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 @ EMNLP 2021 Online & in the Dominican Republic

EDGAR-CRAWLER

edgar-crawler
Turn unstructured financial documents into clean JSON files.

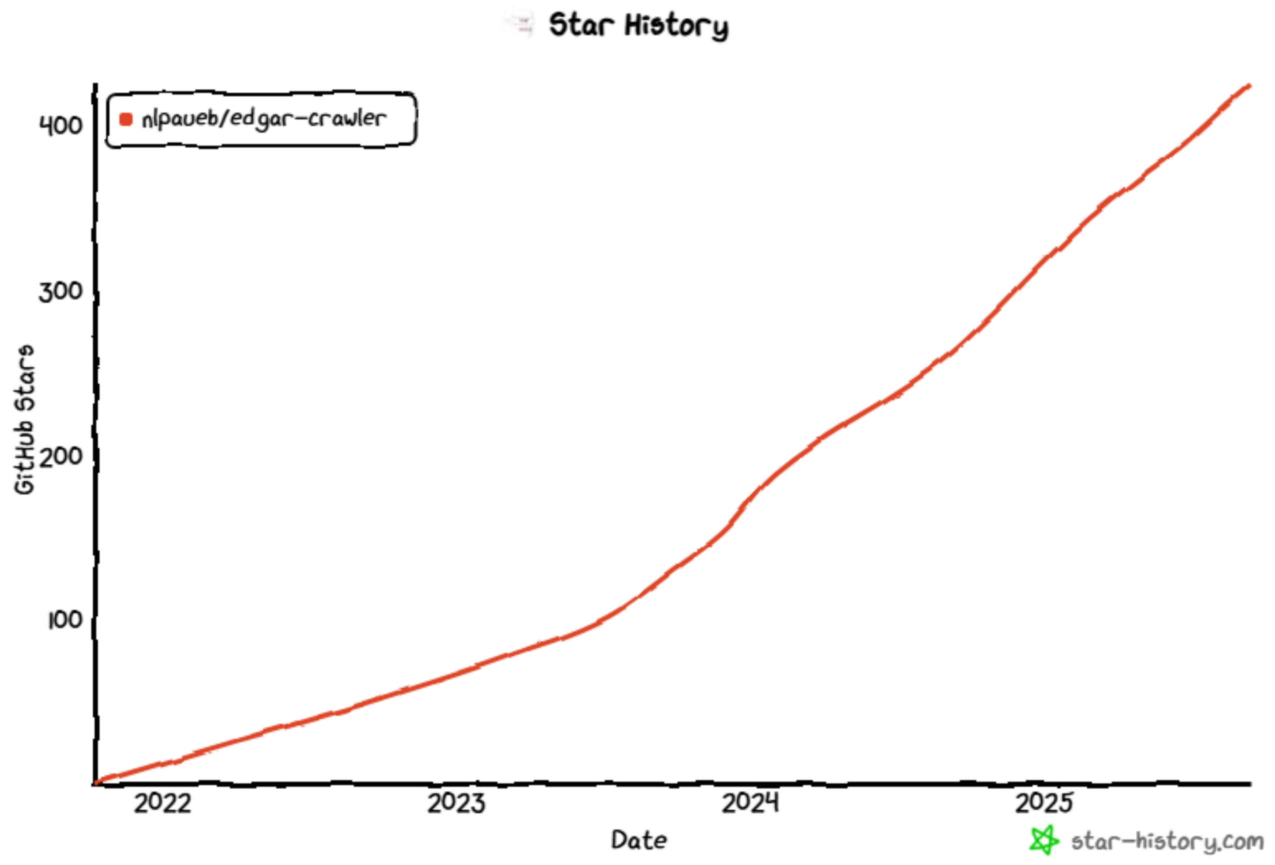
- The go-to NLP toolkit for business/financial data/text preprocessing from the SEC
- 420+ stars on Github — Fork 110 | Star 426
- Details:
  - **Converts unstructured documents** of 100+ pages to a structured **JSON**
  - Supports **multiple filters** like company, years, stock ticker
  - Supports **multiple filings** (10-K, 10-Q, 8-K)
  - Clean, normalize and **remove tables** with 2 clicks
- Lightning Talk at NLP-OSS @ EMNLP 2023
- Started in 2020, continuing until now
- Earned **grant** from Google Summer of Code
- **Paper at WWW 2025 (A* CORE ranking)** THE WEB CONFERENCE ACM

Hugging Face

4K VIEWS

3K DOWNLOADS

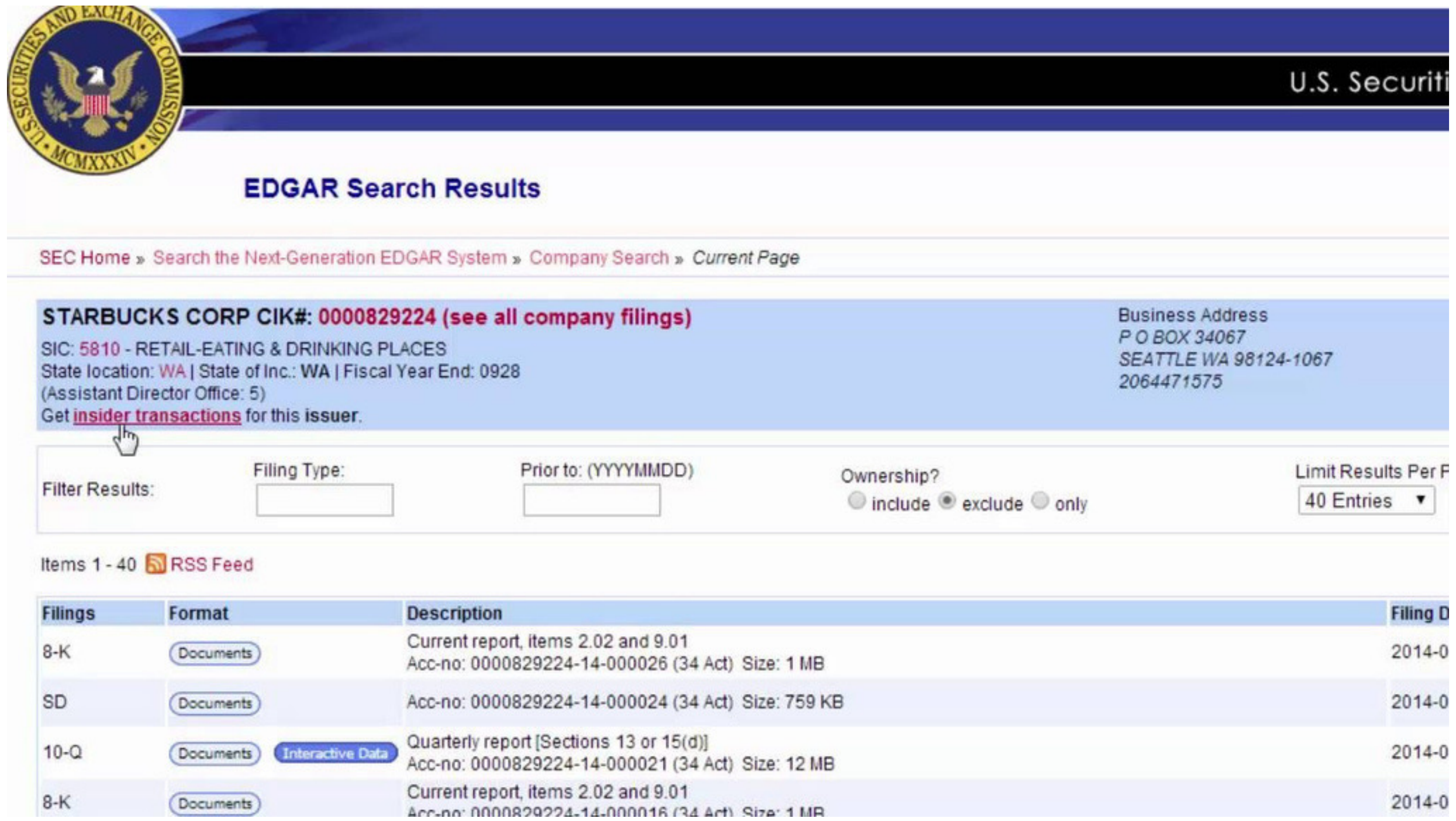
CITED BY	YEAR
50	2021



GitHub

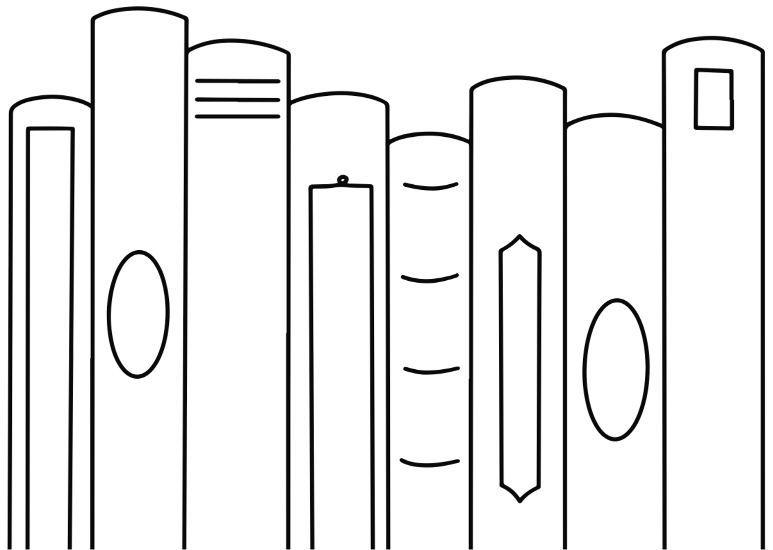
Motivation

- 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)*



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
- <u>Largest financial NLP corpus available up to date!</u>



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; <u>most notable text resource for business/economic NLP</u>
- Total of **6.5B tokens inside!**

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

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

EDGAR documents have <u>hundreds of pages</u>

- An annual report (10-K report) is organized in 4 parts and **20 different items**, with **each item having specific points of interests**
- <u>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</u>
- Extracting specific items from documents with hundreds of pages <u>requires extensive, manual work</u> 😤
- 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
	Item 9A	Controls and Procedures
	Item 9B	Other Information
Part III	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
	Item 14	Principal Accounting Fees and Services
Part IV	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)	**94-2404110** (I.R.S. Employer Identification No.)
1 Infinite Loop **Cupertino, California** (Address of principal executive offices)	**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)

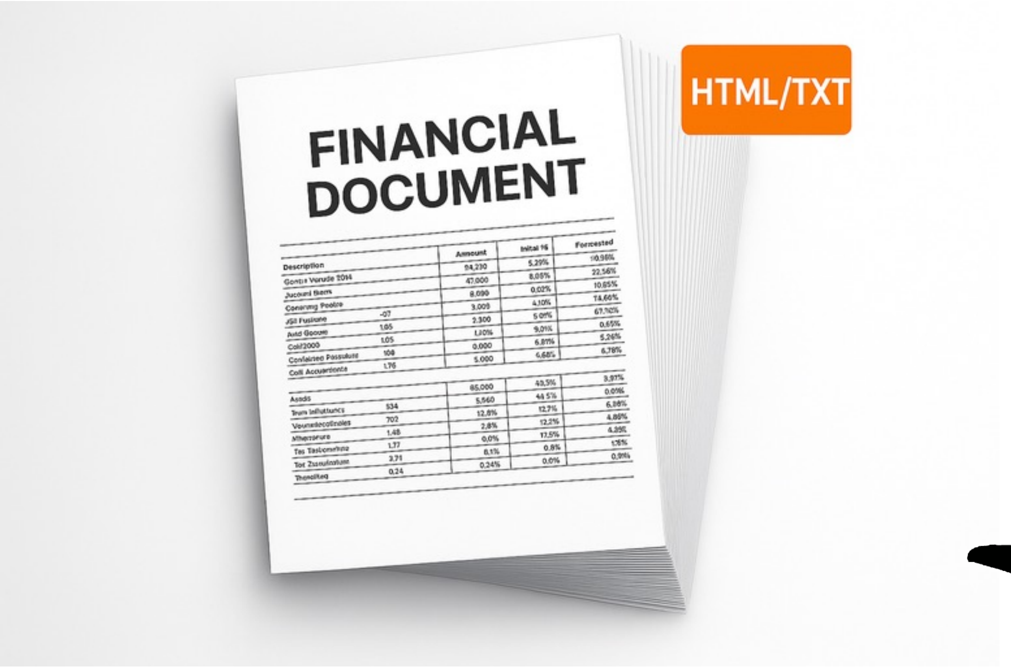
	December 31, 2022	December 31, 2023
ASSETS		
Current assets:		
Cash and cash equivalents	$ 53,888	$ 73,387
Marketable securities	16,138	13,393
Inventories	34,405	33,318
Accounts receivable, net and other	42,360	52,253
Total current assets	146,791	172,351
Property and equipment, net	186,715	204,177
Operating leases	66,123	72,513
Goodwill	20,288	22,789
Other assets	42,758	56,024
Total assets	$ 462,675	$ 527,854
LIABILITIES AND STOCKHOLDERS' EQUITY		
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

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 <u>false positives</u> inside: table of contents, inline references..
    - lots of domain-specific rules to deal with those
- Finally, we have a way to <u>convert unstructured documents with hundreds of pages to some structured JSON files with clean text data</u> 😊
- Annual reports of every public US company + more than 20 years of data => we have **EDGAR-CORPUS!**



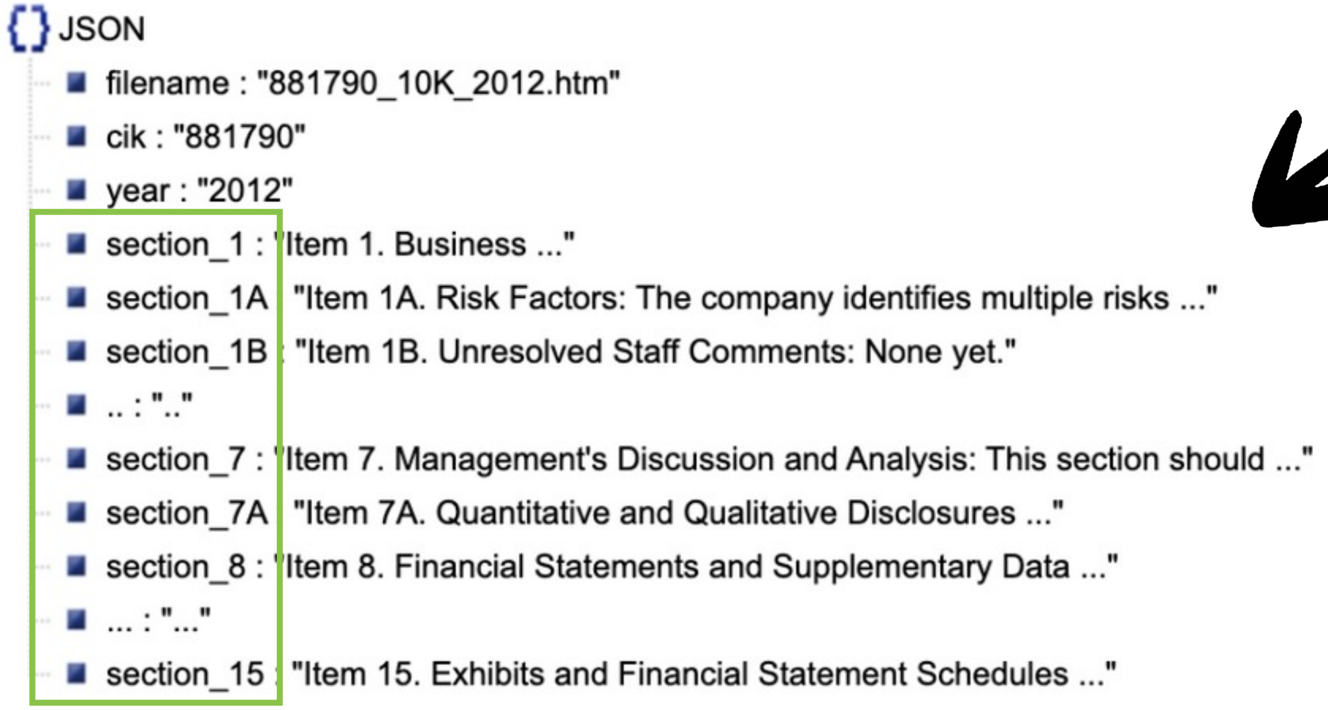


Figure 1: An example of a 10-K report in JSON format as downloaded and extracted by EDGAR-CRAWLER.

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

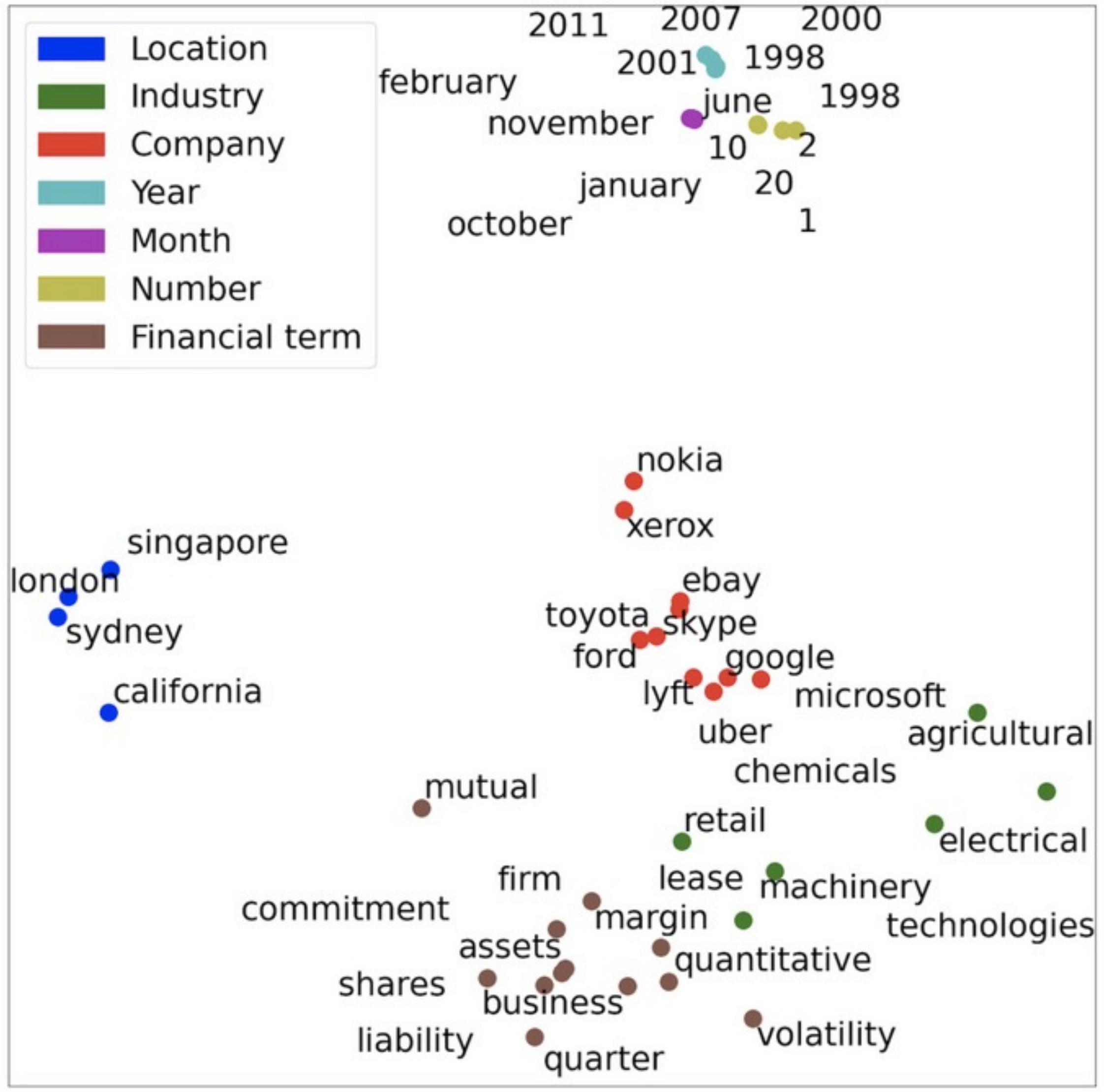


Figure 2: Visualization of the **EDGAR-W2V** embeddings. Different colors indicate different entity types.

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. ↑	Rank ↓	F1 ↑	MSE ↓	R^2 ↑
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)	**87.9**	**1.21**	**77.3**	**0.141**	**0.176**
jina-embeddings-v4	83.9	1.32	72.3	**0.110**	**0.302**

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/nlpaueb/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.

Find all relevant links here!

Chapter #1

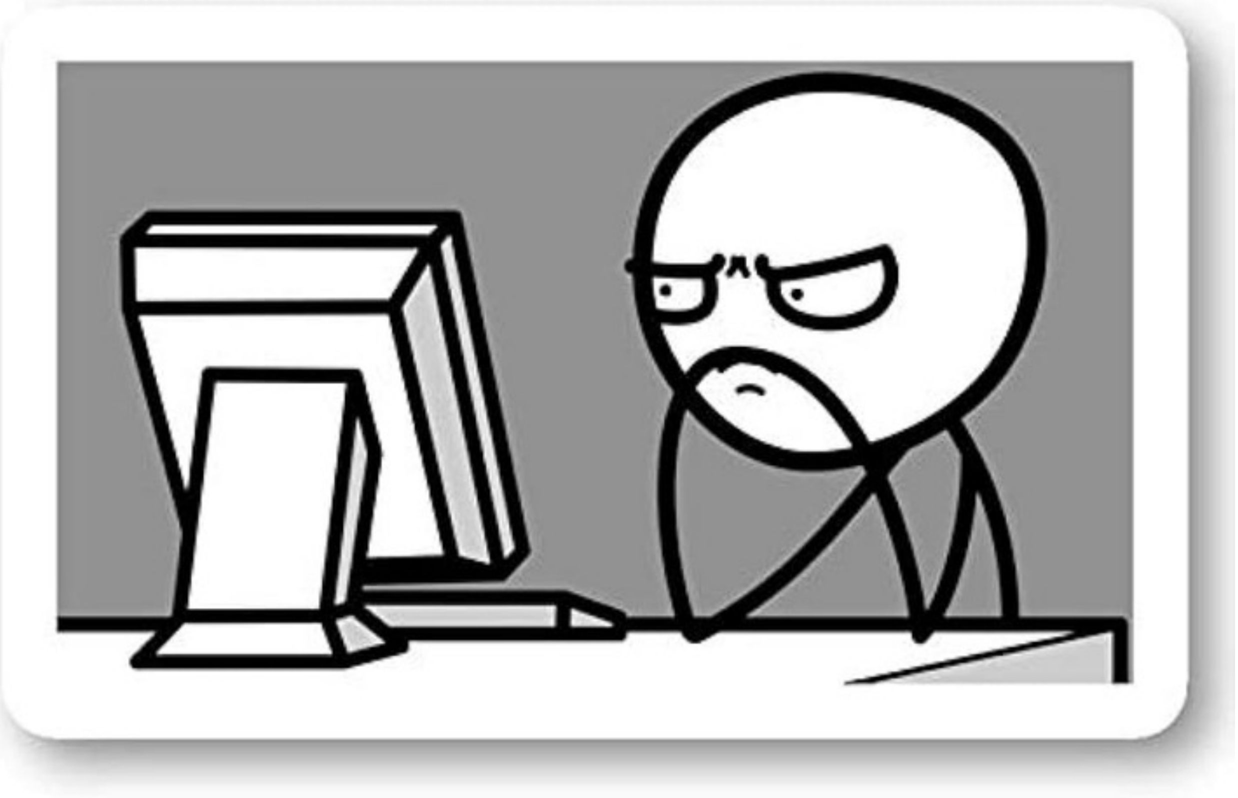
Research Question #1: "How can we use open-access documents for financial NLP?"

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 :)

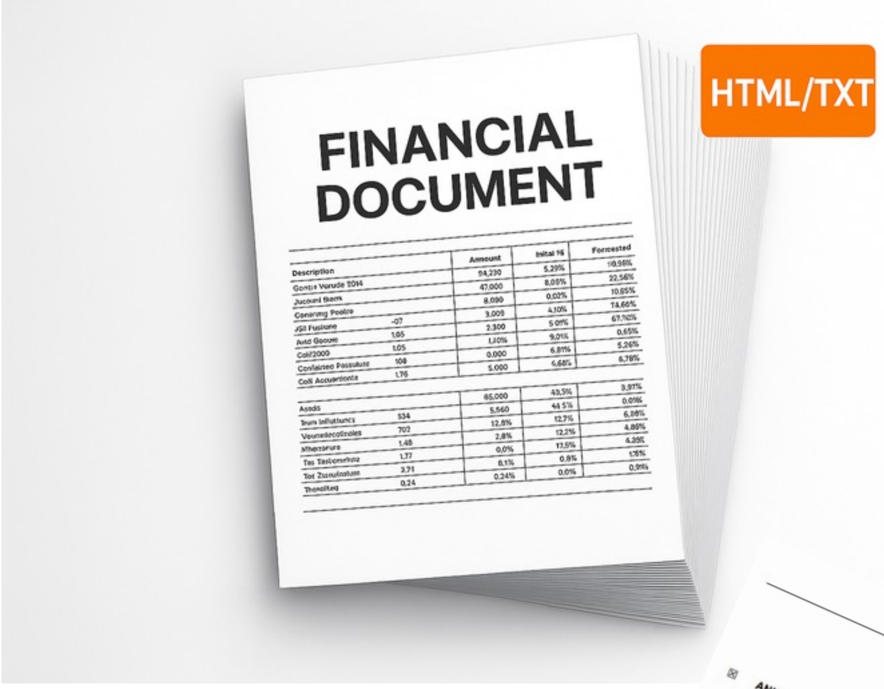
Support more types of documents like those for insider trading.

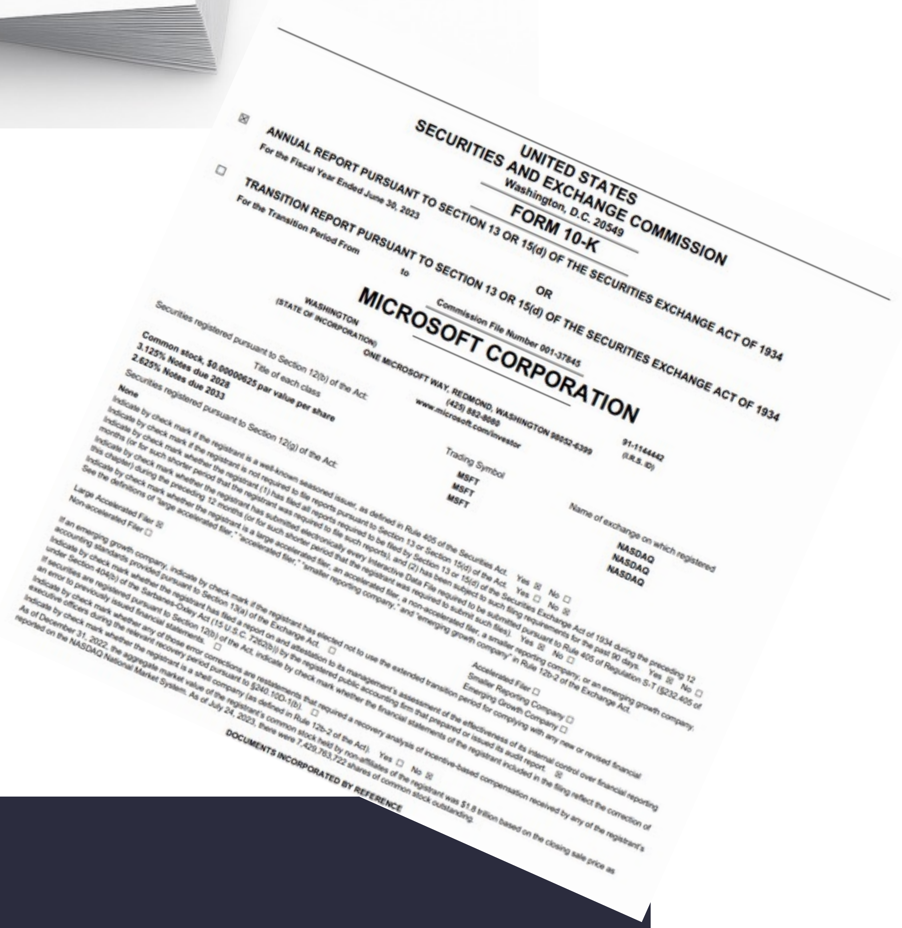
Create a GUI for more user-friendly configuration.

edgar-crawler

Turn unstructured financial documents into clean JSON files.

Hundreds of pages from unstructured company filings?





Structured JSON to bootstrap your research!

edgar-crawler

Turn unstructured financial documents into clean JSON files.

Fork 110 Star 426


```
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: ..."
```

REMINDER

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

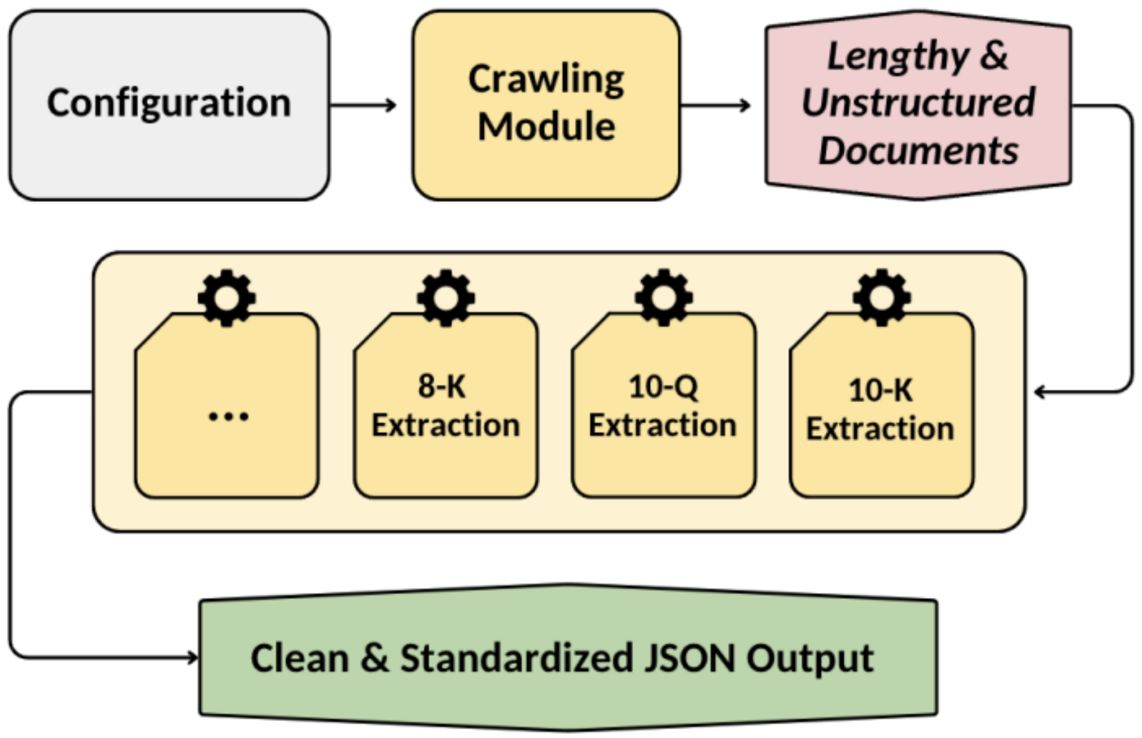


Figure 2.4: EDGAR-CRAWLER’s architecture. First, the user specifies (Configuration) what data they want (companies, years, filing types) and the software downloads them. Then, the item extraction pipeline extracts the item-specific sections and converts them to a standardized JSON format.

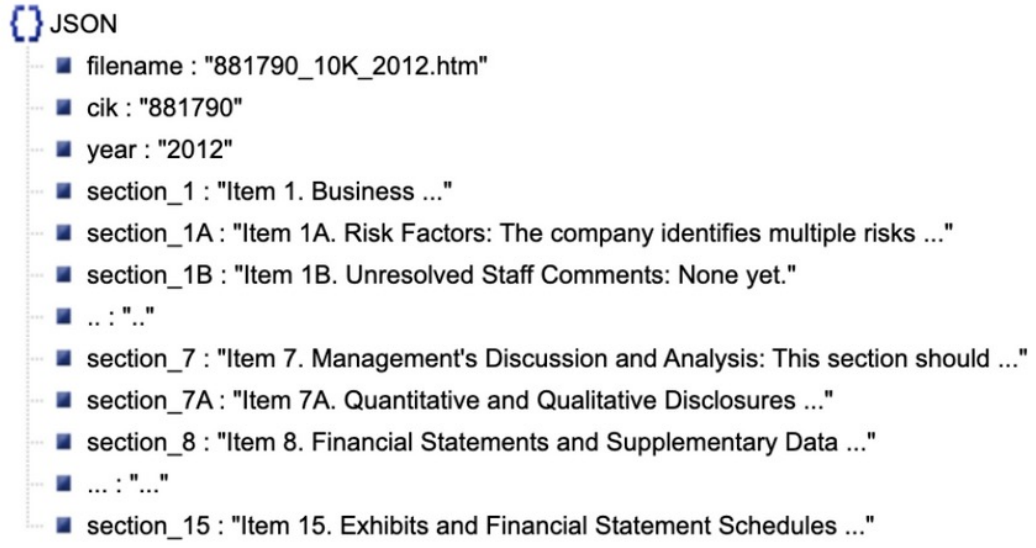


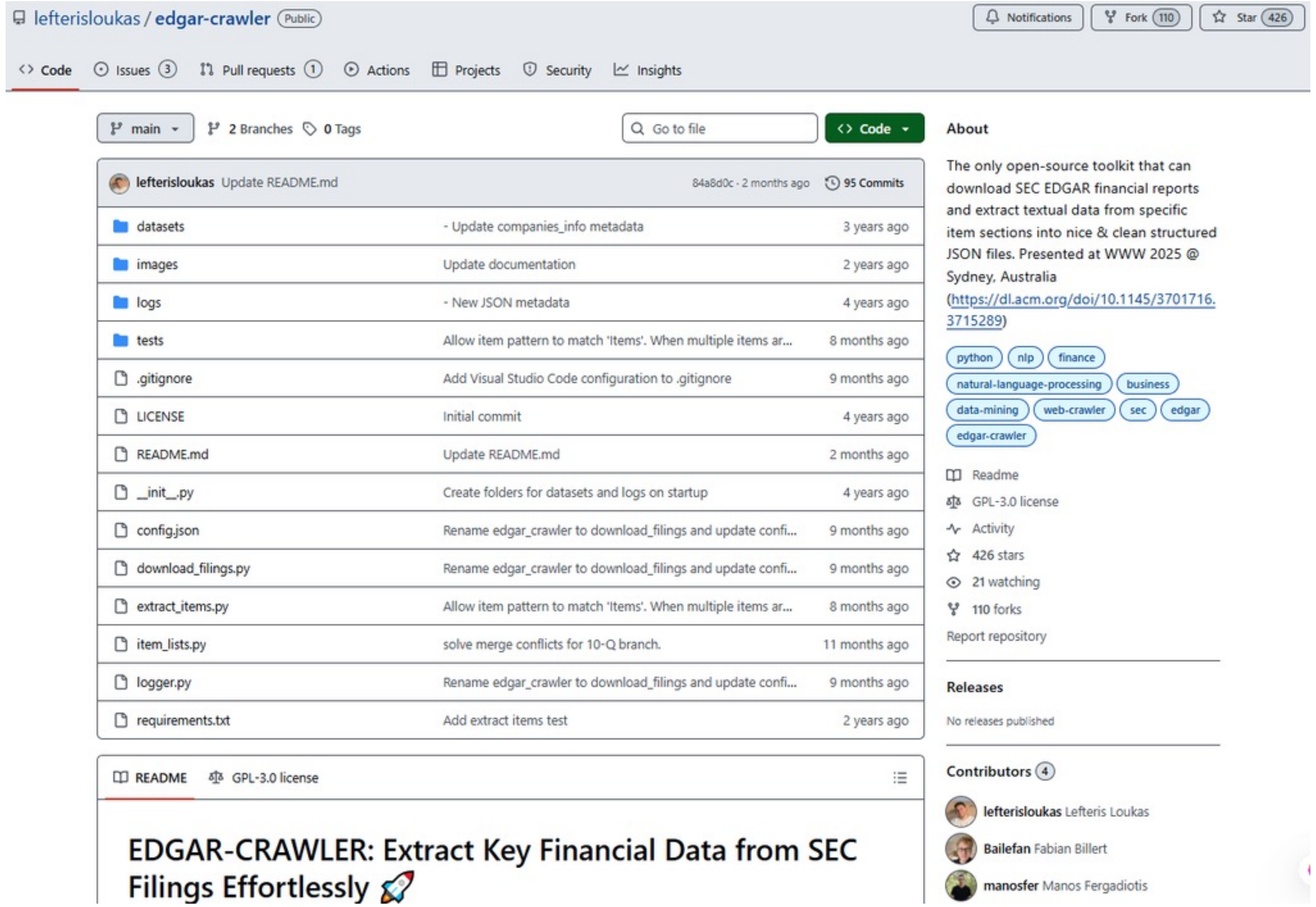
Figure 1: An example of a 10-K report in JSON format as downloaded and extracted by EDGAR-CRAWLER.

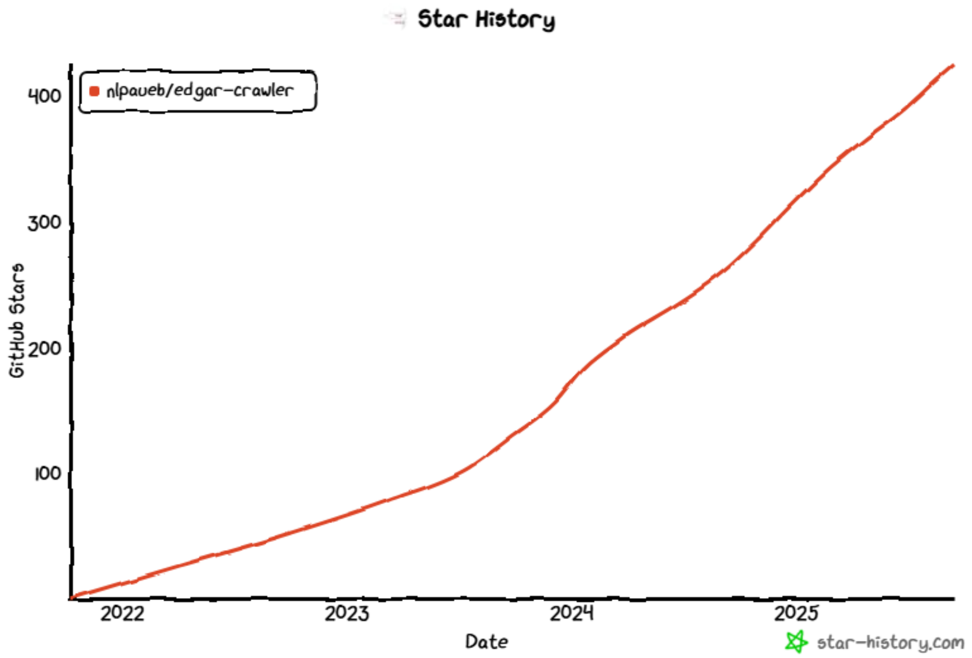
Nop, LLMs can not process SEC documents easily

- 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

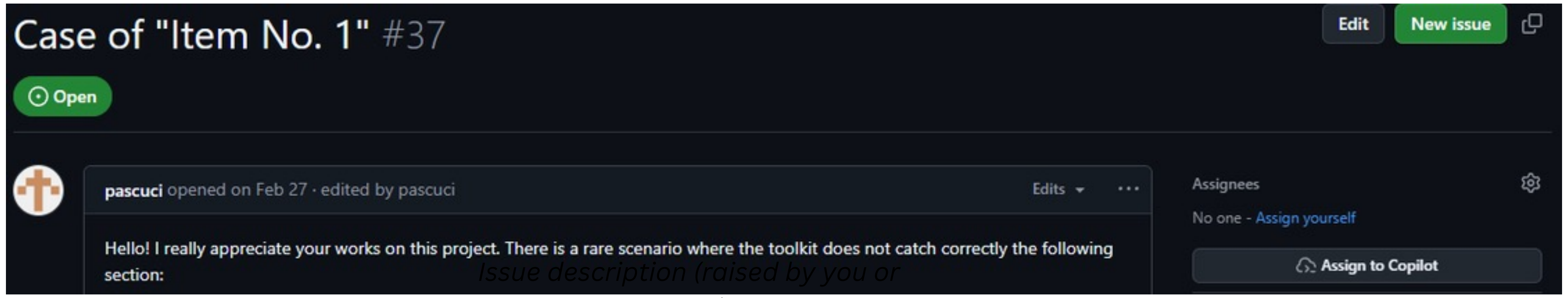
GitHub

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

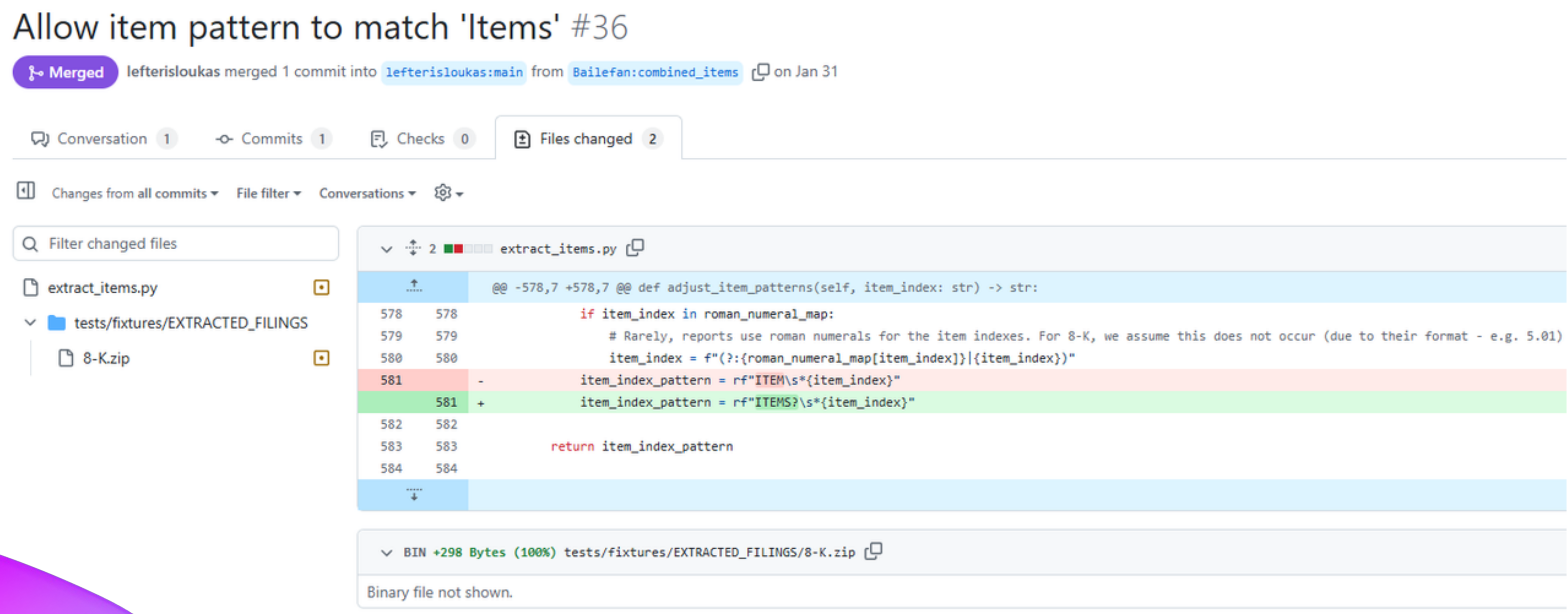




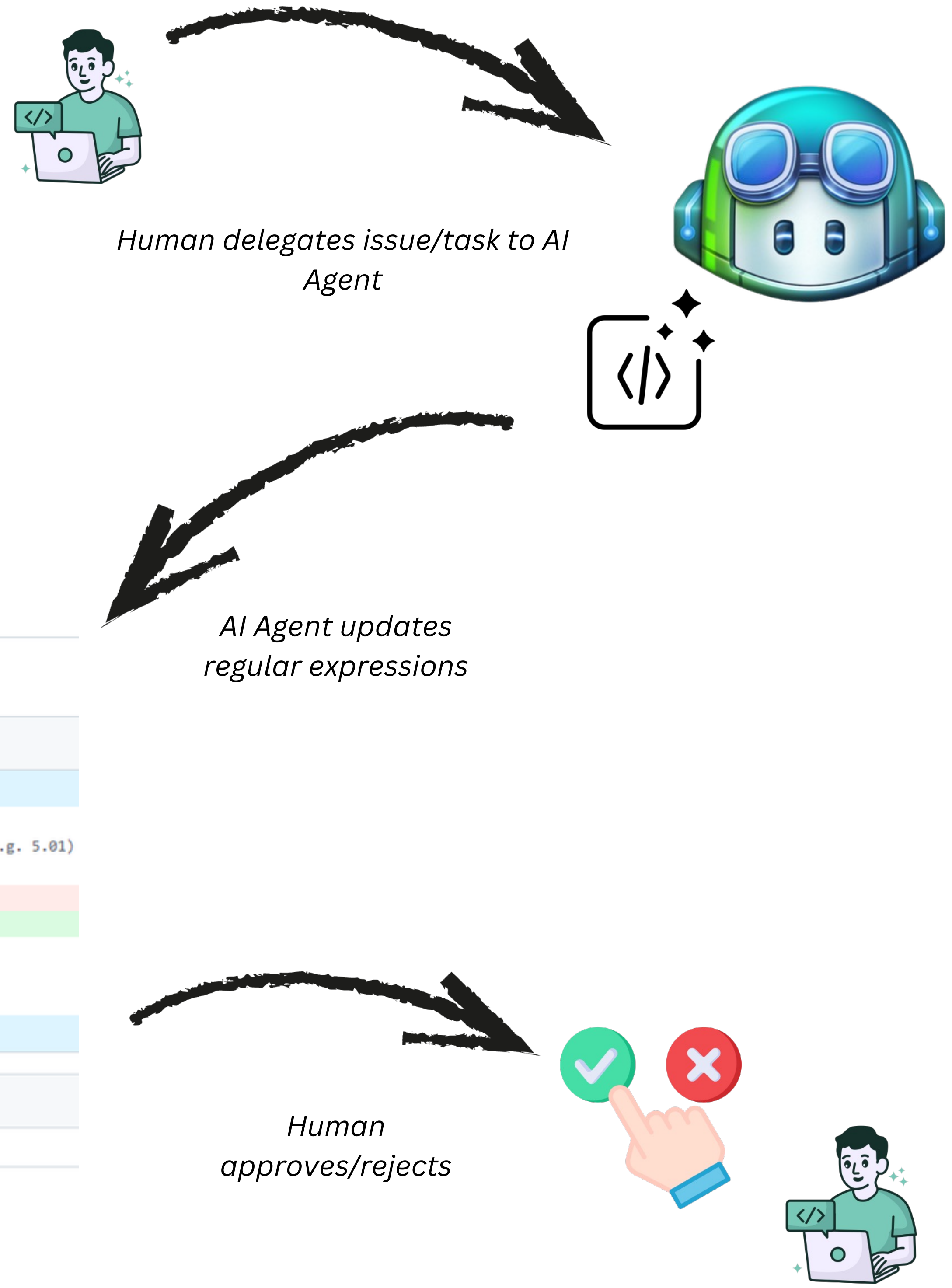
Updating the software’s rules is easy when you use AI (Coding) Agents



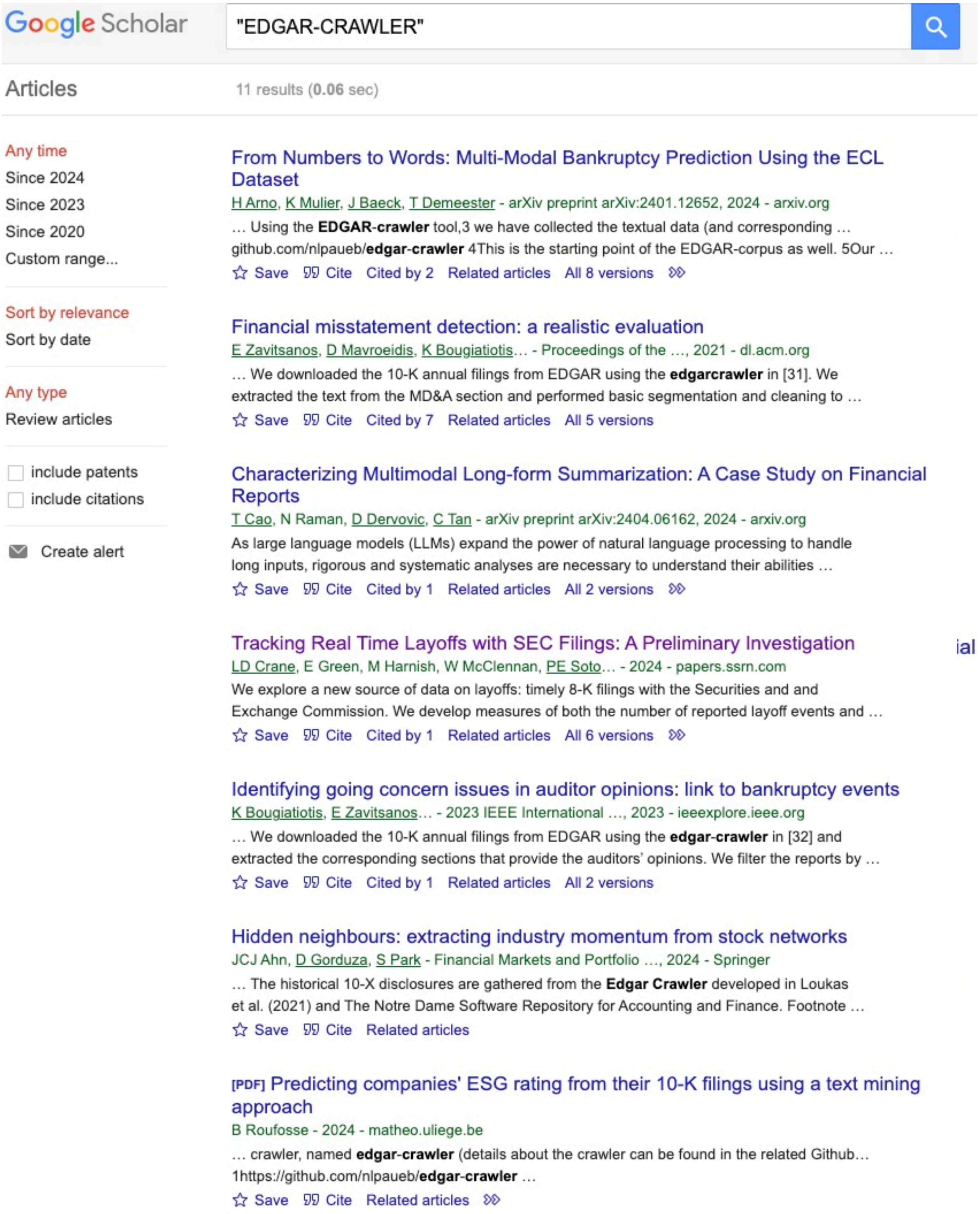
Issue description (raised by you or users)



Code changes implemented by Copilot / Agent



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



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 @ EMNLP 2021 Online & in the Dominican Republic

EDGAR-CRAWLER

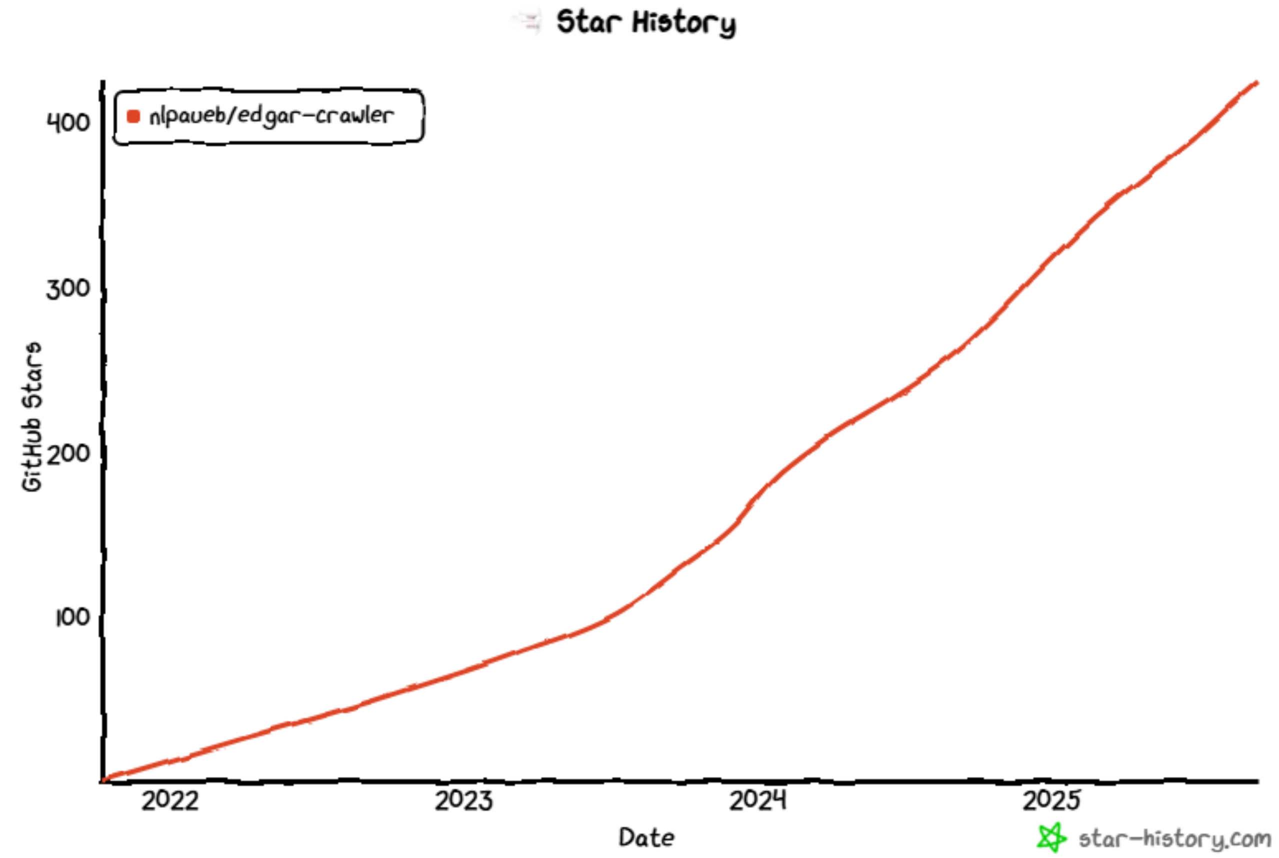
edgar-crawler
Turn unstructured financial documents into clean JSON files.

- The go-to NLP toolkit for business/financial data/text preprocessing from the SEC
- 420+ stars on Github   Fork 110   Star 426
- 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 @ EMNLP 2023
- Started in 2020, continuing until now
- Earned **grant** from Google Summer of Code
- **Paper at WWW 2025 (A* CORE ranking)** THE WEB CONFERENCE ACM

Hugging Face

4K VIEWS   3K DOWNLOADS

CITED BY	YEAR
50	2021



GitHub

Chapter #2

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

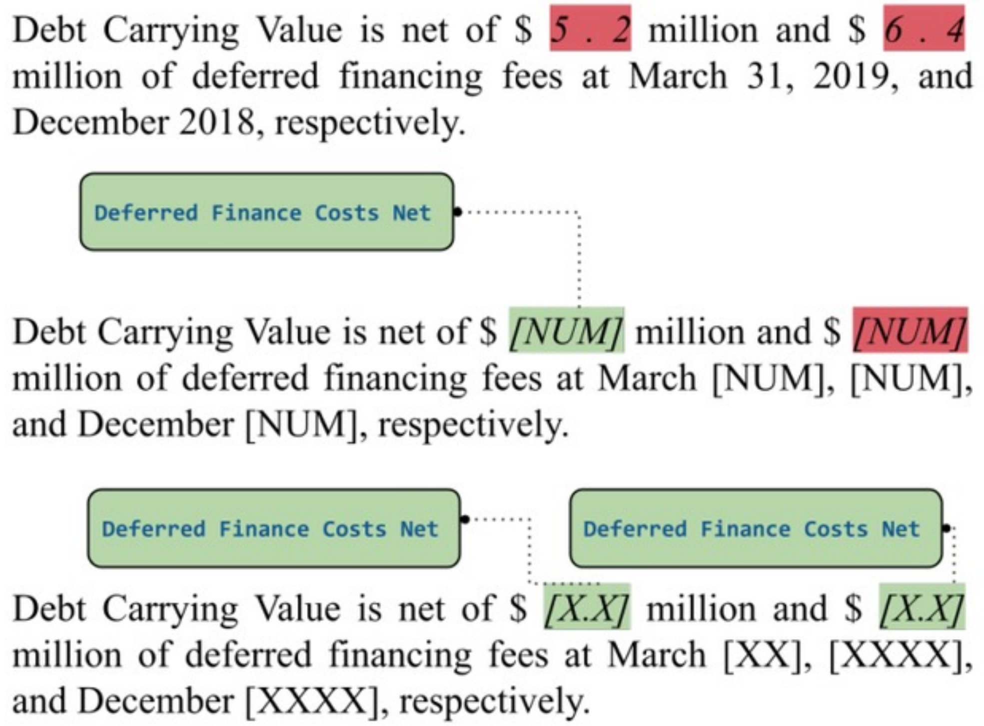
Numeric Entity Recognition for XBRL Tagging

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

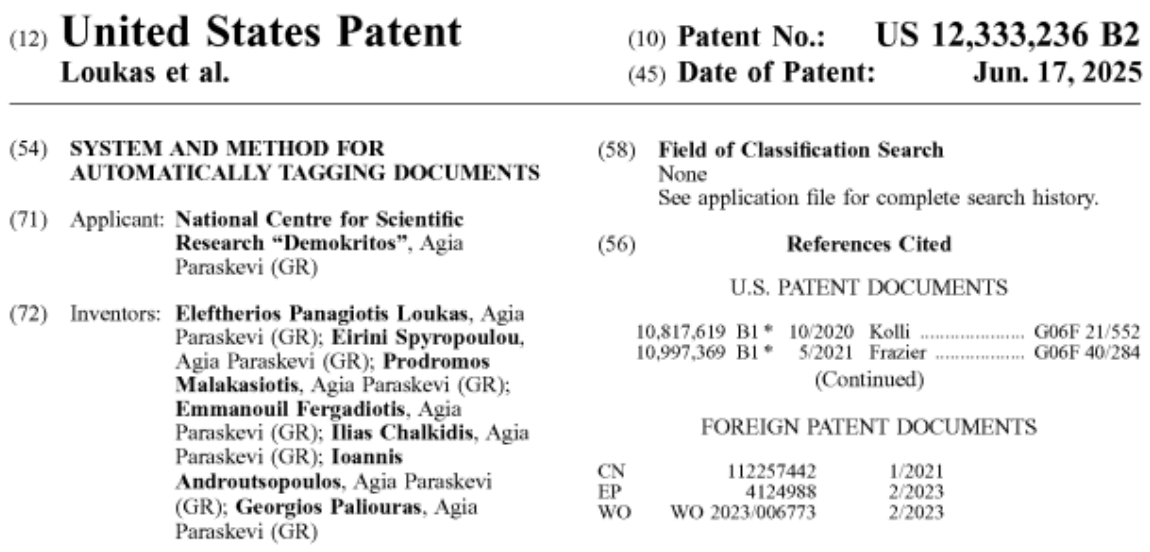
ACL 2022
22ND – 27TH MAY | 60TH MEETING | DUBLIN

CITED BY	YEAR
98	2022



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**


(12) **United States Patent**
Loukas et al.

(10) **Patent No.:** **US 12,333,236 B2**
(45) **Date of Patent:** **Jun. 17, 2025**

(54) **SYSTEM AND METHOD FOR AUTOMATICALLY TAGGING DOCUMENTS**

(71) Applicant: **National Centre for Scientific Research "Demokritos"**, Agia Paraskevi (GR)

(72) Inventors: **Eleftherios Panagiotis Loukas**, Agia Paraskevi (GR); **Eirini Spyropoulou**, Agia Paraskevi (GR); **Prodromos Malakasiotis**, Agia Paraskevi (GR); **Emmanouil Fergadiotis**, Agia Paraskevi (GR); **Ilias Chalkidis**, Agia Paraskevi (GR); **Ioannis Androutsopoulos**, Agia Paraskevi (GR); **Georgios Paliouras**, Agia Paraskevi (GR)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

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

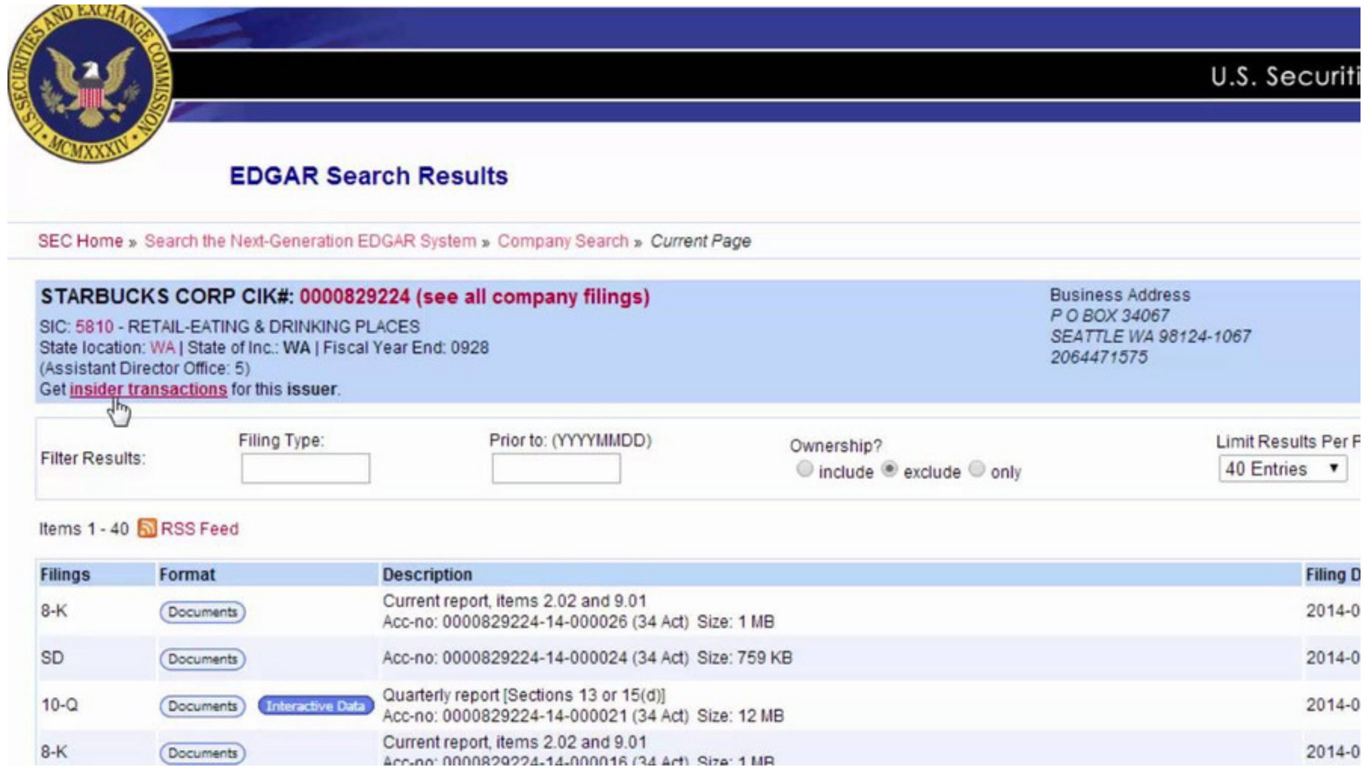
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

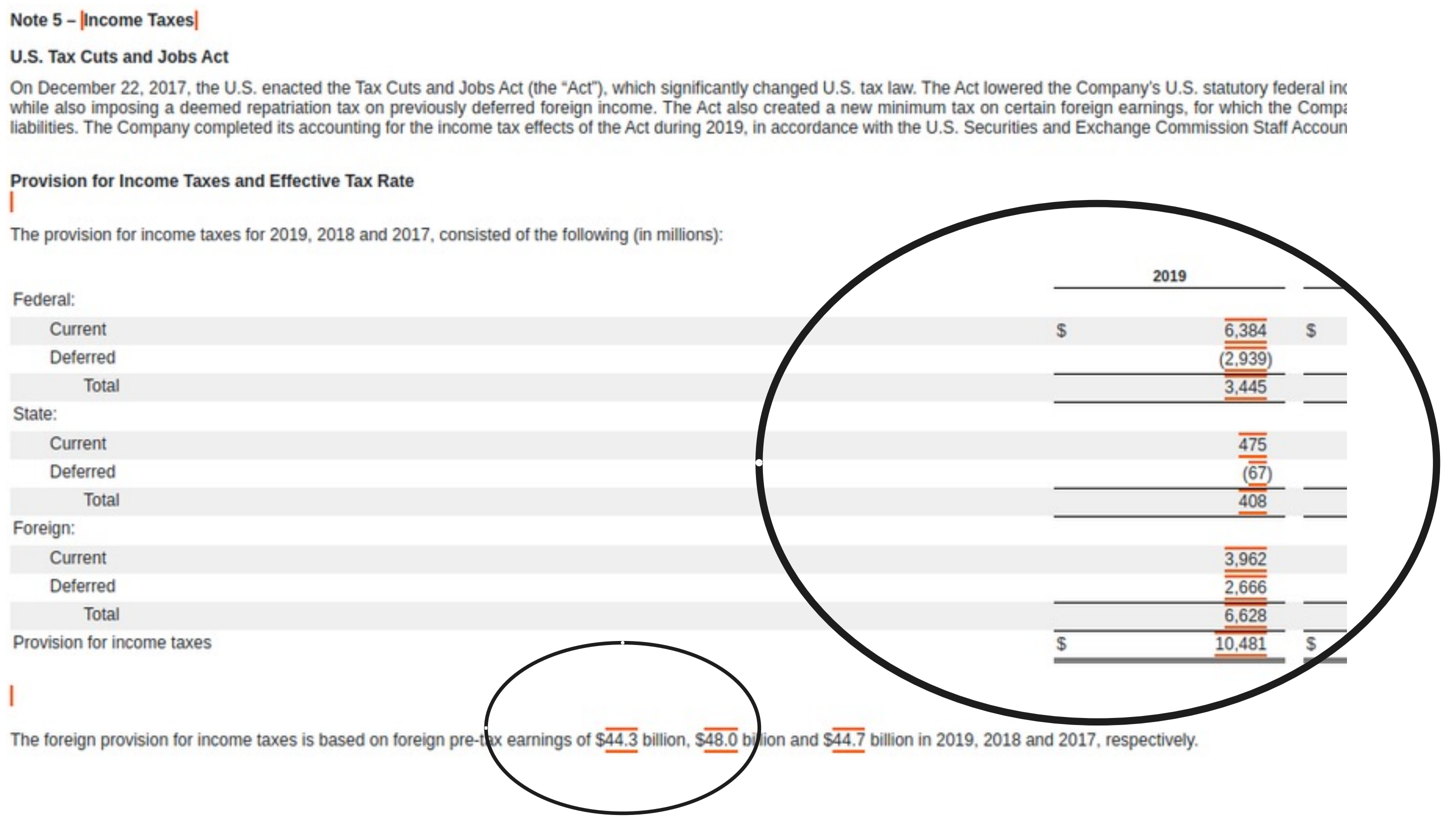
ΔΗΜΟΚΡΙΤΟΣ
DEMOKRITOS

EY

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





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

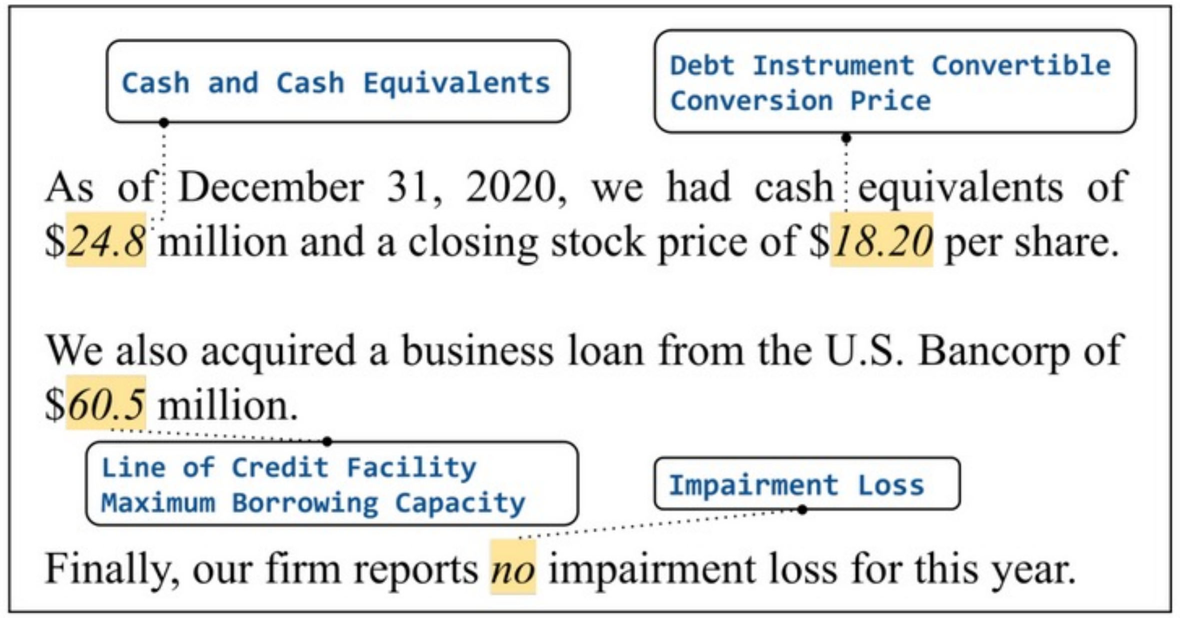
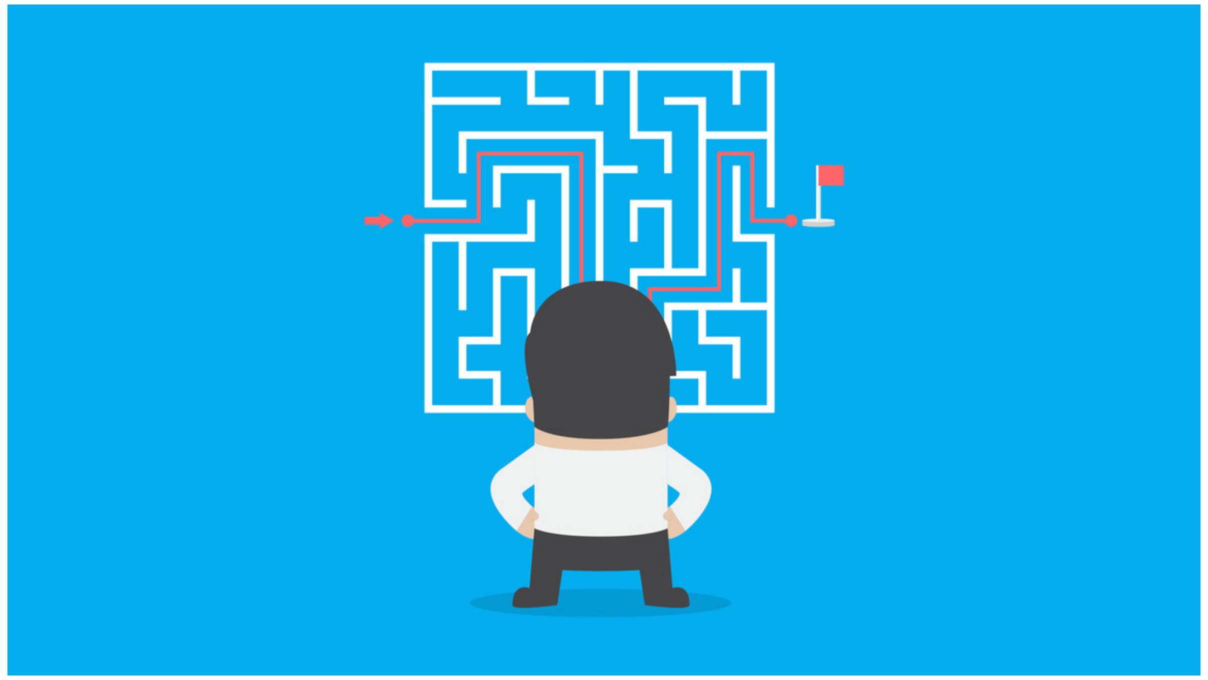


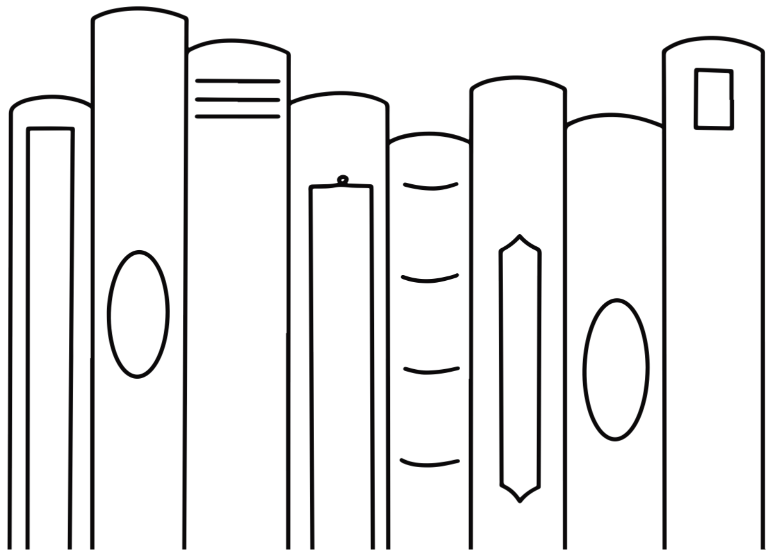
Figure 1: Sentences from **FinER**-139, with **XBRL** tags on numeric and non-numeric tokens. **XBRL** tags are actually **XML**-based and most tagged tokens are numeric.



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



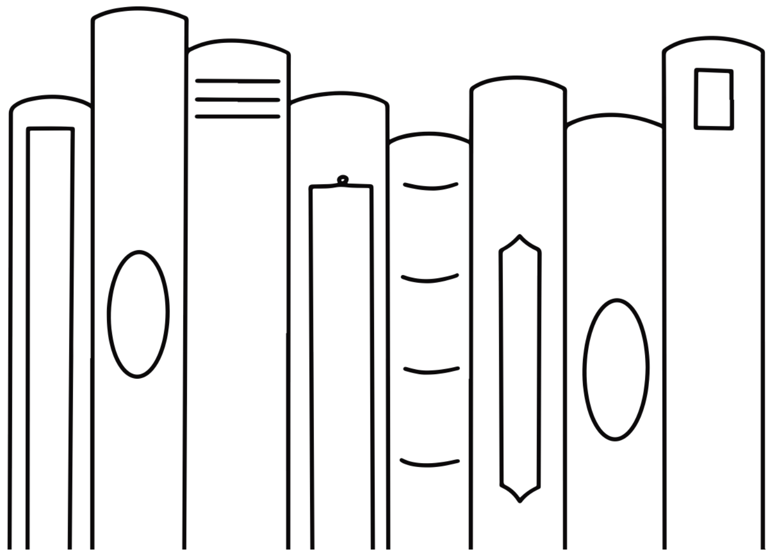
Dataset	Domain	Entity Types
CONLL-2003	Generic	4
ONTONOTES-V5	Generic	18
ACE-2005	Generic	7
GENIA	Biomedical	36
Chalkidis et al. (2019)	Legal	14
Francis et al. (2019)	Financial	9
FiNER-139 (ours)	Financial	**139**

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 ± 33.9	1.8 ± 1.2
Dev	112,494	45.4 ± 35.9	1.7 ± 1.2
Test	108,378	46.5 ± 38.9	1.7 ± 1.1

Table 2: FiNER-139 statistics, using SPACY's tokenizer and the 139 tags of this work (± standard deviation).

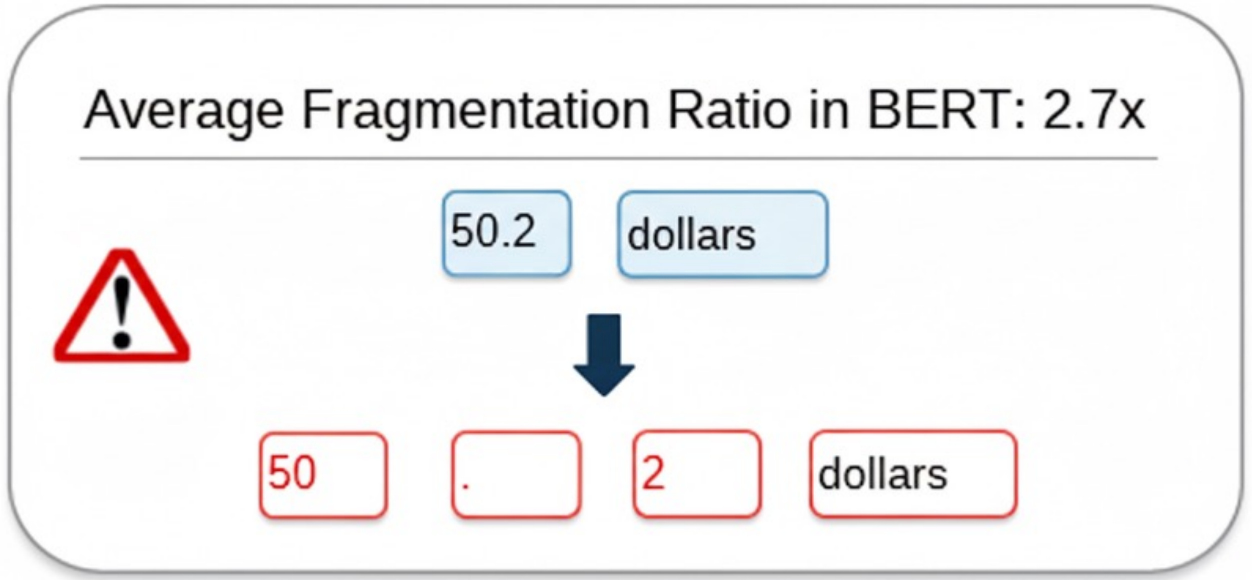
FiNER-139 characteristics

- 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**



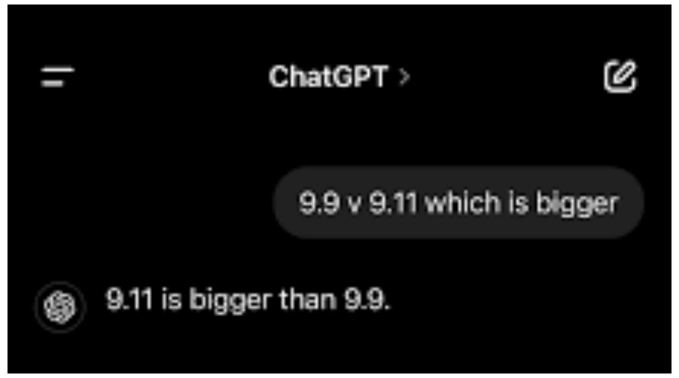
Baseline methods	μ-F_1	m-F_1
SPACY (words)	48.6 ± 0.4	37.6 ± 0.2
BILSTM (words)	77.3 ± 0.6	73.8 ± 1.8
BILSTM (subwords)	71.3 ± 0.2	68.6 ± 0.2
BERT (subwords)	75.1 ± 1.1	72.6 ± 1.4
BILSTM (words) + CRF	69.4 ± 1.2	67.3 ± 1.6
BILSTM (subwords) + CRF	76.2 ± 0.2	73.4 ± 0.3
BERT (subwords) + CRF	**78.0** ± 0.5	**75.2** ± 0.6

Table 3: Entity-level μ-F_1 and m-F_1 (%, avg. of 3 runs with different random seeds, ± 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 LOLLAPALOOZA. 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.

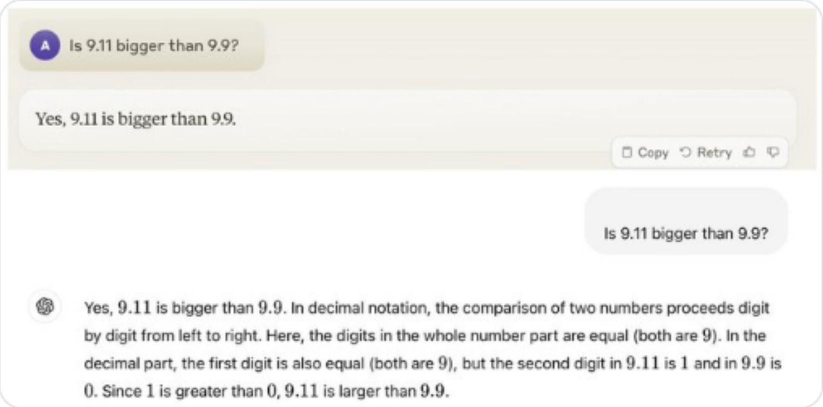
7:25 AM · Feb 18, 2025 · **3.6M** Views

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...



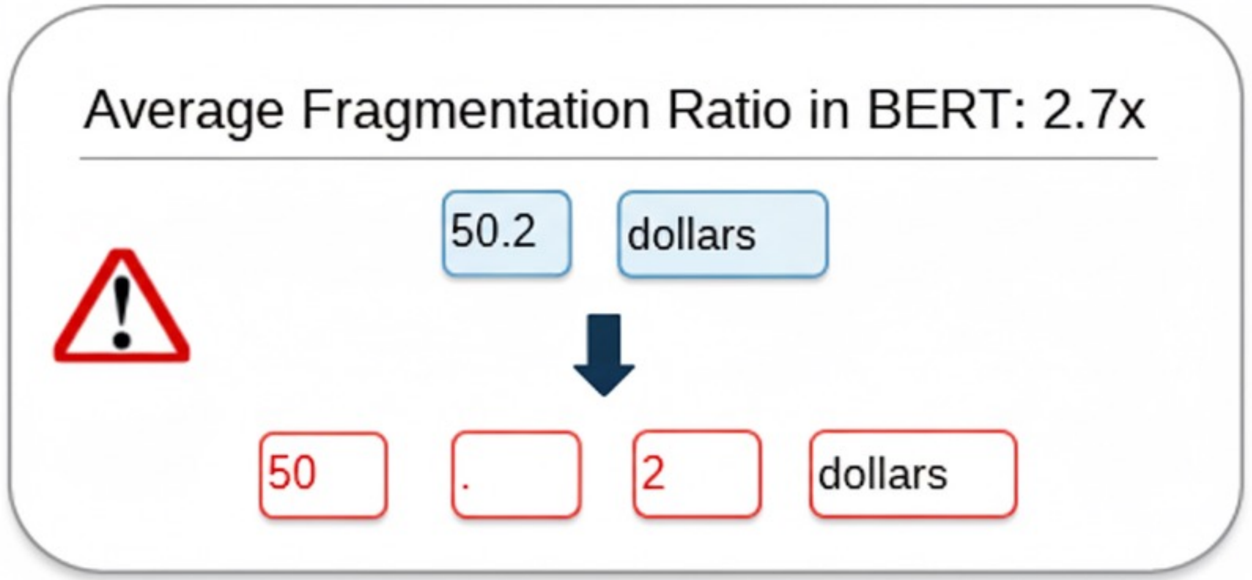
8:50 PM · Jul 25, 2024 · **399.3K** Views

FiNER-139 (2022)

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]**
- <u>Solves fragmentation</u> 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	μ-F_1	m-F_1
BERT	75.1 ± 1.1	72.6 ± 1.4
BERT + CRF	$\underline{78.0} \pm 0.5$	$\underline{75.2} \pm 0.6$
BERT + [NUM]	78.3 ± 0.7	75.7 ± 0.9
BERT + [SHAPE]	$\underline{79.4} \pm 0.2$	$\underline{77.2} \pm 0.2$

Table 4: Entity-level Micro-F1 (μ-F_1) and Macro-F1 (m-F_1) Score ± std (3 runs) on the test data for BERT-based models.

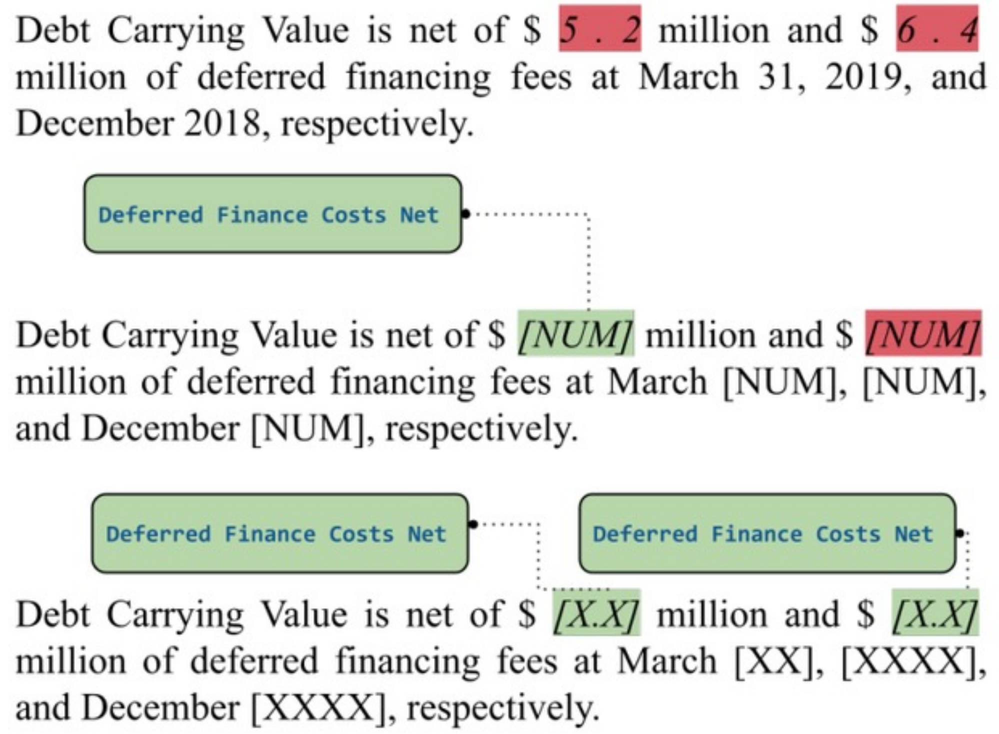
Experimental studies

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	μ-F_1	m-F_1
BERT	75.1 ± 1.1	72.6 ± 1.4
BERT + CRF	78.0 ± 0.5	75.2 ± 0.6
BERT + [NUM]	78.3 ± 0.7	75.7 ± 0.9
BERT + [SHAPE]	79.4 ± 0.2	77.2 ± 0.2

Table 4: Entity-level Micro-F1 (μ-F_1) and Macro-F1 (m-F_1) Score ± 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]

BERT-based methods	μ-F_1	m-F_1
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
FIN-BERT	74.0 ± 1.1	71.3 ± 1.2
FIN-BERT + [NUM]	78.8 ± 0.3	76.3 ± 0.5
FIN-BERT + [SHAPE]	<u>80.1</u> ± 1.4	<u>77.8</u> ± 2.0

Table 4: Entity-level Micro-F1 (μ-F_1) and Macro-F1 (m-F_1) Score ± 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	μ-F_1	m-F_1
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
FIN-BERT	74.0 ± 1.1	71.3 ± 1.2
FIN-BERT + [NUM]	78.8 ± 0.3	76.3 ± 0.5
FIN-BERT + [SHAPE]	<u>80.1</u> ± 1.4	<u>77.8</u> ± 2.0
SEC-BERT (ours)	75.7 ± 0.1	72.6 ± 0.4

Table 4: Entity-level Micro-F1 (μ-F_1) and Macro-F1 (m-F_1) Score ± 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)
- Pre-training SEC-BERT using special numeric tokens is a better strategy than trying to acquire this knowledge only during fine-tuning
- <u>SEC-BERT-SHAPE > all other methods</u>

BERT-based methods	μ-F_1	m-F_1
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
FIN-BERT	74.0 ± 1.1	71.3 ± 1.2
FIN-BERT + [NUM]	78.8 ± 0.3	76.3 ± 0.5
FIN-BERT + [SHAPE]	<u>80.1</u> ± 1.4	<u>77.8</u> ± 2.0
SEC-BERT (ours)	75.7 ± 0.1	72.6 ± 0.4
SEC-BERT-NUM (ours)	80.4 ± 1.4	78.3 ± 1.6
SEC-BERT-SHAPE (ours)	**82.1** ± 0.1	**80.1** ± 0.1

Table 4: Entity-level Micro-F1 (μ-F_1) and Macro-F1 (m-F_1) Score ± 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:

	μ-F_1	m-F_1
Claude Sonnet 4	8.31%	7.88%
Claude Sonnet 4 + [NUM]	9.64%	9.61%
Claude Sonnet 4 + [SHAPE]	**10.60%**	**10.35%**

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!**

	μ-F$_1$	m-F$_1$
BILSTM (subwords)	71.3 ± 0.2	68.6 ± 0.2
BILSTM (subwords) + CRF	76.2 ± 0.2	73.4 ± 0.3
BILSTM-NUM (subwords)	75.6 ± 0.3	72.7 ± 0.4
BILSTM-SHAPE (subwords)	**76.8** ± 0.2	**74.1** ± 0.3

Table 6: Entity-level μ-F$_1$ and m-F$_1$ (%, avg. of 3 runs with different random seeds, ± 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

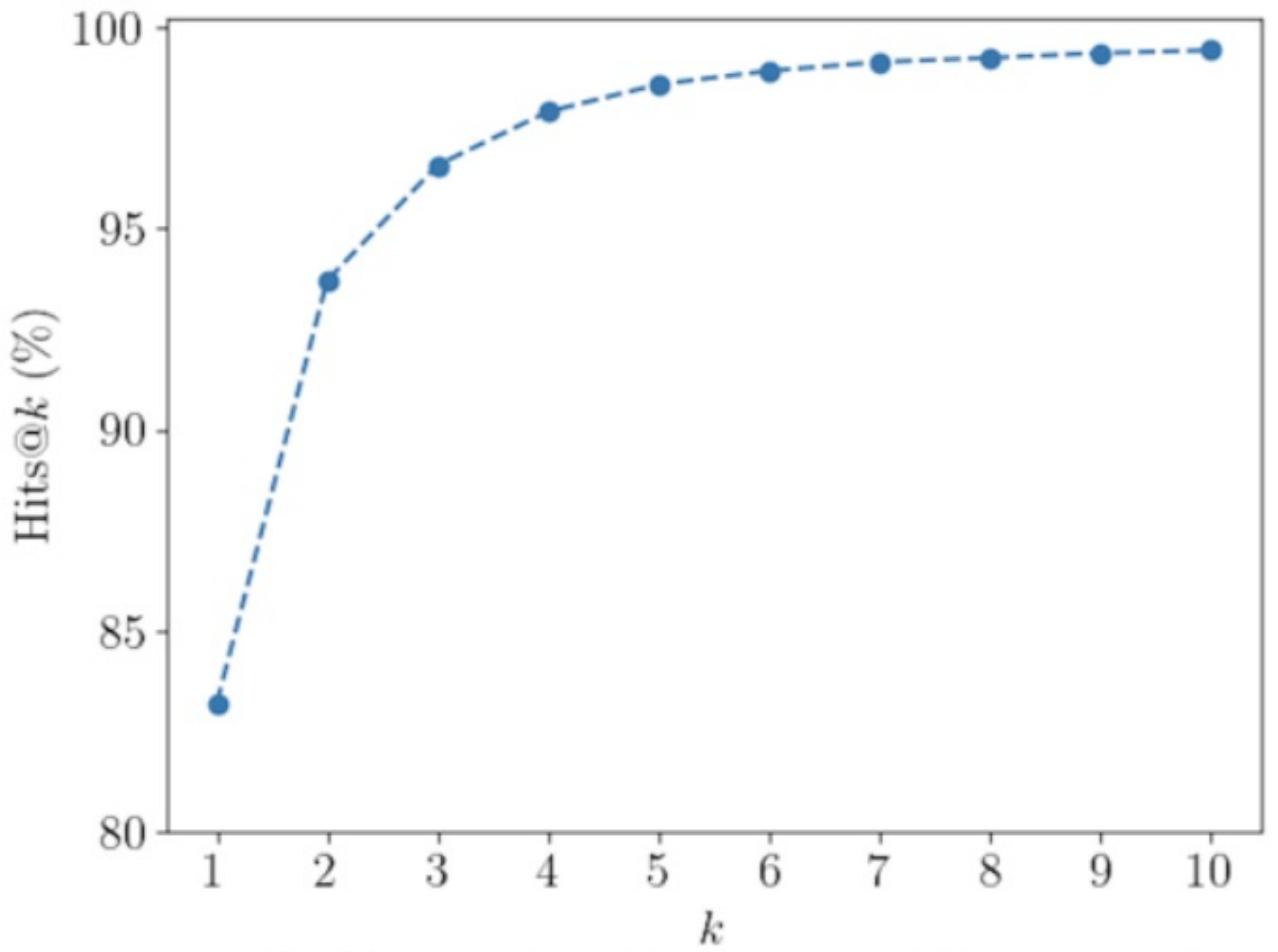
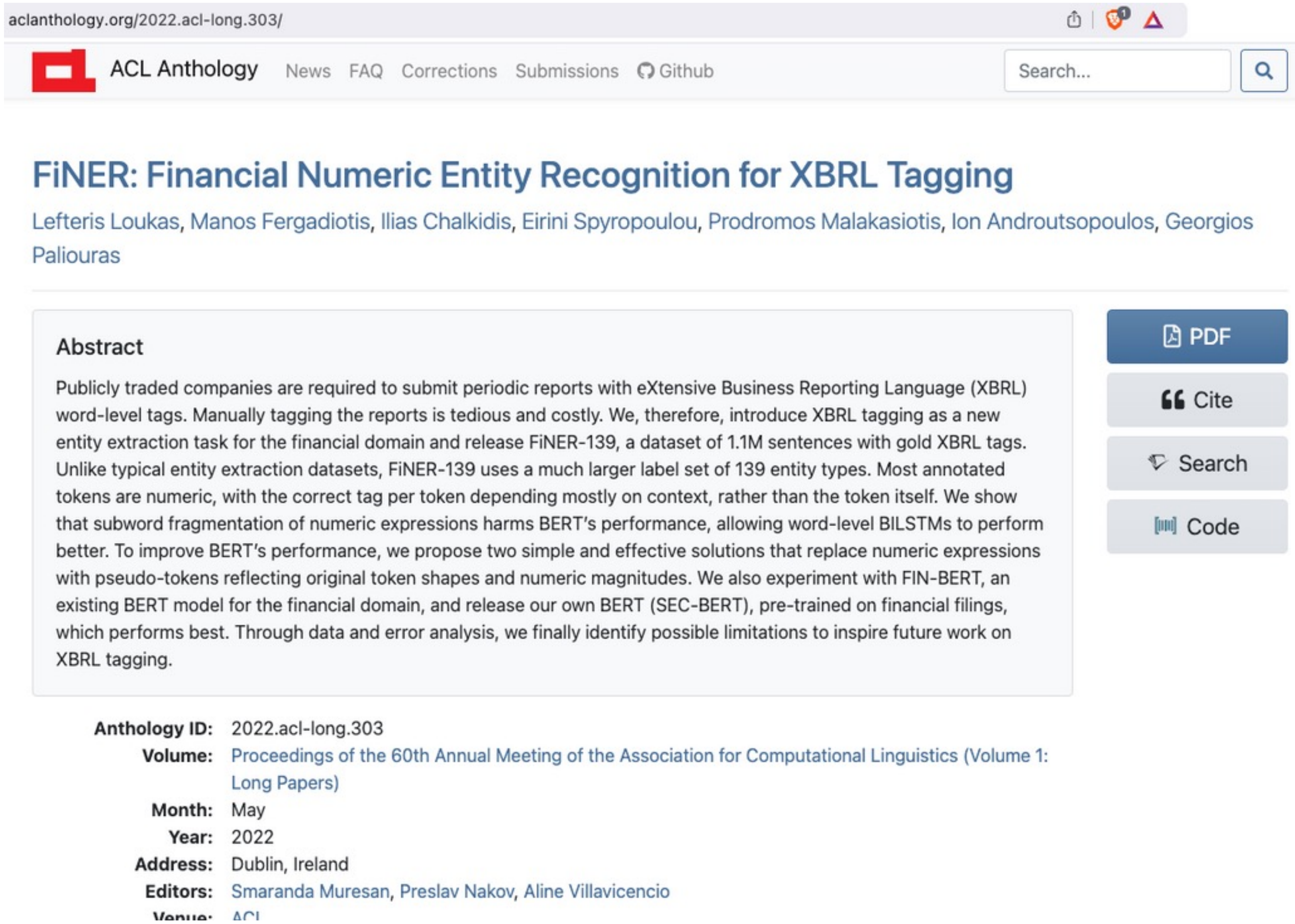


Figure 4: Hits@k results (%, avg. of 3 runs with different random seeds) on test data, for different k values. Standard deviations were very small and are omitted.

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.



aclanthology.org/2022.acl-long.303/

ACL Anthology News FAQ Corrections Submissions Github Search...

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

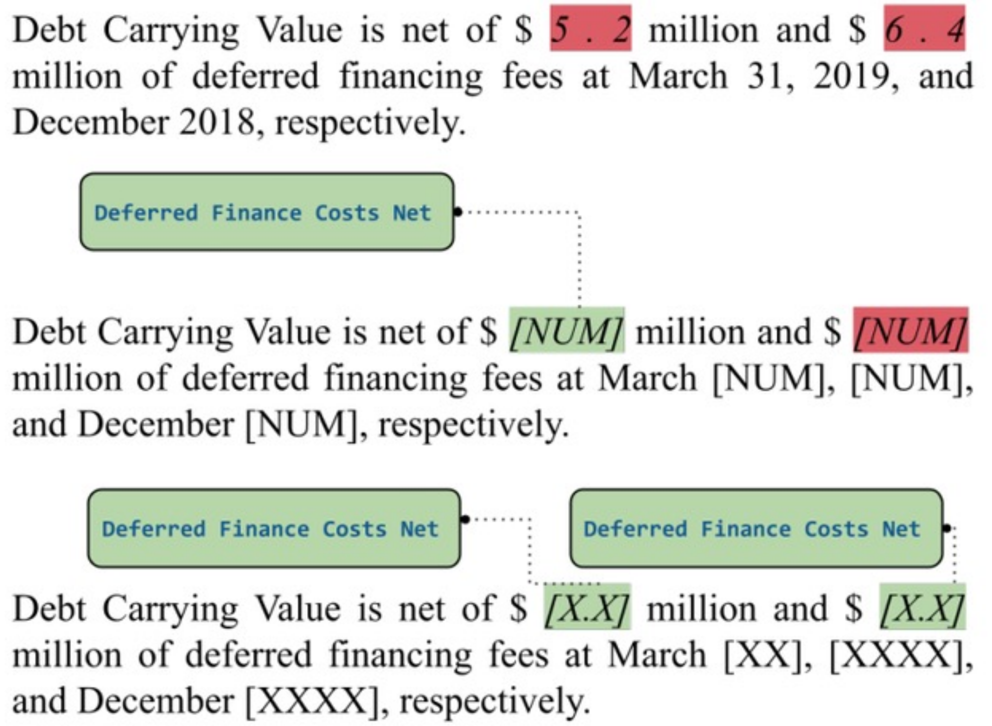
Anthology ID: 2022.acl-long.303
Volume: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)
Month: May
Year: 2022
Address: Dublin, Ireland
Editors: Smaranda Muresan, Preslav Nakov, Aline Villavicencio

- https://huggingface.co/nlpaueb/sec-bert-base
- https://huggingface.co/nlpaueb/sec-bert-num
- https://huggingface.co/nlpaueb/sec-bert-shape
- https://huggingface.co/datasets/nlpaueb/finer-139

FiNER: Financial Numeric Entity Recognition for XBRL Tagging

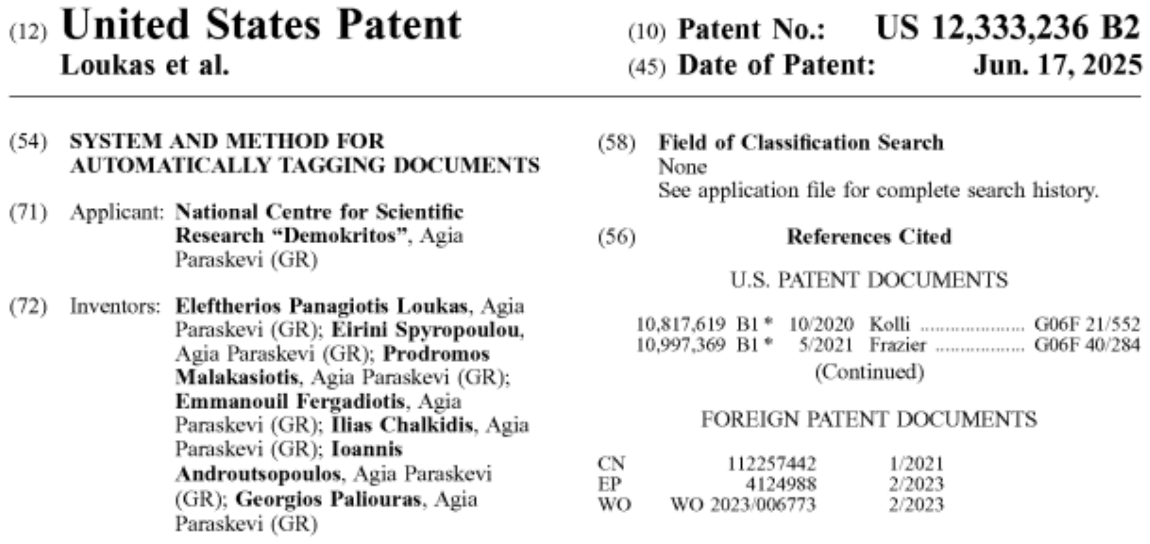
- Published at **ACL 2022** (A* CORE ranking conference)
- **Problem**: token overfragmentation of Transformer models in numbers
- **Solution**: new tokenization method for pre-training and fine-tuning so transformers learn better number representations
- Created new BERT models & dataset (https://huggingface.co/nlpaueb/sec-bert-base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

CITED BY	YEAR
98	2022



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**



(12) **United States Patent**
Loukas et al.

(10) **Patent No.:** US 12,333,236 B2
(45) **Date of Patent:** Jun. 17, 2025

(54) **SYSTEM AND METHOD FOR AUTOMATICALLY TAGGING DOCUMENTS**

(71) Applicant: **National Centre for Scientific Research "Demokritos"**, Agia Paraskevi (GR)

(72) Inventors: **Eleftherios Panagiotis Loukas**, Agia Paraskevi (GR); **Eirini Spyropoulou**, Agia Paraskevi (GR); **Prodromos Malakasiotis**, Agia Paraskevi (GR); **Emmanouil Fergadiotis**, Agia Paraskevi (GR); **Ilias Chalkidis**, Agia Paraskevi (GR); **Ioannis Androutsopoulos**, Agia Paraskevi (GR); **Georgios Paliouras**, Agia Paraskevi (GR)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

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

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

DEMOKRITOS

EY

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

Dataset
/banking77

User queries

- My card is needed soon.
- My transfer got declined!
- How can I replace my expired card?

Intent labels

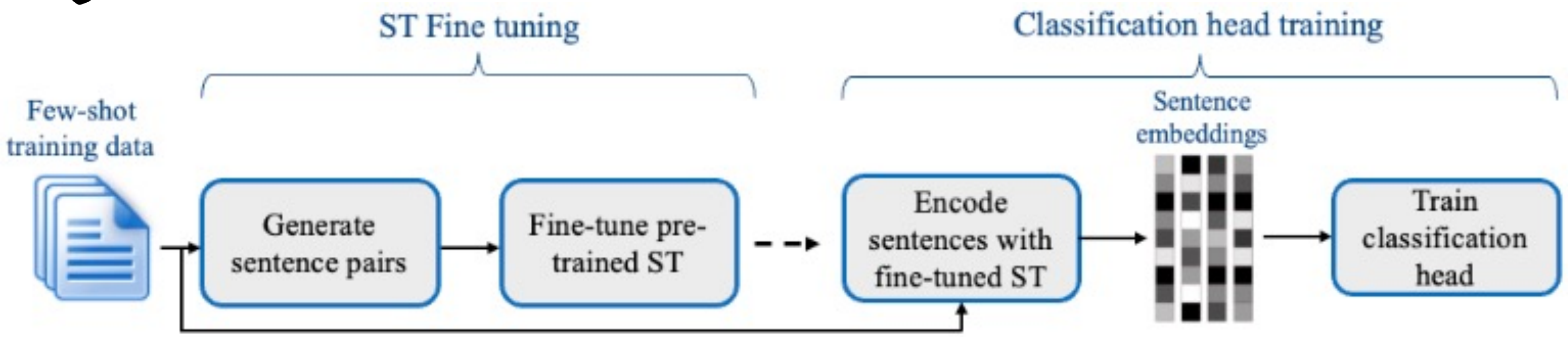
- Card Delivery Estimate
- Declined Transfer
- Card About To Expire

- **Intent detection is a classification task**, which can be solved in numerous ways
  - Full-Data Setting (>1,000 samples per class)
    - In <u>business</u> settings, it is unfeasible to get so much data :(
  - Few-shot Setting (1-20 samples per class)
    - Contrastive Learning with MLMs (<= 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 💲💲💲 :(

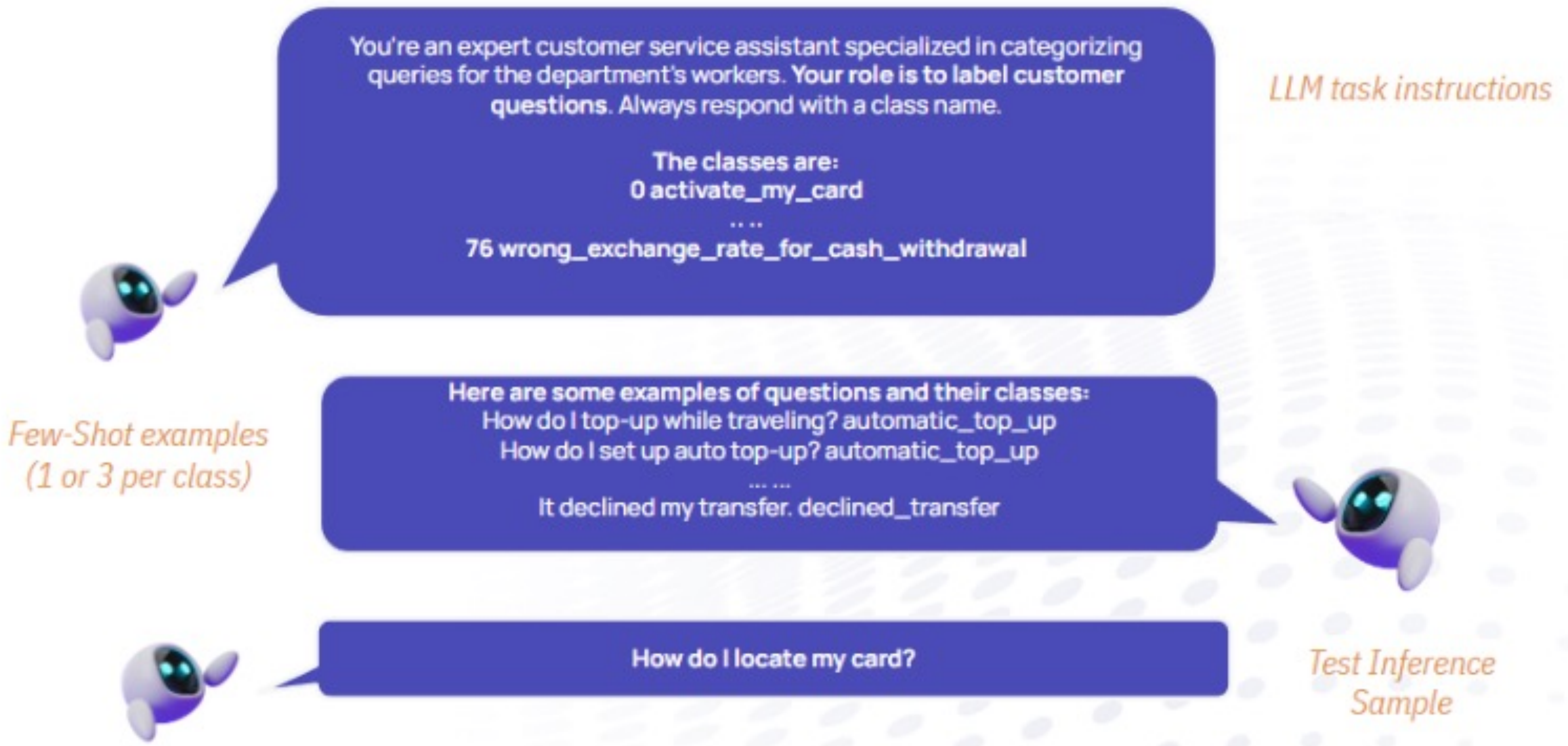
1. *"Breaking the Bank with ChatGPT: Few-Shot Text Classification for Finance"*. L. Loukas, I. Stogiannidis, P. Malakasiotis, S. Vassos. **FinNLP @ IJCAI 2023**
2. *"Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking"*. L. Loukas, I. Stogiannidis. O. Diamantopoulos, P. Malakasiotis, 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 MLMs. 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.

helvia.ai

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 >=3 samples using SetFit
- **When trained on only 3 samples, MPNet-v2 achieves scores of 76.7 μ-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	μ-F_1	m-F_1
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	**94.1**	**94.1**
MPNet-v2 (SetFit)	1-shot x 77	**57.4**	**55.9**
GPT-3.5 (representative samples)	1-shot x 77	**75.2**	**74.3**
GPT-3.5 (random samples)	1-shot x 77	74.0	72.3
GPT-4 (representative samples)	1-shot x 77	**80.4**	**78.1**
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	**76.7**	**75.9**
GPT-3.5 (random samples)	3-shot x 77	57.9	59.8
GPT-3.5 (representative samples)	3-shot x 77	**65.5**	**65.3**
GPT-4 (representative samples)	3-shot x 77	**83.1**	**82.7**
GPT-4 (random samples)	3-shot x 77	74.2	73.7
MPNet-v2 (SetFit)	5-shot x 77	**83.5**	**83.3**
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	**91.2**	**91.3**

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	μ-F_1	m-F_1
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	**94.1**	**94.1**
MPNet-v2 (SetFit)	1-shot x 77	**57.4**	**55.9**
GPT-3.5 (representative samples)	1-shot x 77	**75.2**	**74.3**
GPT-3.5 (random samples)	1-shot x 77	74.0	72.3
GPT-4 (representative samples)	1-shot x 77	**80.4**	**78.1**
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	**76.7**	**75.9**
GPT-3.5 (random samples)	3-shot x 77	57.9	59.8
GPT-3.5 (representative samples)	3-shot x 77	**65.5**	**65.3**
GPT-4 (representative samples)	3-shot x 77	**83.1**	**82.7**
GPT-4 (random samples)	3-shot x 77	74.2	73.7
MPNet-v2 (SetFit)	5-shot x 77	**83.5**	**83.3**
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	**91.2**	**91.3**

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

helvia.ai

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

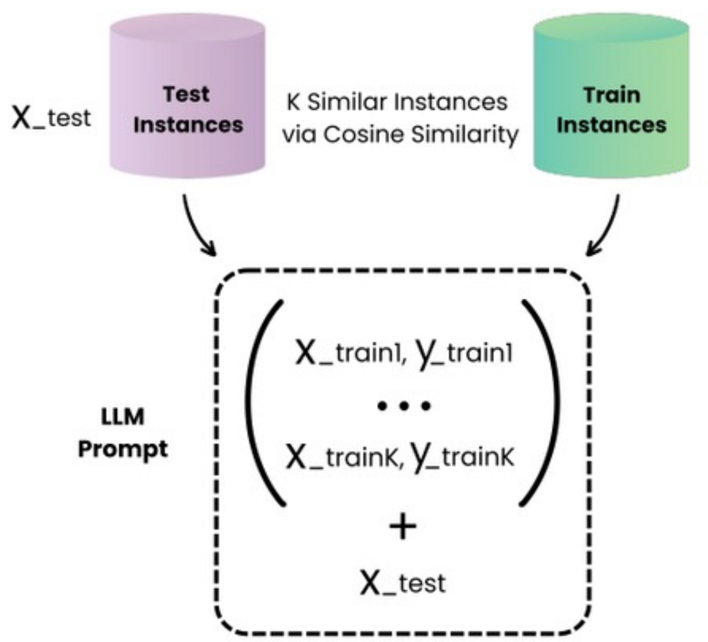


Figure 3: Dynamic LLM prompt construction through Retrieval-Augmented Generation (RAG), using cosine similarity for in-context data selection. We use K=5, 10, 20.

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**

In-Context Learning for Text Classification with Many Labels

Aristides Milios[1], Siva Reddy[1,2,3], Dzmitry Bahdanau[1,2]
Mila and McGill University[1], ServiceNOW Research[2], Facebook CIFAR AI Chair[3]
{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.*

- 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$ 🤯

Cost-effective LLM inference with Dynamic Few-Shot Prompting

Summary (cost-wise):

- **APIs charge based on token usage**
- **Standard Few-Shot: Cost ∝ S × C (shots × classes)**
- **Dynamic Few-Shot: Cost ∝ K (retrieved examples only during inference)**

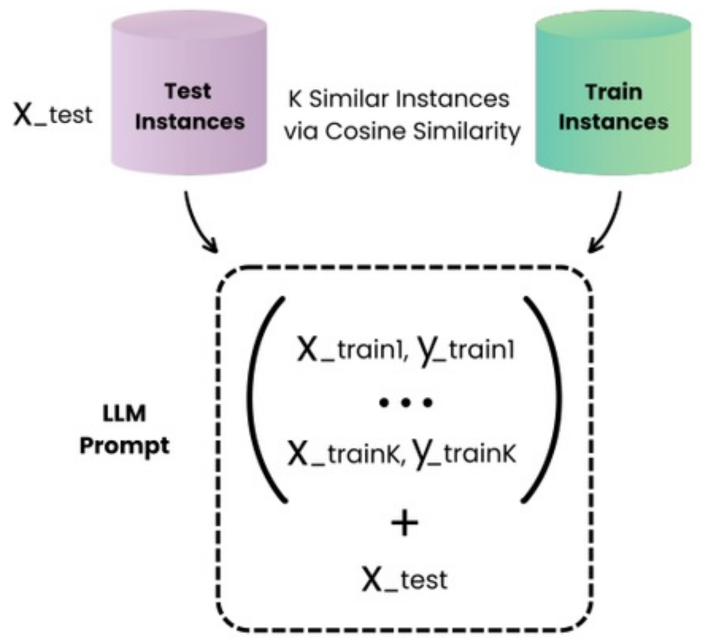


Figure 3: Dynamic LLM prompt construction through Retrieval-Augmented Generation (RAG), using cosine similarity for in-context data selection. We use K=5, 10, 20.

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	μ-F_1	m-F_1
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	**94.1**	**94.1**
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	**80.4**	**78.1**

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 Retrievel-augmented Generation (RAG), they analyzed the performance-cost curve for a wide variety of workflows.

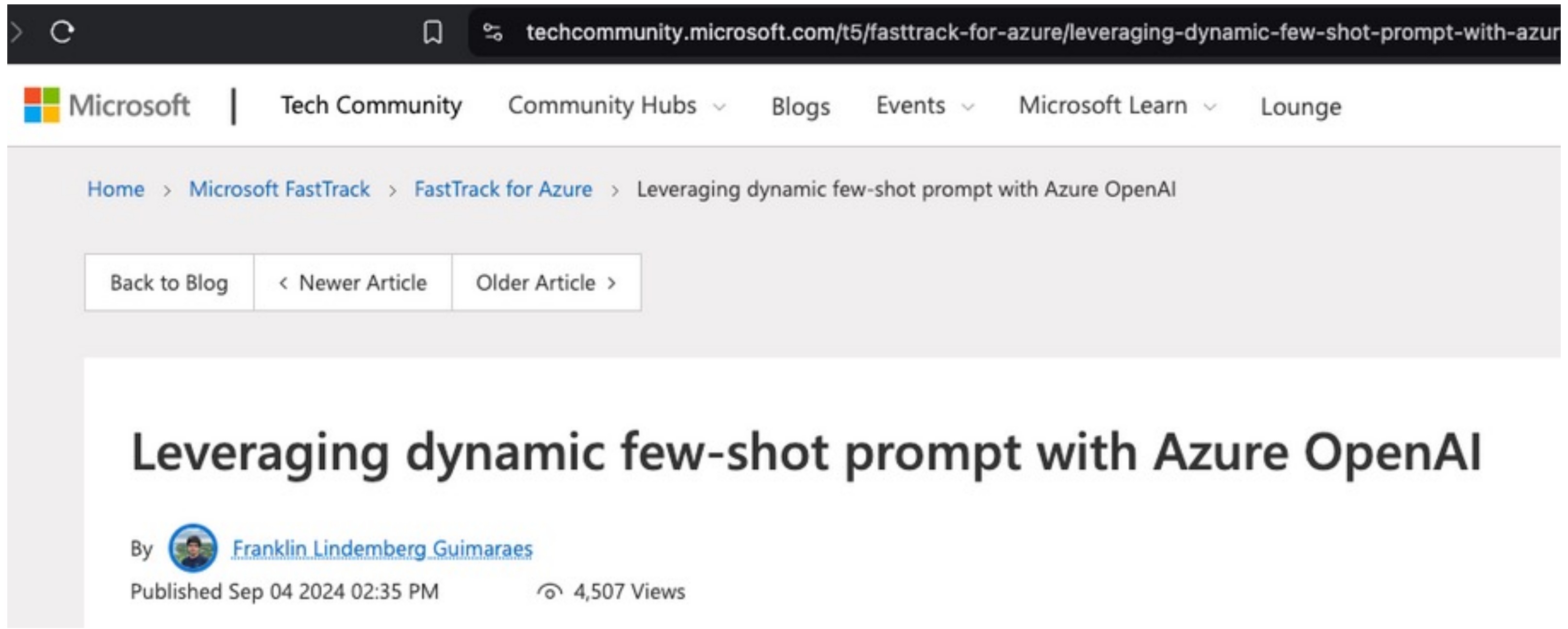
What's your experience? Are you still using SFT methods because they're cheaper and - given the data - more powerful? Or did you fall in love with the flexibility of few-shot learning and RAG? Let us know.

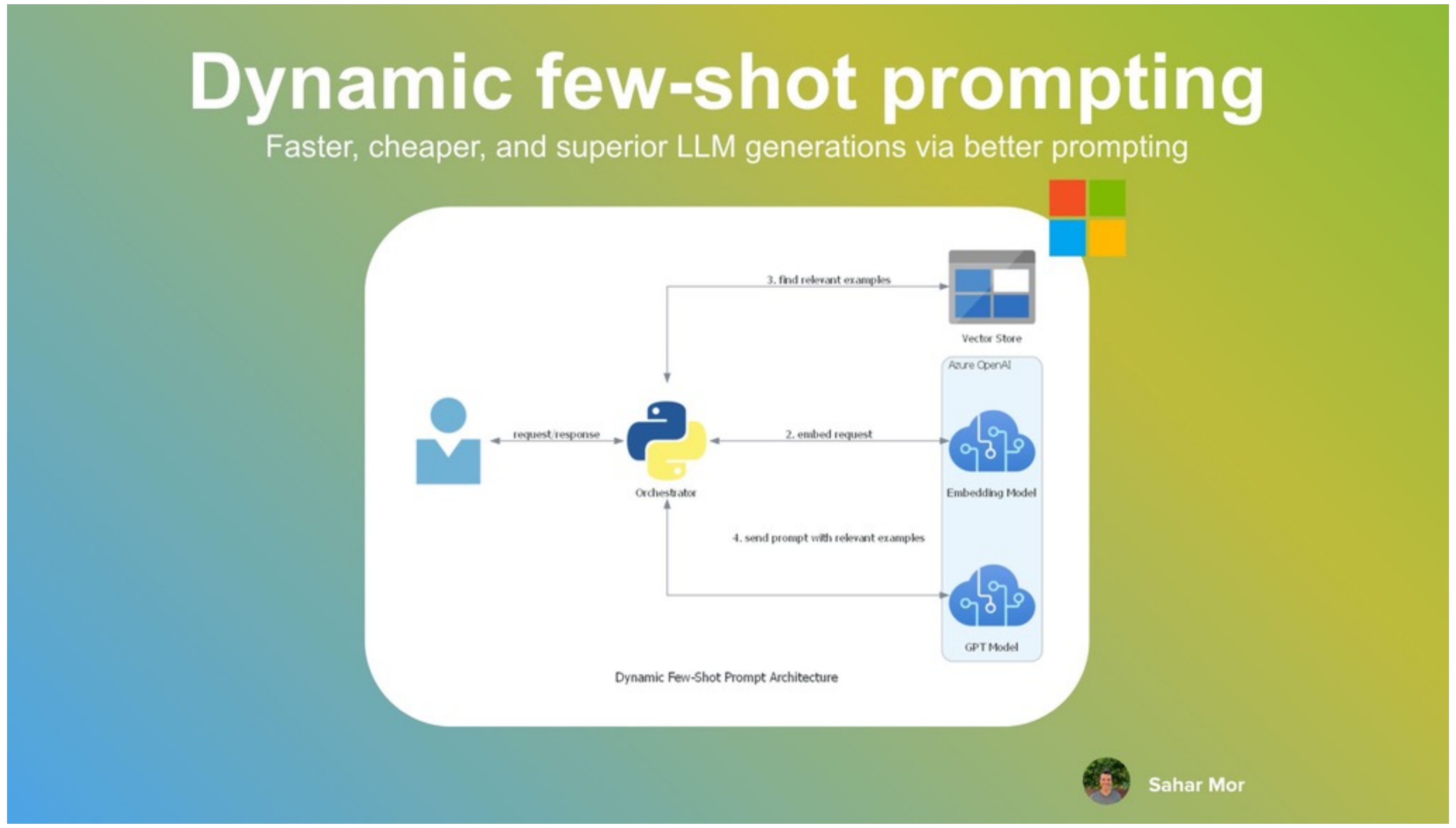
[arXiv] https://lnkd.in/dGMQzzY4
↓
Liked this post? Get weekly AI highlights and papers-of-the-week directly to your inbox 👉 llmwatch.com

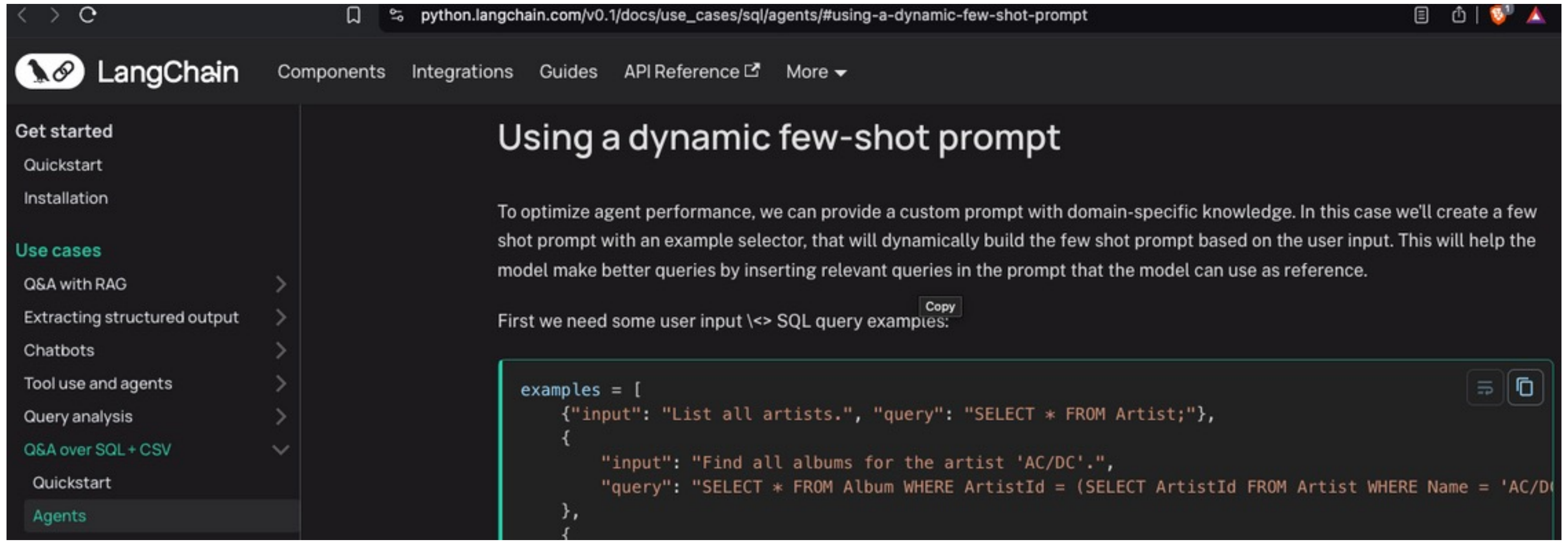
For relatively "easy" tasks, GPT-4 may often not be worth it. RAG can level the playing field even further.

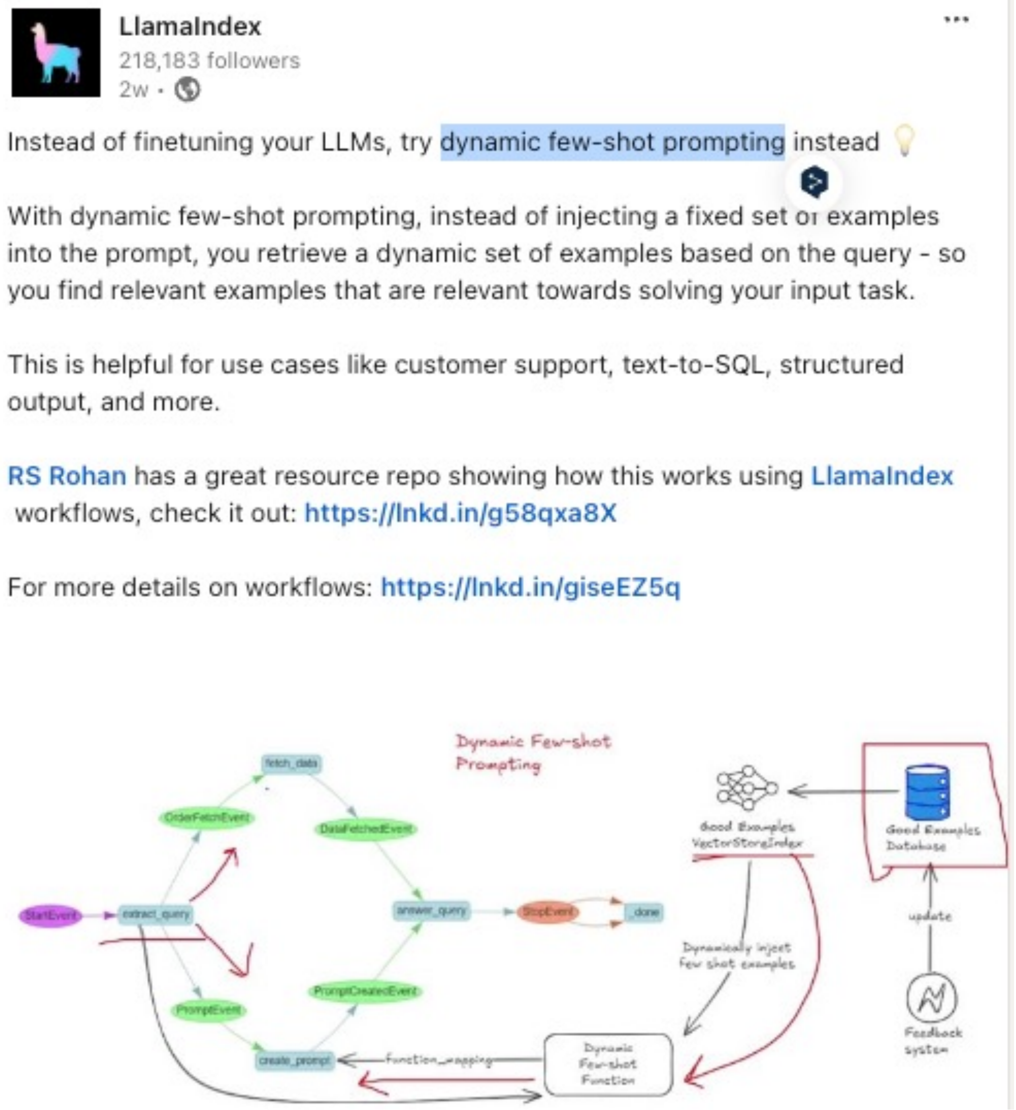
Model	Setting	Micro-F1↑	Cost↓
GPT-4	1-shot	**80.4**	620$
GPT-3.5	1-shot	75.2	31$
Anthropic Claude 2	1-shot	76.8	**15$**
Command-nightly	1-shot	58.4	22$

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









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/

helvia.ai

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. *"DICoE@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

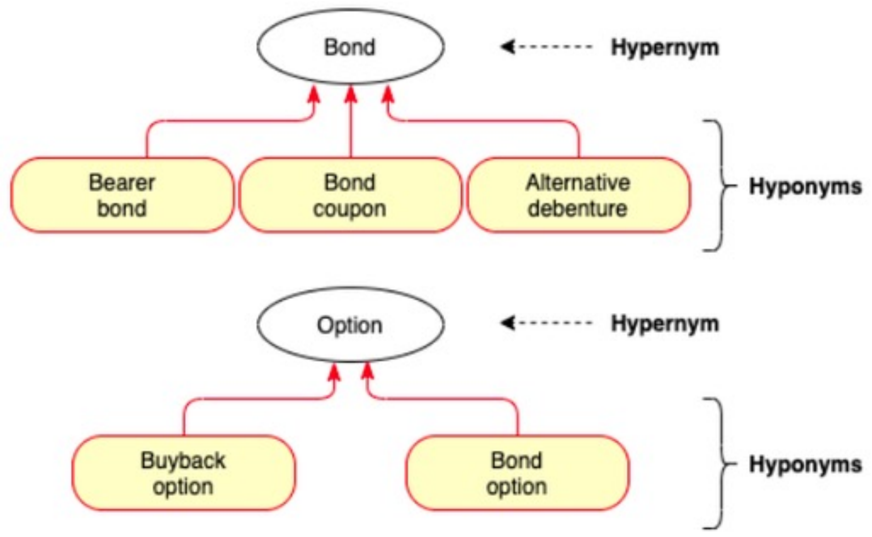


Figure 1: Examples of hypernym relations from the FIBO ontology. Interestingly, "Bond coupon" is a kind of "Bond", but "Bond option" is a kind of "Option".

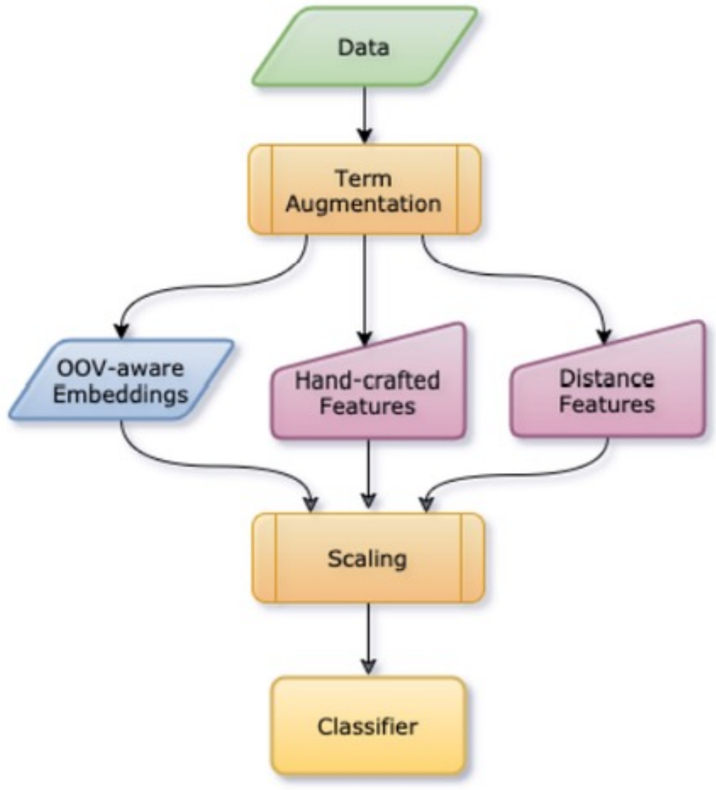


Figure 2: The pipeline of our best system.

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!