + [NUM]	$78.8 \pm 0.3$	$76.3 \pm 0.5$
FIN-BERT + [SHAPE]	<u><math>80.1 \pm 1.4</math></u>	<u><math>77.8 \pm 2.0</math></u>
SEC-BERT (ours)	$75.7 \pm 0.1$	$72.6 \pm 0.4$
SEC-BERT-NUM (ours)	$80.4 \pm 1.4$	$78.3 \pm 1.6$
SEC-BERT-SHAPE (ours)	<b><math>82.1 \pm 0.1</math></b>	<b><math>80.1 \pm 0.1</math></b>

Table 4: Entity-level Micro-F1 ( $\mu\text{-F}_1$ ) and Macro-F1 (m-F<sub>1</sub>) Score  $\pm$  std (3 runs) on the test data for BERT-based models.

# Off-the-shelf LLMs are not suitable 😓

- We also tested LLMs on FiNER-139 (zero-shot setting)
- Combining proprietary LLMs + prompt engineering + NUM/SHAPE masking, LLMs still struggle:

	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
Claude Sonnet 4	8.31%	7.88%
Claude Sonnet 4 + [NUM]	9.64%	9.61%
Claude Sonnet 4 + [SHAPE]	<b>10.60%</b>	<b>10.35%</b>

Dataset	Metric	GPT-3.5-turbo	GPT-4	Gemini 1.0	LLaMA2-70B	LLaMA3-8B	FinMA-7B	Mistral-7B
FNXL	EntityF1	0.00	0.01	0.00	0.00	0.00	0.00	0.00

Table 3.10: The zero-shot performance of different LLMs on FNXL, according to the FinBen paper (Xie et al., 2024). All results are the average of three runs.



# Additional experiments

Do [NUM] and [SHAPE] work in BiLSTMs too?

- Effectiveness of **pseudo-tokens** – generalization?
- We **incorporated** them in the **BiLSTMs** operating on **subword embeddings**
- We **replace each number** by a single **[NUM]** pseudo-token or one of 214 **[SHAPE]** pseudo-tokens.
- The replacement happens when pre-training word2vec subword embeddings; hence, an embedding is obtained for each pseudo-token
- Results further support our hypothesis
- **The proposed pseudo-tokens can help subword models generalize over numeric expressions in such tasks!**

	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
BILSTM (subwords)	71.3 $\pm$ 0.2	68.6 $\pm$ 0.2
BILSTM (subwords) + CRF	76.2 $\pm$ 0.2	73.4 $\pm$ 0.3
BILSTM-NUM (subwords)	75.6 $\pm$ 0.3	72.7 $\pm$ 0.4
BILSTM-SHAPE (subwords)	<b>76.8 <math>\pm</math> 0.2</b>	<b>74.1 <math>\pm</math> 0.3</b>

Table 6: Entity-level  $\mu$ -F<sub>1</sub> and m-F<sub>1</sub> (% , avg. of 3 runs with different random seeds,  $\pm$  std. dev.) on test data for BILSTM models with [NUM] and [SHAPE] tokens.

# Additional experiments

## A business use-case

- XBRL Tagging is derived from a real-world need!
- Practical use case: XBRL Tag recommendation
- Evaluate with business metrics: **hits@k**
- We use the model to return the k most probable XBRL tags
- If the correct tag is among the top k, add +1
- Divide by the total number of tokens to be annotated
- **Hits@3: 96.7%**
- **Hits@5: 98.6%**
- **Hits@10: 99.4%**
- Results: An end-user (auditor) has to examine at most 5-10 recommended tags (out of 139) to find the correct one

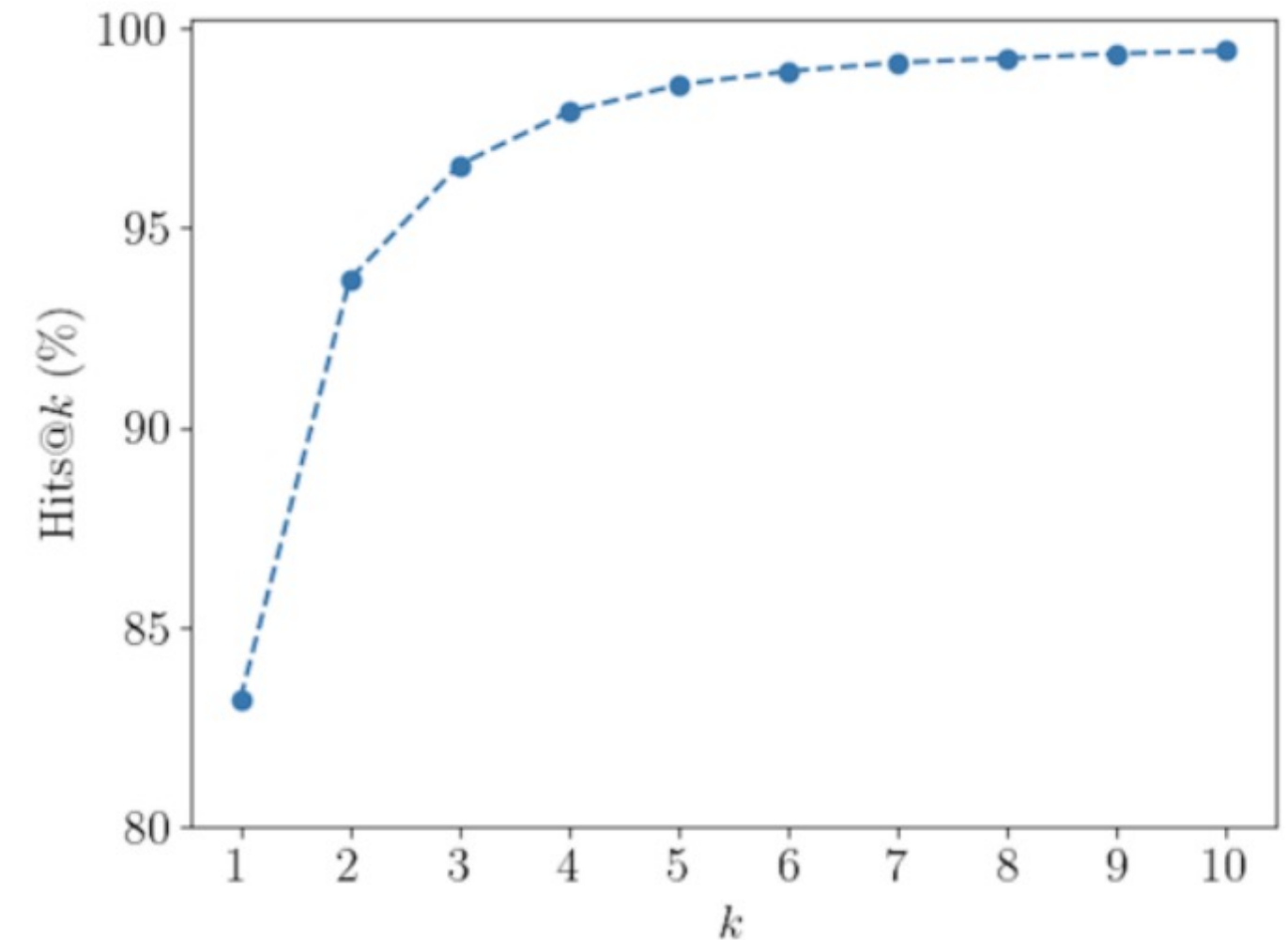
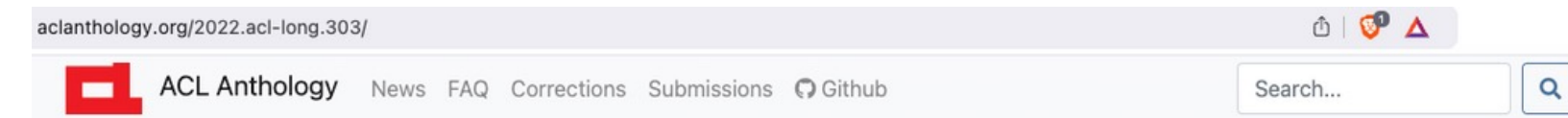


Figure 4: Hits@ $k$  results (% , avg. of 3 runs with different random seeds) on test data, for different  $k$  values. Standard deviations were very small and are omitted.

# Summary

- **New real-word NLP task** for the financial domain
- Released **FiNER-139**, a **dataset** with 1.1M sentences containing 139 XBRL labels for **XBRL Tagging**
- Experimented with several neural classifiers, showing that **a BILSTM outperforms BERT** (and LLMs) due to the excessive numeric token fragmentation of the latter
- **Alleviated the overfragmentation of transformers** by proposing **special tokens** to generalize over the **shapes and magnitudes** of numeric expressions
- We **pre-trained and released** our own **BERT model family, SEC-BERT**, leading to improved performance
- Publication at **ACL 2022** (98 citations)
- [Goldman Sachs](#) did a follow-up on our work and published it in [ACL 2023](#)

[Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. 2022. FiNER: Financial Numeric Entity Recognition for XBRL Tagging. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics \(Volume 1: Long Papers\), pages 4419–4431, Dublin, Ireland. Association for Computational Linguistics.](#)



## FiNER: Financial Numeric Entity Recognition for XBRL Tagging

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, Georgios Paliouras

### Abstract

Publicly traded companies are required to submit periodic reports with eXtensive Business Reporting Language (XBRL) word-level tags. Manually tagging the reports is tedious and costly. We, therefore, introduce XBRL tagging as a new entity extraction task for the financial domain and release FiNER-139, a dataset of 1.1M sentences with gold XBRL tags. Unlike typical entity extraction datasets, FiNER-139 uses a much larger label set of 139 entity types. Most annotated tokens are numeric, with the correct tag per token depending mostly on context, rather than the token itself. We show that subword fragmentation of numeric expressions harms BERT's performance, allowing word-level BILSTMs to perform better. To improve BERT's performance, we propose two simple and effective solutions that replace numeric expressions with pseudo-tokens reflecting original token shapes and numeric magnitudes. We also experiment with FIN-BERT, an existing BERT model for the financial domain, and release our own BERT (SEC-BERT), pre-trained on financial filings, which performs best. Through data and error analysis, we finally identify possible limitations to inspire future work on XBRL tagging.

[PDF](#)[Cite](#)[Search](#)[Code](#)

**Anthology ID:** 2022.acl-long.303  
**Volume:** Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)  
**Month:** May  
**Year:** 2022  
**Address:** Dublin, Ireland  
**Editors:** Smaranda Muresan, Preslav Nakov, Aline Villavicencio  
**Venue:** ACL

- <https://huggingface.co/nlpauieb/sec-bert-base>
- <https://huggingface.co/nlpauieb/sec-bert-num>
- <https://huggingface.co/nlpauieb/sec-bert-shape>
- <https://huggingface.co/datasets/nlpauieb/finer-139>



# Overview of Chapter #2

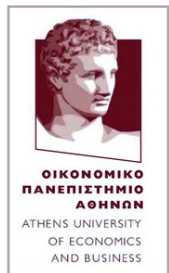
## FiNER: Financial Numeric Entity Recognition for XBRL Tagging

- Published at **ACL 2022** (A\* CORE ranking conference)
- Problem:** token overfragmentation of Transformer models in numbers
- Solution:** new tokenization method for pre-training and fine-tuning so transformers learn better number representations
- Created new BERT models & dataset  
(<https://huggingface.co/nlpaueb/sec-bert-base> & <https://huggingface.co/datasets/nlpaueb/finer-139>)
  - downloaded around 7,000 times
- Solves a real-life business problem task of XBRL Tagging
  - US SEC requires publicly traded companies to tag their documents with XBRL tags



## 1 granted US patent based on this methodology/task

- 1 granted US patent
- Also submitted to the EU / World Patent Office
- Assigned to **Ernst and Young (EY) & NCSR Demokritos** for commercial use



CITED BY	YEAR
98	2022

Debt Carrying Value is net of \$ 5.2 million and \$ 6.4 million of deferred financing fees at March 31, 2019, and December 2018, respectively.

Deferred Finance Costs Net

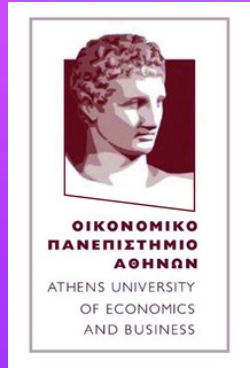
Debt Carrying Value is net of \$ [NUM] million and \$ [NUM] million of deferred financing fees at March [NUM], [NUM], and December [NUM], respectively.

Deferred Finance Costs Net

Deferred Finance Costs Net

Debt Carrying Value is net of \$ [X.X] million and \$ [X.X] million of deferred financing fees at March [XX], [XXXX], and December [XXXX], respectively.

(12) <b>United States Patent</b> Loukas et al.	(10) <b>Patent No.:</b> US 12,333,236 B2 (45) <b>Date of Patent:</b> Jun. 17, 2025
(54) <b>SYSTEM AND METHOD FOR AUTOMATICALLY TAGGING DOCUMENTS</b>	(58) <b>Field of Classification Search</b> None See application file for complete search history.
(71) Applicant: <b>National Centre for Scientific Research "Demokritos"</b> , Agia Paraskevi (GR)	(56) <b>References Cited</b> U.S. PATENT DOCUMENTS 10,817,619 B1 * 10/2020 Kolli ..... G06F 21/552 10,997,369 B1 * 5/2021 Frazier ..... G06F 40/284 (Continued) FOREIGN PATENT DOCUMENTS CN 112257442 1/2021 EP 4124988 2/2023 WO WO 2023/006773 2/2023
(72) Inventors: <b>Eleftherios Panagiotis Loukas</b> , Agia Paraskevi (GR); <b>Eirini Spyropoulou</b> , Agia Paraskevi (GR); <b>Prodromos Malakasiotis</b> , Agia Paraskevi (GR); <b>Emmanouil Fergadiotis</b> , Agia Paraskevi (GR); <b>Ilias Chalkidis</b> , Agia Paraskevi (GR); <b>Ioannis Androutsopoulos</b> , Agia Paraskevi (GR); <b>Georgios Paliouras</b> , Agia Paraskevi (GR)	



# Chapter #3

Research Question #3: “What is the most accurate and cost-efficient way for resource-limited intent recognition? Should one use BERT-based models or LLMs? How?”

**Resource-Limited Intent Recognition**



## Motivation

### User queries

My card is needed soon.

My transfer got declined!

How can I replace my expired card?



### Intent labels

Card Delivery Estimate

Declined Transfer

Card About To Expire

Dataset  
/banking77

- Intent detection is a classification task, which can be solved in numerous ways
  - Full-Data Setting (>1,000 samples per class)
    - In business settings, it is unfeasible to get so much data :(
  - Few-shot Setting (1-20 samples per class)
    - Contrastive Learning with MLMs ( $\leq 20$  samples)
    - In-Context Learning with LLMs like GPT-3.5 and GPT-4 (1-5 samples)
      - LLMs might work well, but they cost lots of \$\$\$ :(

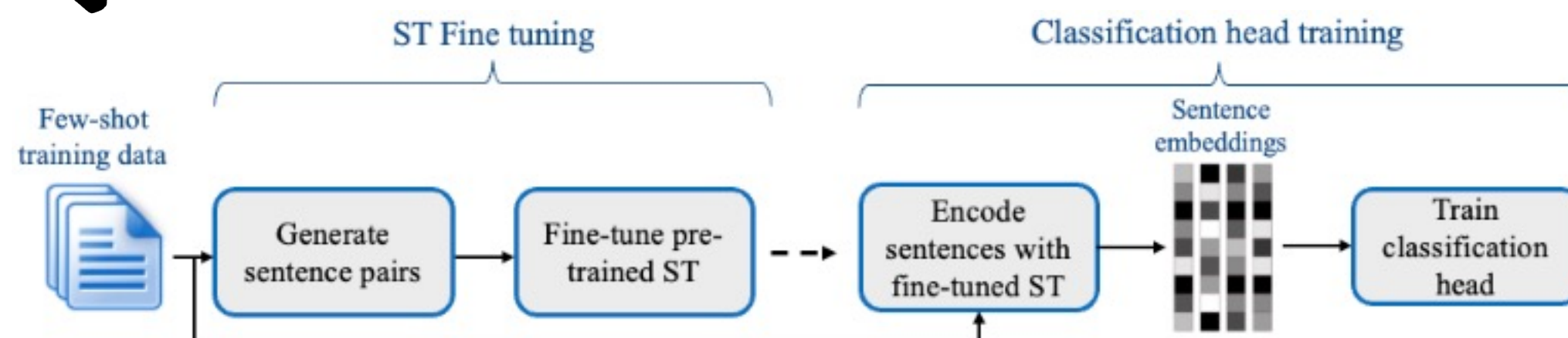


1. "Breaking the Bank with ChatGPT: Few-Shot Text Classification for Finance". L. Loukas, I. Stogiannidis, P. Malakasiotis, S. Vassos. **FinNLP @ IJCAI 2023**
2. "Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking". L. Loukas, I. Stogiannidis, O. Diamantopoulos, P. Malakasiotis, S. Vassos. **(ACM ICAIF 2023) - Selected by ACM for ACM's Research Highlights**

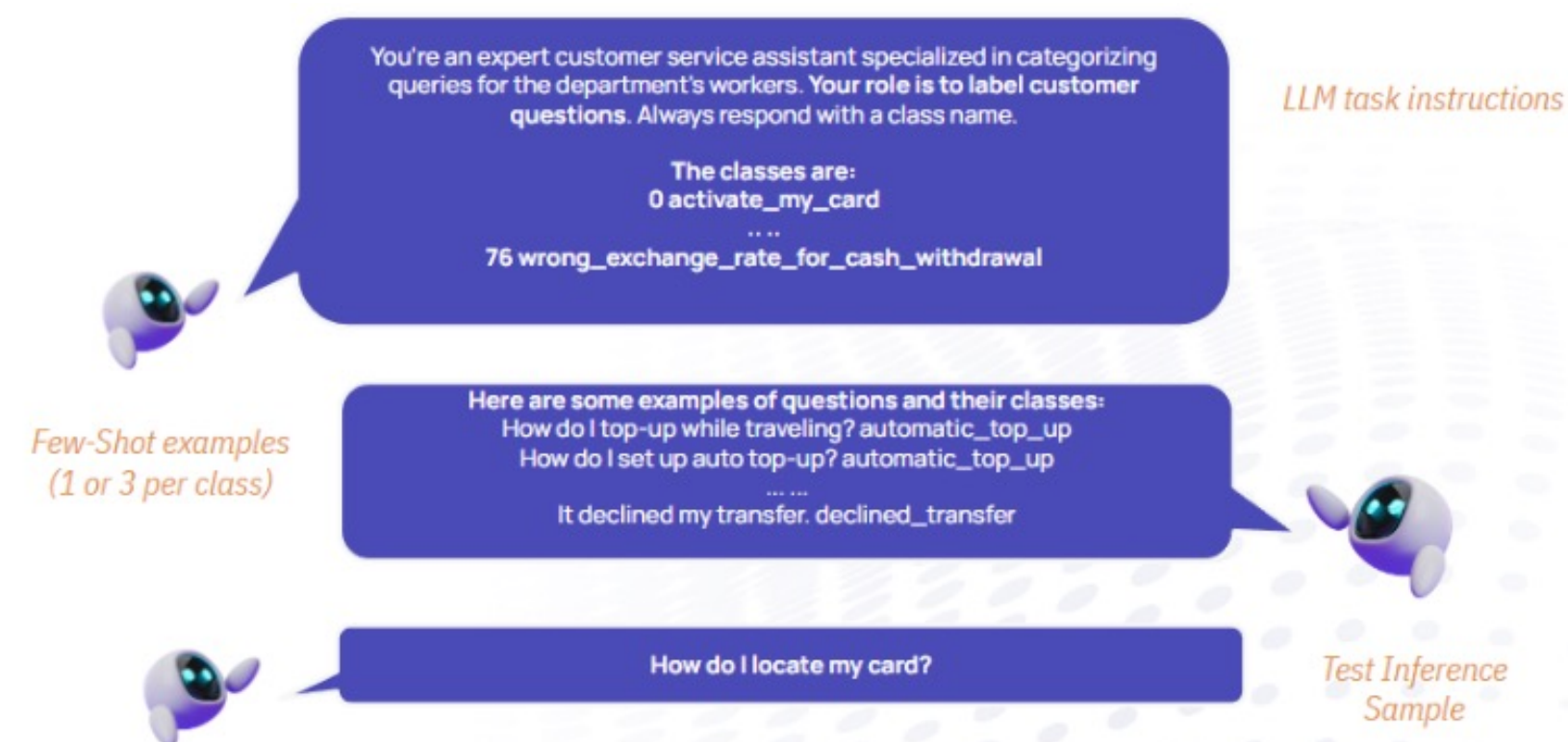


We tackle text classification *mainly* in **Few-Shot Settings** (limited samples per class) via 2 ways:

- Contrastive Learning (SetFit) with BERT-based models
- In-Context Learning (Prompting) with Large Language Models (LLMs)



An overview of Contrastive Learning (SetFit), as used in MLs. Setfit was first introduced by HuggingFace (Tunstall et al., 2022). It utilizes Sentence Transformers (like MPNet) in a Siamese + Supervised Fine-Tuning Manner by having an objective function to minimize the distance between samples of the same labels. The result is that it produces rich vector representations, even when providing only 10 to 20 samples per class for your text classification problem.



An overview of In-Context Learning, as used in LLMs. We leverage the pre-trained knowledge of LLMs and extend it with our specific task instructions and a few examples per class. This is done for each inference sample. We use a variety of proprietary LLMs, like OpenAI GPT-3.5 and GPT-4, Anthropic Claude 2, and Cohere's Command-Nightly.

## Results (BERT-based models)

*[full-data and few-shot with SetFit aka contrastive learning]*

- We report micro- and macro- F1 Scores.
- MPNet-v2 achieves competitive results across few-shot settings with  $\geq 3$  samples using SetFit
- **When trained on only 3 samples, MPNet-v2 achieves scores of 76.7  $\mu$ -F1 and 75.9 m-F1**
- As we increase the samples, the performance improves, reaching a 91.2 micro-F1 and 91.3 macro-F1 score with 20 samples per class
- **This is only 3 percentage points (pp) lower than fine-tuning the model in a Full-Data Setting**

Methods	Setting	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
Mehri and Eric (2021)	Full-Data	93.8	NA
Mehri and Eric (2021)	10-shot x 77	85.9	NA
Ying and Thomas (2022)	Full-Data	NA	92.0
MPNet-v2	Full-Data	<b>94.1</b>	<b>94.1</b>
MPNet-v2 (SetFit)	1-shot x 77	<b>57.4</b>	<b>55.9</b>
GPT-3.5 (representative samples)	1-shot x 77	<b>75.2</b>	<b>74.3</b>
GPT-3.5 (random samples)	1-shot x 77	74.0	72.3
GPT-4 (representative samples)	1-shot x 77	<b>80.4</b>	<b>78.1</b>
GPT-4 (random samples)	1-shot x 77	77.6	76.7
Command-nightly (representative samples)	1-shot x 77	58.4	57.8
Anthropic Claude 1 (representative samples)	1-shot x 77	73.8	72.1
Anthropic Claude 2 (representative samples)	1-shot x 77	76.8	75.1
MPNet-v2 (SetFit)	3-shot x 77	<b>76.7</b>	<b>75.9</b>
GPT-3.5 (random samples)	3-shot x 77	57.9	59.8
GPT-3.5 (representative samples)	3-shot x 77	<b>65.5</b>	<b>65.3</b>
GPT-4 (representative samples)	3-shot x 77	<b>83.1</b>	<b>82.7</b>
GPT-4 (random samples)	3-shot x 77	74.2	73.7
MPNet-v2 (SetFit)	5-shot x 77	<b>83.5</b>	<b>83.3</b>
MPNet-v2 (SetFit)	10-shot x 77	88.1	88.1
MPNet-v2 (SetFit)	15-shot x 77	90.6	90.5
MPNet-v2 (SetFit)	20-shot x 77	<b>91.2</b>	<b>91.3</b>

## Results (LLMs)

*[few-shot with in-context learning]*

- **Nearly all LLMs achieve competitive results** despite shown only 1 sample
- **GPT-4 outperforms the BERT-based model by 23 points (in the 1-shot setting)**
- OpenAI **GPT-4 is superior** to Anthropic's Claude and Command-nightly
  - reminder: research was done at times when no benchmarks or techniques like prefix caching were around! (May 2023)
  - one of the first studies in LLMs and cost-efficiency
- **GPT-4 shows the best performance when shown 3 samples per class**, but performance is comparable vs. SetFit (on 3/5 samples)
- **Using human-curated *representative* samples leads to better in-context learning results!**
  - <https://huggingface.co/datasets/helvia/banking77-representative-samples> (our annotated subset)

Methods	Setting	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
Mehri and Eric (2021)	Full-Data	93.8	NA
Mehri and Eric (2021)	10-shot x 77	85.9	NA
Ying and Thomas (2022)	Full-Data	NA	92.0
MPNet-v2	Full-Data	<b>94.1</b>	<b>94.1</b>
MPNet-v2 (SetFit)	1-shot x 77	<b>57.4</b>	<b>55.9</b>
GPT-3.5 (representative samples)	1-shot x 77	<b>75.2</b>	<b>74.3</b>
GPT-3.5 (random samples)	1-shot x 77	74.0	72.3
GPT-4 (representative samples)	1-shot x 77	<b>80.4</b>	<b>78.1</b>
GPT-4 (random samples)	1-shot x 77	77.6	76.7
Command-nightly (representative samples)	1-shot x 77	58.4	57.8
Anthropic Claude 1 (representative samples)	1-shot x 77	73.8	72.1
Anthropic Claude 2 (representative samples)	1-shot x 77	76.8	75.1
MPNet-v2 (SetFit)	3-shot x 77	<b>76.7</b>	<b>75.9</b>
GPT-3.5 (random samples)	3-shot x 77	57.9	59.8
GPT-3.5 (representative samples)	3-shot x 77	<b>65.5</b>	<b>65.3</b>
GPT-4 (representative samples)	3-shot x 77	<b>83.1</b>	<b>82.7</b>
GPT-4 (random samples)	3-shot x 77	74.2	73.7
MPNet-v2 (SetFit)	5-shot x 77	<b>83.5</b>	<b>83.3</b>
MPNet-v2 (SetFit)	10-shot x 77	88.1	88.1
MPNet-v2 (SetFit)	15-shot x 77	90.6	90.5
MPNet-v2 (SetFit)	20-shot x 77	<b>91.2</b>	<b>91.3</b>





# Cost Analysis

Most **LLM inference** today is done through **provider**-hosted **APIs**, and **commercial** closed-source models can be very expensive to use \$ \$ (2023 pricing below)

Model	Setting	Micro-F1↑	Cost↓
Standard Few-Shot			
GPT-4	1-shot x 77	80.4	\$620
GPT-3.5	1-shot x 77	75.2	\$31
Anthropic Claude 2	1-shot x 77	76.8	\$15
Command-nightly	1-shot x 77	58.4	\$22
GPT-3.5	3-shot x 77	65.5	\$62
GPT-4	3-shot x 77	83.1	\$740

# Cost-effective LLM inference with Dynamic Few-Shot Prompting

- **Can we cut API costs** on this text classification problem? **YES!** 😊
- ❌ Right now, we feed the LLM N examples per class (classic N-shot settings)
- ✓ Instead, we found out that **during inference**, we can **retrieve** only the **top K** similar **and** their **labels** , and **perform better** while **reducing** context size and **API costs**
  - We call this **“Dynamic Few-Shot Prompting”** (kNN augmentation inspired by Liu et al., 2022)
  - It is like using **“RAG”** (Retrieval-Augmented Generation) for your LLM classification prompt

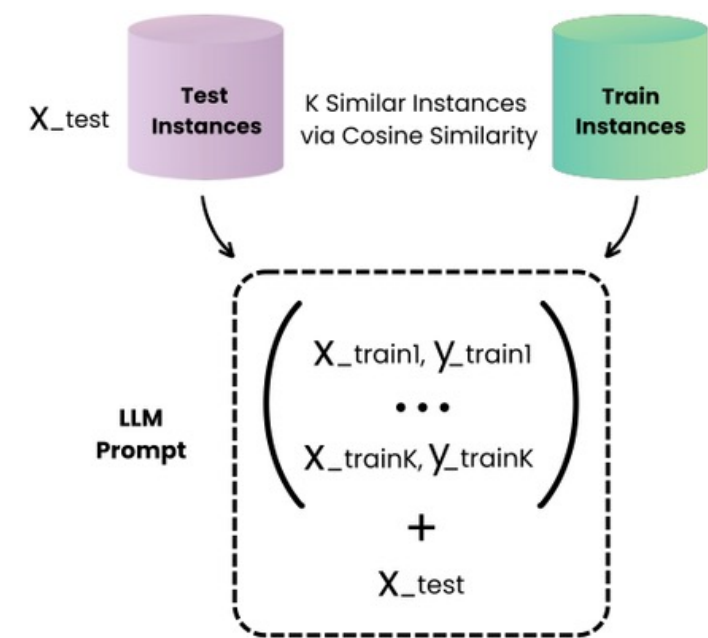


Figure 3: Dynamic LLM prompt construction through Retrieval-Augmented Generation (RAG), using cosine similarity for in-context data selection. We use K=5, 10, 20.

Model	Setting	Micro-F1↑	Cost↓
Standard Few-Shot			
GPT-4	1-shot x 77	80.4	\$620
GPT-3.5	1-shot x 77	75.2	\$31
Anthropic Claude 2	1-shot x 77	76.8	\$15
Command-nightly	1-shot x 77	58.4	\$22
GPT-3.5	3-shot x 77	65.5	\$62
GPT-4	3-shot x 77	83.1	\$740
Dynamic Few-Shot (RAG)			
GPT-4	5 similar (RAG)	84.5	\$205
Anthropic Claude 2	5 similar (RAG)	84.8	\$33
GPT-4	10 similar (RAG)	81.2	\$230
Anthropic Claude 2	10 similar (RAG)	85.2	\$37
GPT-4	20 similar (RAG)	87.7	\$270
Anthropic Claude 2	20 similar (RAG)	85.5	\$42

- Using **“Dynamic Few-Shot Prompting”** is **better and cheaper than classic Few-Shot** (see GPT-4)
- Highlight: Claude 2 on K=20 similar yields 85.5% on 42\$ vs. GPT-4's 83.1% on 740\$ 🤯

**In-Context Learning for Text Classification with Many Labels**

Aristides Milios<sup>1</sup>, Siva Reddy<sup>1,2,3</sup>, Dzmitry Bahdanau<sup>1,2</sup>  
Mila and McGill University<sup>1</sup>, ServiceNOW Research<sup>2</sup>, Facebook CIFAR AI Chair<sup>3</sup>  
{aristides.milios, siva.reddy, bahdanau}@mila.quebec

Research from the **MILA lab** also show that the same method works well in even more datasets and other open-weight LLMs! (paper at EMNLP 2023, **2 weeks later**). Related studies (Lewis et al., 2020 & Liu et al., 2022) show that kNN augmentation helps performance in older text generation models.



# Cost-effective LLM inference with Dynamic Few-Shot Prompting

## Summary (cost-wise):

- APIs charge based on token usage
- Standard Few-Shot:  $\text{Cost} \propto S \times C$  (shots  $\times$  classes)
- Dynamic Few-Shot:  $\text{Cost} \propto K$  (retrieved examples only during inference)

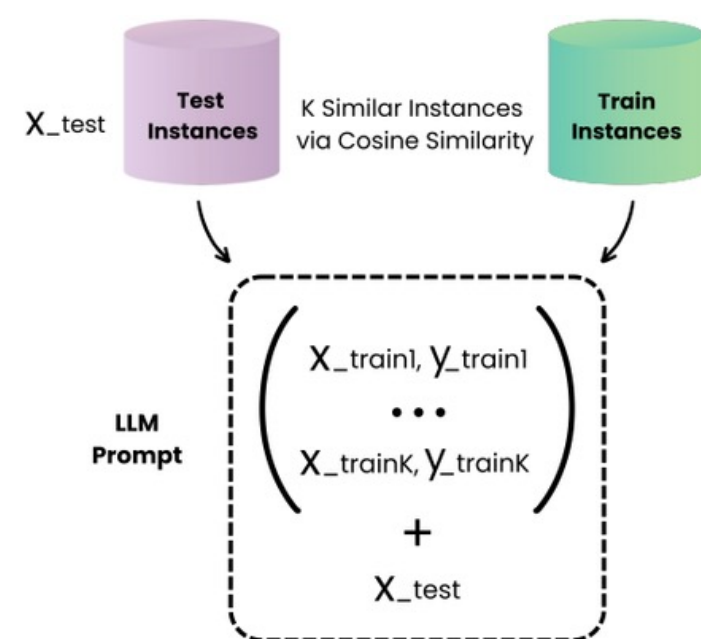


Figure 3: Dynamic LLM prompt construction through Retrieval-Augmented Generation (RAG), using cosine similarity for in-context data selection. We use K=5, 10, 20.

Model	Setting	Micro-F1↑	Cost↓
Standard Few-Shot			
GPT-4	1-shot x 77	80.4	\$620
GPT-3.5	1-shot x 77	75.2	\$31
Anthropic Claude 2	1-shot x 77	76.8	\$15
Command-nightly	1-shot x 77	58.4	\$22
GPT-3.5	3-shot x 77	65.5	\$62
GPT-4	3-shot x 77	83.1	\$740
Dynamic Few-Shot (RAG)			
GPT-4	5 similar (RAG)	84.5	\$205
Anthropic Claude 2	5 similar (RAG)	84.8	\$33
GPT-4	10 similar (RAG)	81.2	\$230
Anthropic Claude 2	10 similar (RAG)	85.2	\$37
GPT-4	20 similar (RAG)	87.7	\$270
Anthropic Claude 2	20 similar (RAG)	85.5	\$42


We could also formalize the cost-efficiency more by creating a Cost-Effectiveness Score (CES)

- **CES = Micro-F1 / Cost (performance per dollar spent)**
  - For example:
    - **GPT-4 & Standard 3-shot** setting: **0.11**
    - **GPT-4 & Dynamic Few-shot with K=5: 0.41** (3.7× better!)
    - **GPT 4 & Dynamic Few-Shot with K=20: 0.32** (diminishing returns, but still better than Standard Few-Shot)
    - **Anthropic Claude 2 & Dynamic Few-Shot with K=5: 2.57**
- One could use this for tuning in small batches in the development set to find out the best approach for their test/inference set



# Our paper recognized by others in the communtiy

Our paper reshared by [HuggingFace](#) ,emphasizing that BERT-based models and Contrastive Learning is a viable alternative vs. . heavily-paid LLM APIs.



Julien SIMON • Following

Chief Evangelist, HuggingFace

10mo • Edited •

Interesting insights in "Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking" <https://lnkd.in/e3FCvHAB>

1) The MPNet-v2 Sentence Transformer, fine-tuned with [Hugging Face](#) SetFit, outperforms GPT-3.5 on 3-shot prompting.

2) With 5 shots, it also outperforms 3-shot GPT-4... and surely, if you can find 3 examples per class, you can find 5.

3) When fine-tuned on the full dataset, MPNet-v2 blows away all large general-purpose models.

4) Keep in mind that MPNet-v2 is a 438MB model, which runs nice and fast enough on a modern CPU. The cost/performance advantage over GPT-4 is \*huge\*.

5) In general, using an LLM for extractive tasks isn't a great idea. Here's another example where FinBERT crushes GPT-4 <https://lnkd.in/e4hJhUc8>


6) The paper also hints (again) that many-shot prompting suffers from the "lost in the middle" effect first introduced in <https://lnkd.in/e-8HPUVW>.

7) Last but not least, the paper shows that RAG is a better option than many-shot prompting, although they don't test open-source models. Check out <https://lnkd.in/e-8HPUVW> for a good study on RAG with large-context models (including Llama 2 70B 32K).

Pretty much what I've been saying for a while: find the smallest open-source model that can do the job and fine-tune it on your data 😊

Methods	Setting	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
Mehri and Eric [28]	Full-Data	93.8	NA
Mehri and Eric [28]	10-shot	85.9	NA
Ying and Thomas [46]	Full-Data	NA	92.0
MPNet-v2	Full-Data	<b>94.1</b>	<b>94.1</b>
MPNet-v2 (SetFit)	1-shot	57.4	55.9
GPT-3.5 (representative samples)	1-shot	75.2	74.3
GPT-3.5 (random samples)	1-shot	74.0	72.3
GPT-4 (representative samples)	1-shot	<b>80.4</b>	<b>78.1</b>

Our paper reshared by Pascal Biese (author of LLMWatch.com, GenAI newsletter with 60k+ followers), explaining how Dynamic Few-Shot Prompting (essentially a RAG mechanism for classification) can help.



Pascal Biese • 1st

Daily AI highlights for 60k+ experts • AI/ML Engineer

[View my newsletter](#)

10mo • Edited •

Making LLMs Worth Every Penny: Balancing Costs & Performance 🤖

The quest for high-efficiency, cost-effective NLP solutions is a never-ending balancing act between the desires to deploy both cheap and performant models.

There are several factors that can be adjusted in order to optimize for efficiency. The most important ones being: data quality, data quantity and the model itself. Or put even simpler, the data on one side, the model on the other.

In sectors where data is sparse, labeling thousands of examples isn't just challenging—it's often impossible. And a lot of measures to increase data quality will cost extra.

In traditional ML settings, feature engineering and data augmentation could make all the difference. But with current Large Language Models (LLMs) and other Generative AI solutions, we're dealing with prompts. Prompt engineering is slowly becoming the new feature engineering.

Another big factor is model selection - doesn't matter if we're talking about APIs or custom models - the cost can vary dramatically. And let me tell you one thing: the relationship between cost and performance is anything but linear.

Today's paper is focusing on exactly this topic. In the setting of intent detection, the authors are evaluating how much bang they got for their bucks.

Ranging from traditional supervised fine-tuning (SFT) to GPT-4 with Retrieval-augmented Generation (RAG), they analyzed the performance-cost curve for a wide variety of workflows.

What's your experience? Are you still using SFT methods because they're cheaper and - given the data - more powerful? Or did you fall in love with the flexibility of few-shot learning and RAG? Let us know.

[arXiv] <https://lnkd.in/dGMQzzY4>

↓

Liked this post? Get weekly AI highlights and papers-of-the-week directly to your inbox 📧 [llmwatch.com](#)

For relatively "easy" tasks, GPT-4 may often not be worth it. RAG can level the playing field even further.

Model	Setting	Micro-F1↑	Cost↓
GPT-4	1-shot	<b>80.4</b>	<b>620\$</b>
GPT-3.5	1-shot	75.2	31\$
Anthropic Claude 2	1-shot	<b>76.8</b>	<b>15\$</b>
Command-nightly	1-shot	58.4	22\$


## Other organizations employing Dynamic Few-Shot Prompting (1 year later)


Microsoft | Tech Community Community Hubs Blogs Events Microsoft Learn Lounge

Home > Microsoft FastTrack > FastTrack for Azure > Leveraging dynamic few-shot prompt with Azure OpenAI

[Back to Blog](#) [< Newer Article](#) [Older Article >](#)

## Leveraging dynamic few-shot prompt with Azure OpenAI

By  [Franklin Lindemberg Guimaraes](#)

Published Sep 04 2024 02:35 PM  4,507 Views

# Dynamic few-shot prompting

Faster, cheaper, and superior LLM generations via better prompting

The diagram illustrates the Dynamic Few-Shot Prompt Architecture. It features a central 'Orchestrator' (represented by a Python logo) that interacts with a user (represented by a person icon) and a 'Vector Store' (represented by a document icon). The process follows four steps: 1. The user sends a 'request/response' to the Orchestrator. 2. The Orchestrator sends the request to an 'Embedding Model' (labeled 'Azure OpenAI'). 3. The Embedding Model finds 'relevant examples' in the 'Vector Store'. 4. The Embedding Model sends the 'send prompt with relevant examples' back to the Orchestrator, which then generates the final response. The entire system is labeled 'Dynamic Few-Shot Prompt Architecture'.

Dynamic Few-Shot Prompt Architecture

Sahar Mor

python.langchain.com/v0.1/docs/use\_cases/sql/agents/#using-a-dynamic-few-shot-prompt

LangChain

ComponentsIntegrationsGuidesAPI ReferenceMore

Get started

Quickstart

Installation

Use cases

Q&A with RAG

Extracting structured output

Chatbots

Tool use and agents

Query analysis

Q&A over SQL+ CSV

Quickstart

Agents

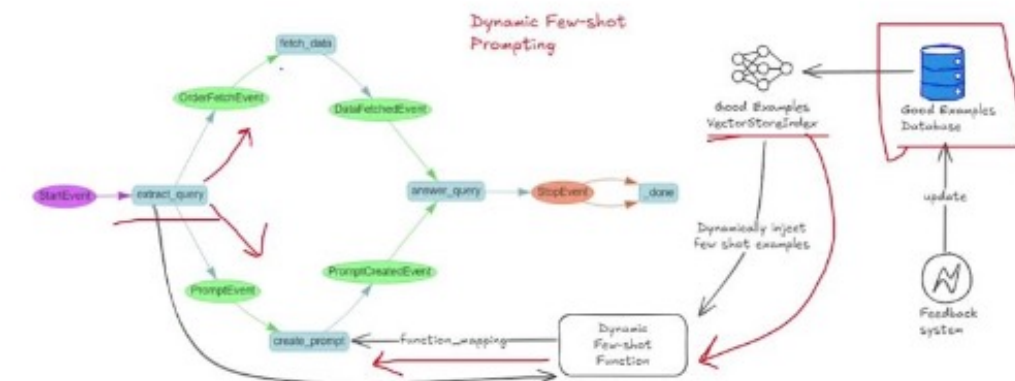
Using a dynamic few-shot prompt

To optimize agent performance, we can provide a custom prompt with domain-specific knowledge. In this case we'll create a few shot prompt with an example selector, that will dynamically build the few shot prompt based on the user input. This will help the model make better queries by inserting relevant queries in the prompt that the model can use as reference.

First we need some user input \<> SQL query examples:

```
examples = [
    {"input": "List all artists.", "query": "SELECT * FROM Artist;"},
    {
        "input": "Find all albums for the artist 'AC/DC'.",
        "query": "SELECT * FROM Album WHERE ArtistId = (SELECT ArtistId FROM Artist WHERE Name = 'AC/DC');"
    },
    ...
]
```

A screenshot of a tweet from the account 'LlamaIndex', which has 218,183 followers and posted 2 weeks ago. The tweet text reads: 'Instead of finetuning your LLMs, try dynamic few-shot prompting instead'. The phrase 'dynamic few-shot prompting' is highlighted in blue. Below the text is a lightbulb icon. A blue retweet button with a white retweet symbol is visible. The tweet is part of a thread, indicated by a '1 of 2' icon. The bottom of the tweet shows the start of another line of text: 'With dynamic few-shot prompting, instead of injecting a fixed set of examples into the prompt, you retrieve a dynamic set of examples based on the query - so you find relevant examples that are relevant towards solving your input task.' Below this is a line of text: 'This is helpful for use cases like customer support, text-to-SQL, structured output, and more.' At the bottom, there is a line of text: 'RS Rohan has a great resource repo showing how this works using LlamaIndex workflows, check it out: https://lnkd.in/g58qxa8X'. The very bottom of the image shows the start of another line of text: 'For more details on workflows: https://lnkd.in/giseEZ5q'.





# Summary

- **BERT-based models** (with SetFit) can be a **strong alternative to expensive LLM APIs**, as long as you can find 3-5 samples per class
  - More studies in more datasets are validating our findings! (e.g. <https://huggingface.co/blog/setfit-absa>)
- **Human-curated samples** give an **easy performance boost** vs. random samples - worth investing!
- **Dynamic Few-Shot Prompting** helps in results, and **dramatically** even more in **reducing LLM operating expenses**
  - More researchers validating our findings (e.g. [Bahdanau paper, 2023](#))
- LLM frameworks like **LangChain** and **LlamaIndex** provide Dynamic Few-Shot prompting out of the box!
- Read more at helvia.ai Labs blogpost:
  - <https://helvia.ai/labs/making-llms-worth-every-penny-resource-limited-text-classification-in-banking/>



1. “Breaking the Bank with ChatGPT: Few-Shot Text Classification for Finance”. L. Loukas, I. Stogiannidis, P. Malakasiotis, S. Vassos. **FinNLP @ IJCAI 2023** (short early version)
2. “Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking”. L. Loukas, I. Stogiannidis, O. Diamantopoulos, P. Malakasiotis, S. Vassos. **(ACM ICAIF 2023)** - **also selected by ACM for ACM's Research Highlights**

	CITED BY	YEAR
Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking L Loukas, I Stogiannidis, O Diamantopoulos, P Malakasiotis, S Vassos ACM ICAIF 2023 - Proceedings of the Fourth ACM International Conference on ...	65	2023
Breaking the Bank with ChatGPT: Few-shot Text Classification for Finance L Loukas, I Stogiannidis, P Malakasiotis, S Vassos Proceedings of the Fifth Workshop on Financial Technology and Natural ...	53	2023





# Rest Publications

1. “DlCoE@FinSim-3: Financial Hypernym Detection using Augmented Terms and Distance-based Features”. **L. Loukas**, K. Bougiatiotis, M. Fergadiotis, D. Mavroeidis. **(FinNLP @ IJCAI 2021)**

- **4th place at shared competition.** Used inference sample augmentation based on external business ontology + OOV embeddings + feature engineering

2. “Financial Misstatement Detection: A Realistic Evaluation” E. Zavitsanos, D. Mavroeidis, K. Bougiatiotis, E. Spyropoulou, L. Loukas, G. Paliouras, Proceedings of the International Conference on AI in Finance **(ACM ICAIF 2021)**

- an early version of **EDGAR-CRAWLER** was used in this work!

3. side quest: Greek NLP Toolkit @ COLING 2025

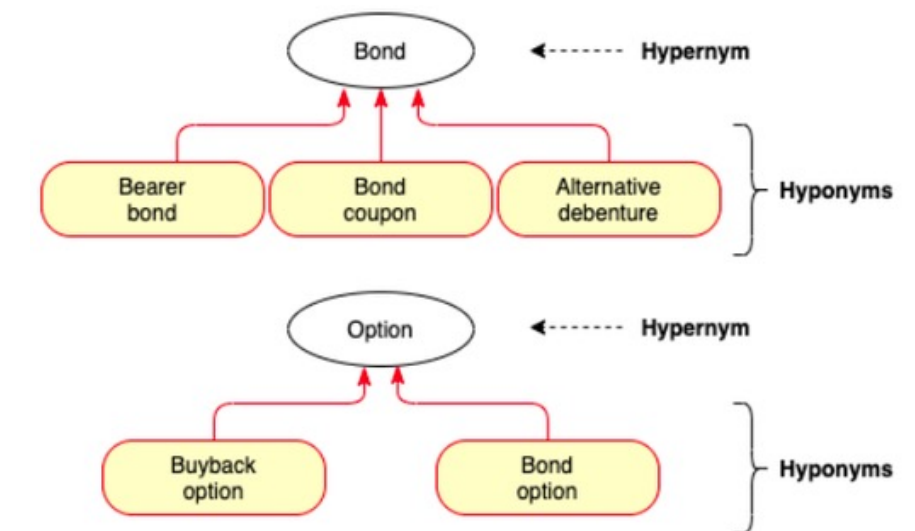
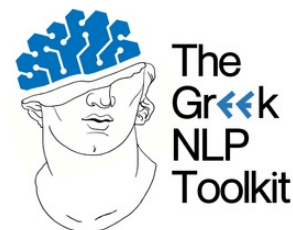


Figure 1: Examples of hypernym relations from the FIBO ontology. Interestingly, “Bond coupon” is a kind of “Bond”, but “Bond option” is a kind of “Option”.

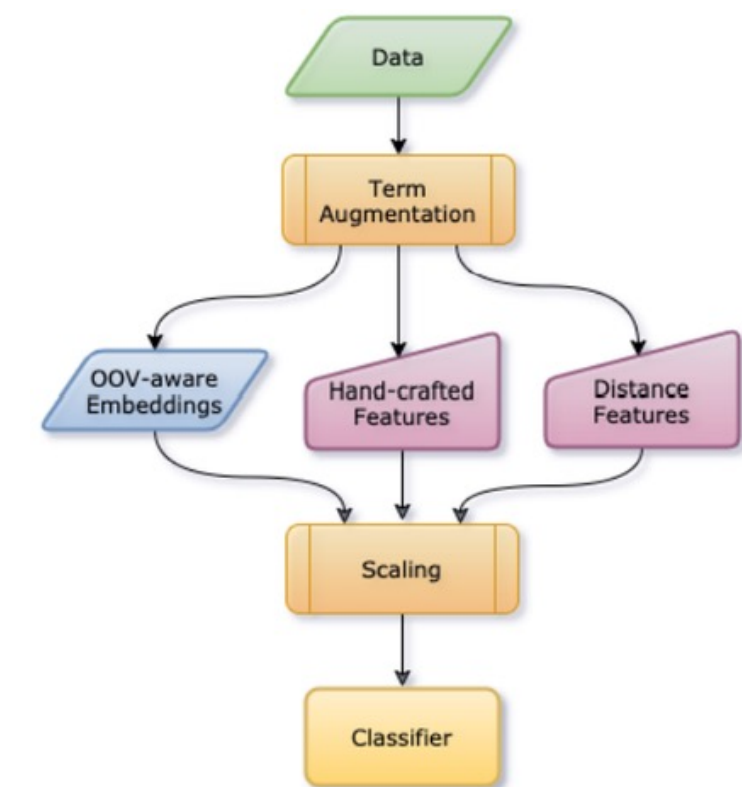


Figure 2: The pipeline of our best system.



# Contributions

- **Papers**
  - **4 conference papers** at **ACL/WWW/ACM ICAIF** venues
  - **3 workshop papers** at **ACL/IJCAI** venues
  - **301 citations overall**
- **Patents**
  - **1 granted US patent** / 2 patent applications to EU/WPTO patent offices
    - assigned to **Ernst & Young** and **NCSR Demokritos** for commercial license
- **Software**
  - 1 open-source software (OSS) with **420+ stars** on Github (EDGAR-CRAWLER)
- **Data**
  - 1 corpus (EDGAR-CORPUS, downloaded 4K times)
  - 2 annotated datasets (FiNER-139 & Banking77 expert-curated samples)
- **Models**
  - 3 open-access BERT models (SEC-BERT, downloaded 7K times)
  - 1 embedding model with SOTA results (EDGAR-W2V)





**Thank you!**