

Open in app ↗

Sign up

Sign In



Search Medium



Using NLP on a company's annual reports to predict near bankruptcy

Can the language of company annual reports predict equity price collapse?



Gary Licht · Follow

Published in Towards Data Science

7 min read · Sep 8, 2020



Listen



Share

After trading fixed income and currencies for many years, I was looking for a project as an introduction to natural language processing (NLP). After witnessing so many scandals in finance and the general corporate world, I was intrigued by the idea that the language of a company annual report could carry a forensic signature of cheating or irresponsibility. A case of read what I say, not the numbers I report.

Maximum drawdown is the maximum mark-to-market loss of a portfolio or security over a given period and is a widely used risk management metric. The goal of the project is to see if sharp and extreme equity drawdown, as a proxy for bankruptcy, credit risk and governance, can be predicted from the text of a company's annual report and to compare this with the performance from more traditional measures using financial metrics (FIN) and market indicators (MKT).

Formal Set-up

The problem is set up as a binary classification problem with the target being the maximum rolling 20-day drawdown of the equity price in the year following the release

of the annual report. The target registers an “almost bankrupt” (positive) event when the drawdown is greater than or equal to 80%. The feature set is derived from preprocessing the text of company annual reports and representing these in a term-frequency-inverse-document-frequency (tf-idf) matrix.

Databases

Sharadar offers a reasonably priced subscription covering over 20 years of history across 14,000+ US companies and is accessible through the Quandl API. Importantly for our application, the Sharadar stock database includes bankrupt and other delisted companies. Public companies are required to file annual reports (10K) with the SEC and these are available from the EDGAR database on the SEC website. An existing Python package was used to scrape this data. The stock price database provided 160,926 potential target events of which 38,807 could be matched with the downloaded annual report database.



Schematic of databases (Image by Author)

Text Preprocessing

Given the multi-decade span of the data, the annual reports are in multiple formats including text, html and XBRL. As a result, Regex was used to parse the reports and remove special characters:

```
def remove_html_tags_char(text):
    '''Takes in string and removes special characters '''

    #Define special Chars
    clean1 = re.compile('\n')
    clean2 = re.compile('\r')
    clean3 = re.compile('&nbsp;')
    clean4 = re.compile('&#160;')
    clean5 = re.compile(' ')
```

```
#Define html tags
clean6 = re.compile('<.*?>')
#remove special characters and html tags
text = re.sub(clean1,' ', text)
text = re.sub(clean2,' ',text)
text = re.sub(clean3,' ',text)
text = re.sub(clean4,' ',text)
text = re.sub(clean5,' ',text)
text = re.sub(clean6,' ',text)
# check spacing
final_text = ' '.join(text.split())

return final_text
```

In keeping with modern methods, the text was minimally processed with no lemmatization, stemming or stop-word removal. This follows the idea that with large enough data sets, one should let the model determine whether these nuances (e.g. tense) are important for the problem at hand.

Data Exploration

The data is highly **imbalanced**:

	# Samples	Ratio Pos
Market Database	160,926	2.1%
After Matching with 10Ks	38,807	3.6%
Training Set	30,569	3.5%
Holdout Set	8,238	4.2%

Evolution of minority class ratio across pipeline (Image by Author)

Not surprisingly, **“almost bankruptcy” spikes over US recessions** (relationship to actual US GDP is more nuanced- more on GitHub):

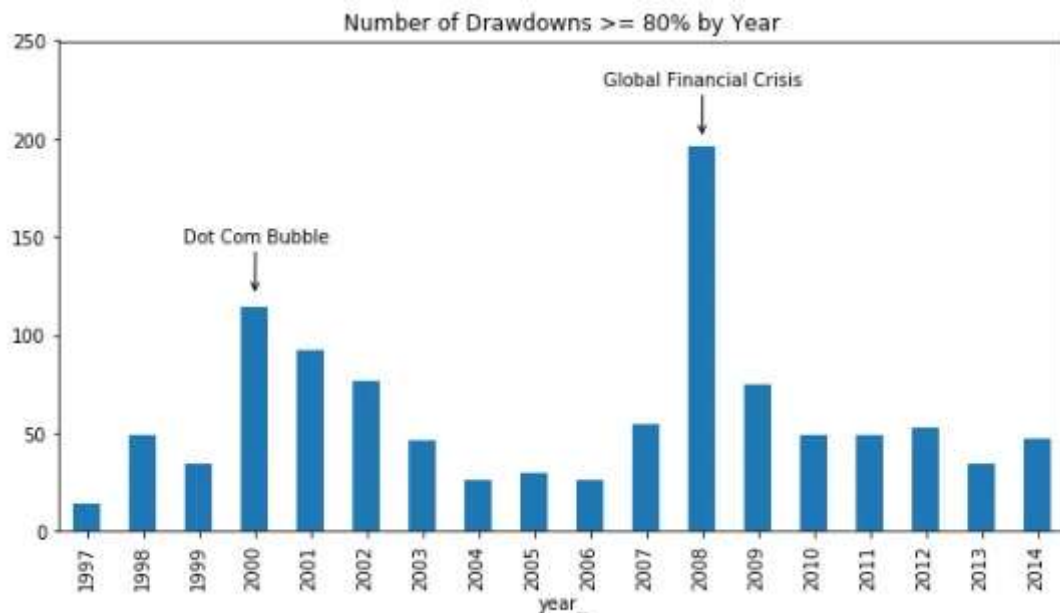


Image by Author

Like many other market price variables, **auto-correlation or persistence** is evident in equity drawdown:

Histogram: Previous Quartile of Company before a Positive Event

```

quartile 1 = 3.43 %
quartile 2 = 9.19 %
quartile 3 = 23.15 %
quartile 4 = 64.23 %

```

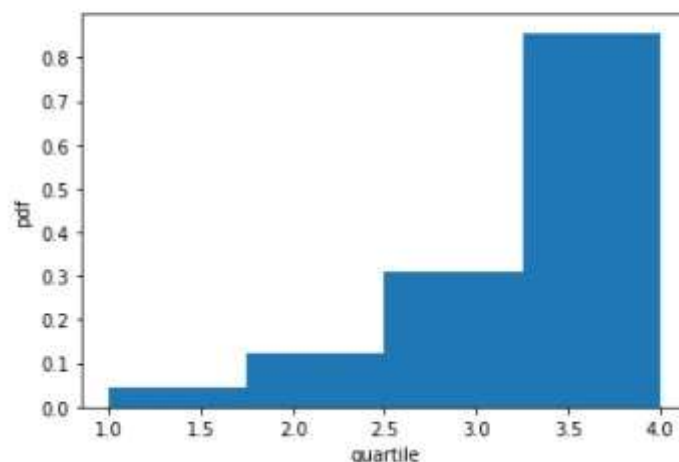


Image by Author

This last observation leads to the introduction of the market (MKT) model for baseline comparative purposes. MKT simply uses a company's prior annual drawdown quartile ranking as the sole feature.

Cross-Validation

Method

Data from 2015 onwards is used as the hold-out set while the training data runs from 1997 to 2014. An **expanding window** method is used with the CV set split into 5 equal parts:

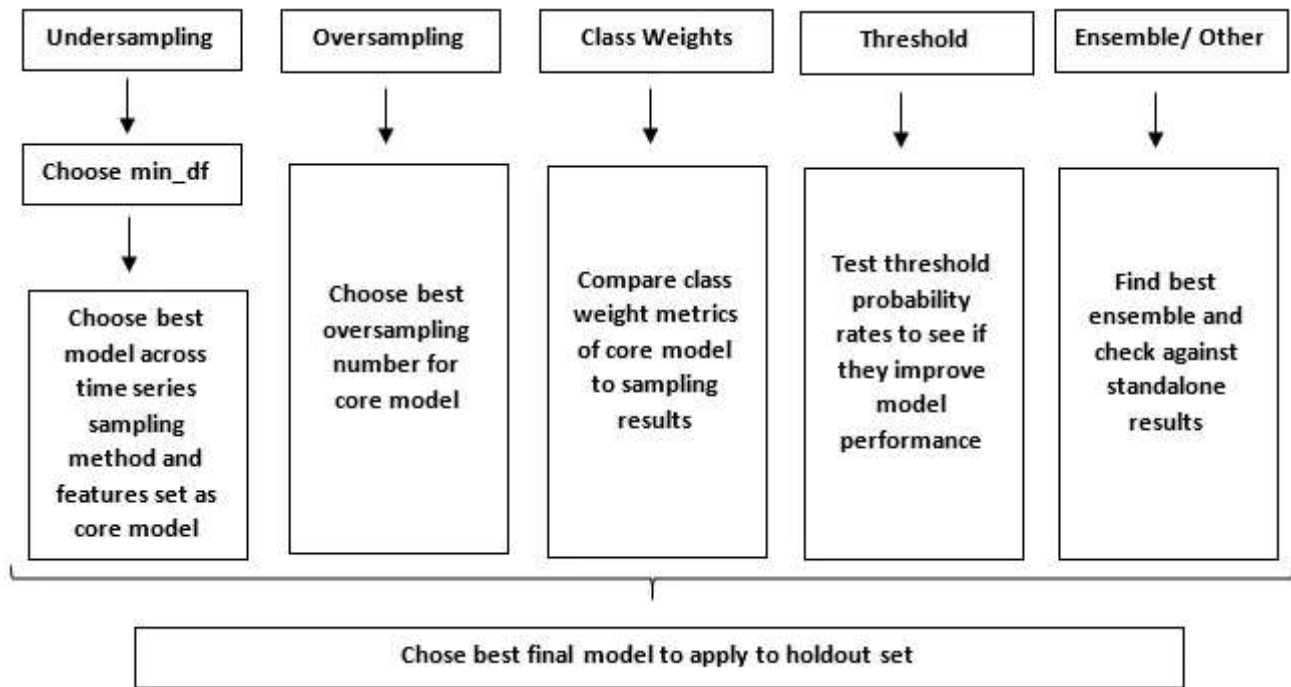


CV set runs from 1997 to 2015 and is split into 4 iterations using the expanding window method (Image by Author)

The tf-idf vectorizer is formed on the CV training data and *transform* is then used to convert the testing documents into its tf-idf matrix. Macro (harmonic) recall is used as the CV score and is weighted in proportion to the amount of data in the training set (i.e. CV1 has weight 10%, CV2 has weight 20% and so on). In cases requiring a tie-break, the standard deviation of the score across CV sets was used.

Structure

Undersampling, oversampling and class-weights were used to tackle data imbalance. Another sample bias considered was the potential for an unequal distribution of financial statements over time to harm the generalization of the model. Indeed, we might expect the changing regulatory, accounting, legal and economic landscape of the last 23 years to be reflected in the language and structure of the annual reports. In order to test for this, validation was performed on (i) a random selection of the majority class data and (ii) a preprocessed sample that kept the ratio of the majority class events per year equal to that of the minority class (*time equalization*). Finally, various probability threshold values and ensembles were tested.



Schematic of Machine Learning Implementation (Image by Author)

Key Results

All Optimal Model CV Summary:				
OPTIMAL_MODEL	mh_recall	pos_recall	neg_recall	notes
undersampling	0.660	0.760	0.580	grad_boost
oversampling	0.640	0.550	0.770	grad_boost & n=2
class_weights	0.004	0.002	0.995	SK_Learn "balanced" RF
under_thresh	0.670	0.650	0.700	grad_boost & thresh_60%
over_thresh	0.660	0.700	0.640	grad_boost & thresh_40%
ensemble_under	0.680	0.700	0.660	grad_boost_w_50%, rand_forest_w_50%, thresh_55%
ensemble_over	0.680	0.710	0.650	grad_boost_w_40%, log_reg_w_60%, thresh_40%
ensemble_mixed	0.690	0.720	0.660	over_grad_boost_w_25%, under_grad_boost_w_75%, thresh_50%

CV performance metrics of various models under optimal fine tuning (Image by Author)

- Undersampling performed better than oversampling
- Time equalization improves the undersampling score from 62% to 66%
- Gradient boosting performs better than Random Forest and Logarithmic Regression
- Undersampling favored positive recall (sensitivity) while oversampling favored negative recall (specificity)

- The optimal model is an ensemble of oversampling / undersampling in a ratio of 25% / 75% with gradient boosting used for both models

Holdout Testing

Results

The optimal model (NLP) was applied to the holdout set on an annual expanding window basis. Aggregate results were then compared to baseline financial ratio (FIN) and market (MKT) models:

Holdout Results:

Recall

	mh_recall	pos_recall	neg_recall	accuracy
NLP	0.69	0.67	0.72	0.72
FIN	0.70	0.71	0.70	0.70
MKT	0.73	0.68	0.78	0.77

Other Metrics

	m_prec	pos_prec	neg_prec	m_f1	pos_f1	neg_f1
NLP	0.54	0.10	0.98	0.50	0.18	0.83
FIN	0.54	0.10	0.98	0.50	0.18	0.82
MKT	0.55	0.13	0.98	0.54	0.22	0.87

Aggregate holdout results of NLP compared to FIN and MKT (Image by Author)

The NLP confusion matrix is:

NLP Confusion Matrix

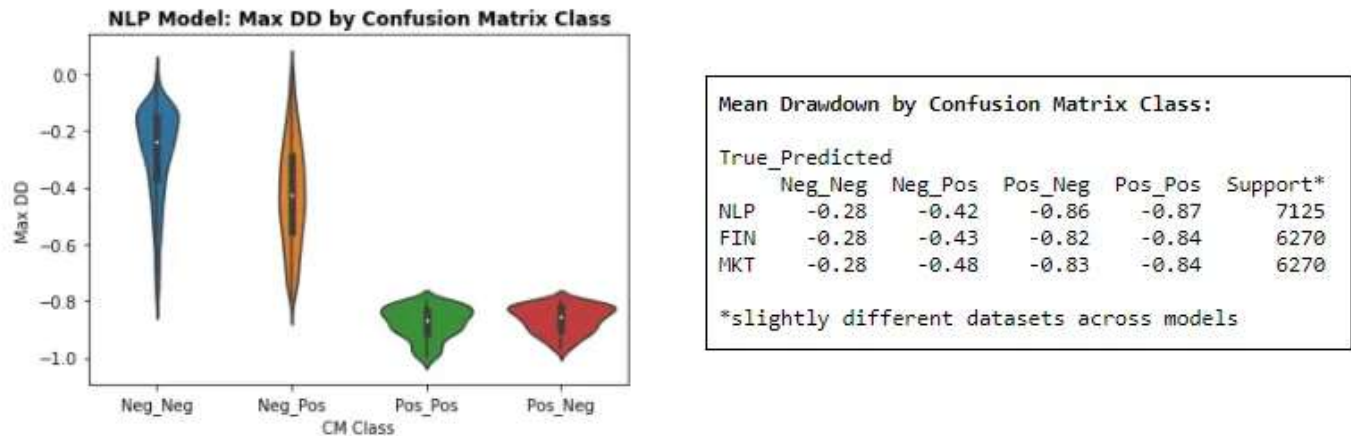
True label	Neg	4,900	1,902
	Pos	107	216
		Neg	Pos
		Predicted label	

Image by Author

Analysis

One of the benefits of having a continuous variable underlying the binary target is that we can compute the cost of model errors by calculating the drawdown statistics for

each class in the confusion matrix:



False positives capture useful information about continuous target space (Image by Author)

Despite the relatively high proportion of false positives and the associated low precision, the model errors reveal interesting information about the classification space

This can be more practically fleshed out in the domain space by forming evenly weighted portfolios of predicted true, predicted negative and all:

Portfolio Drawdown by Prediction:

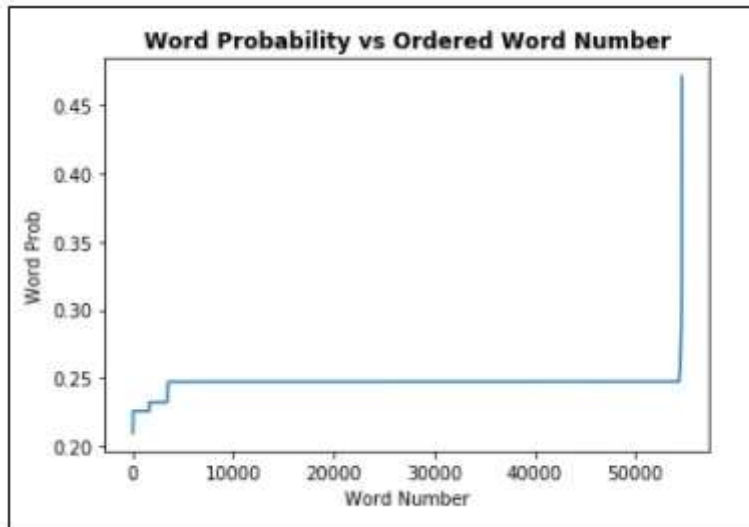
	Pred Pos	Pred Neg	Neg-Pos	All
NLP	-0.47	-0.29	0.18	-0.34
FIN	-0.47	-0.29	0.18	-0.35
MKT	-0.52	-0.29	0.23	-0.35

Domain perspective of model utility (Image by Author)

Interpretation

Interpretation of these models is somewhat challenging, but we can look at what words in the holdout set have the highest probabilities on a standalone basis. This is done by forming a document matrix where each document only has one unique word and then running this through model prediction. The caveat is that a word can have a high probability while being specific to one or very few documents. In this respect, not all words will be *generally* predictive.

Starting with the metadata, the maximum single word probability is below 50% and 99.9% of the words are between 20 and 30%. This indicates that it is the combination of words found in the document and not single implicating words that produce documents of high probability.



Word Probability Percentiles		
Percentile	Word	Prob
0.0		0.21
25.0		0.25
50.0		0.25
75.0		0.25
99.9		0.28
100.0		0.47

Standalone word probabilities indicate combination of words drives model (Image by Author)

The below list of the **top words** was manually sorted into categories. Of these categories: accounting/credit, consultant speak/business reorganization and negative sentiment words are intuitively pleasing.

Top Words by Manually Sorted Category					
Accounting/Credit	Consultant Speak	Negative Words	Healthcare	Gender	Nonsense?
concern	consultant	reassurance	diabetic	wife	sapphire
lenders	restructure	doubt	trial	girl	intermedia
going	repositioning	bears	enrollment	her	particle
projection	tightened	teeter	cvs		theater
infusion	implementing	congestion			dana
lien	effectuated	endpoint			uunet
dilution	elimination	ceded			extranets
subordinated	resolving	distant			overlying
waived	regain				forrester
revolver	begun				thai
repaid	shorten				supervalu
subprime	stage				coverings
pledge	competent				domes
verified	concurrent				roebuck
raising	innovator				annum
notified	beneficially				lotus
	ven				bt
					heller
					recission

Accounting / Credit and Consultant Speak / Business reorganization are intuitive categories (Image by Author)

Further Thoughts

• It is somewhat surprising to see a relatively simple model such as tf-idf perform so well in comparison to a baseline financial ratio model. The top word categories of credit and business reorganization suggests the model is good at picking up on similar sort of information to that contained in the financial ratios. The next phase of the project is to explore the language related to almost bankrupt events with strong financial ratios. This focus on “additionality” will likely require more sophisticated algorithms (such as BERT) better able to parse context but might be able to shed light on accounting fraud or other dishonest practices absent from the financials.

• Identifying weak companies is one thing but knowing when that weakness will manifest in a positive event is far harder and will also depend on the complex external environment. This is a reasonable interpretation of the high false positive rate. It would be fascinating to see if the text of the reports contained more information on timing than already extracted. One avenue to try would be cosine similarity but this is likely to suffer from the curse of dimensionality. Another approach would be including the annual change in the tf-idf matrix itself for individual companies in the features set. On the aggregate level, one could try to test whether the aggregate annual tf-idf

matrix is predictive of the change in the number of almost bankruptcies in the following year.

Github

• The model, together with other project files, are available on [GitHub](#). Have fun and please feel free to share feedback!

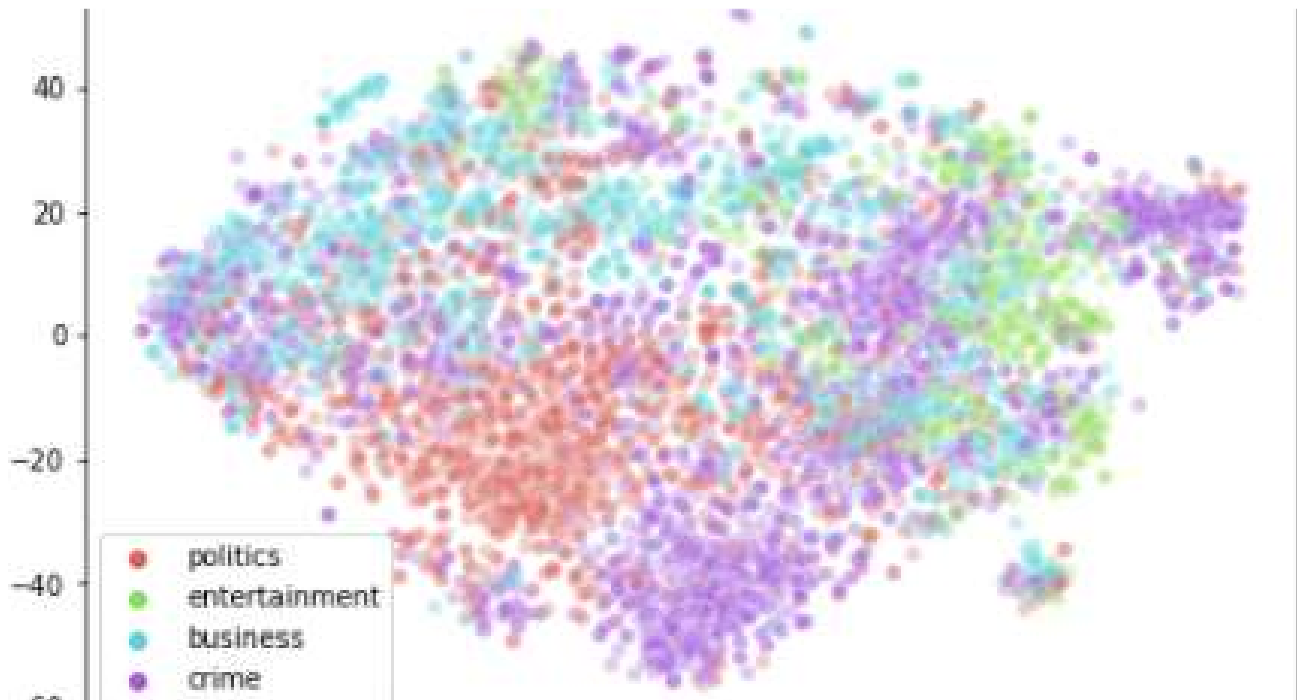
[Data Science](#)[Machine Learning](#)[Naturallanguageprocessing](#)[Quantitative Finance](#)[Stock Market](#)[Follow](#)

Written by Gary Licht

26 Followers · Writer for Towards Data Science

Data / Research / Strategy www.linkedin.com/in/gary-licht-02122548/

More from Gary Licht and Towards Data Science



Gary Licht in Towards Data Science

Extractive Summarization using BERT

A supervised approach harnessing the power of BERT embeddings

8 min read · Oct 30, 2020



186



1





Bex T. in Towards Data Science

130 ML Tricks And Resources Curated Carefully From 3 Years (Plus Free eBook)

Each one is worth your time

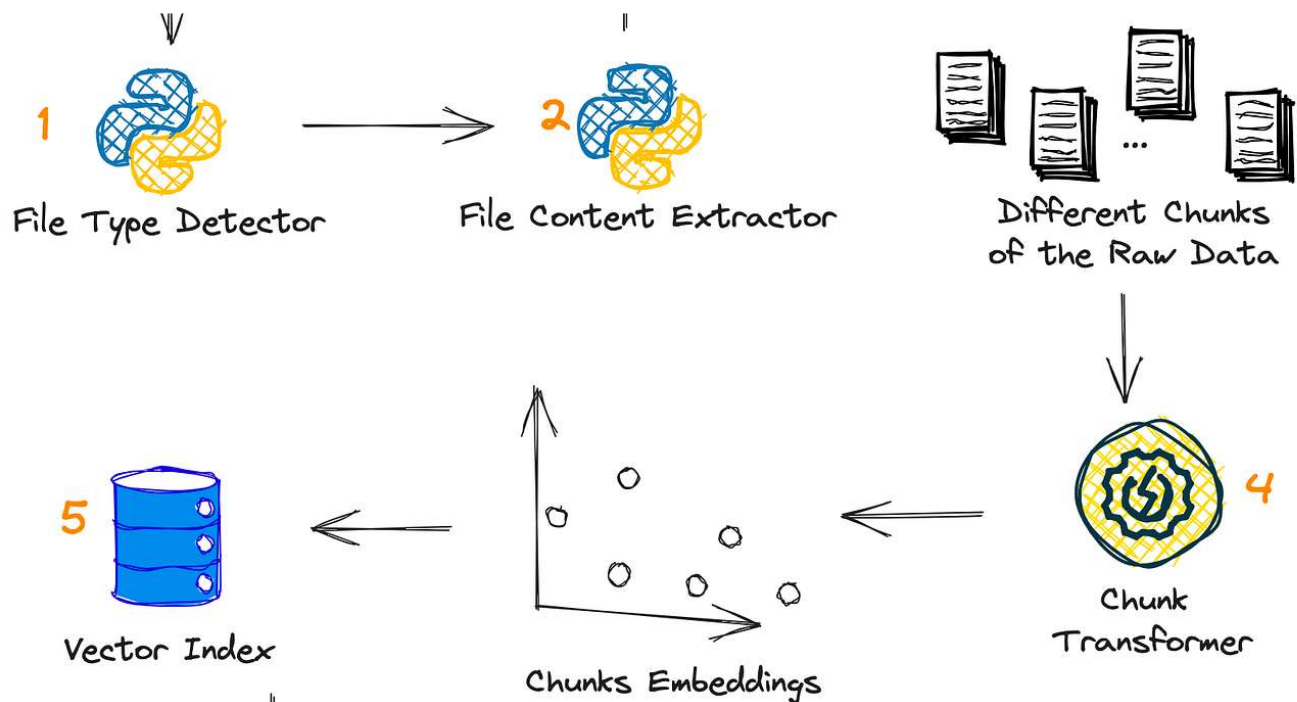
★ • 48 min read • Aug 1



3.4K



11



Zoumana Keita in Towards Data Science

How to Chat With Any File from PDFs to Images Using Large Language Models—With Code

Complete guide to building an AI assistant that can answer questions about any file

★ • 9 min read • Aug 5



1.2K



12





Gary Licht

What Blockchain misses about trust

An old school framework for understanding trust

5 min read · Nov 4, 2020



1



See all from Gary Licht

See all from Towards Data Science

Recommended from Medium

deepla

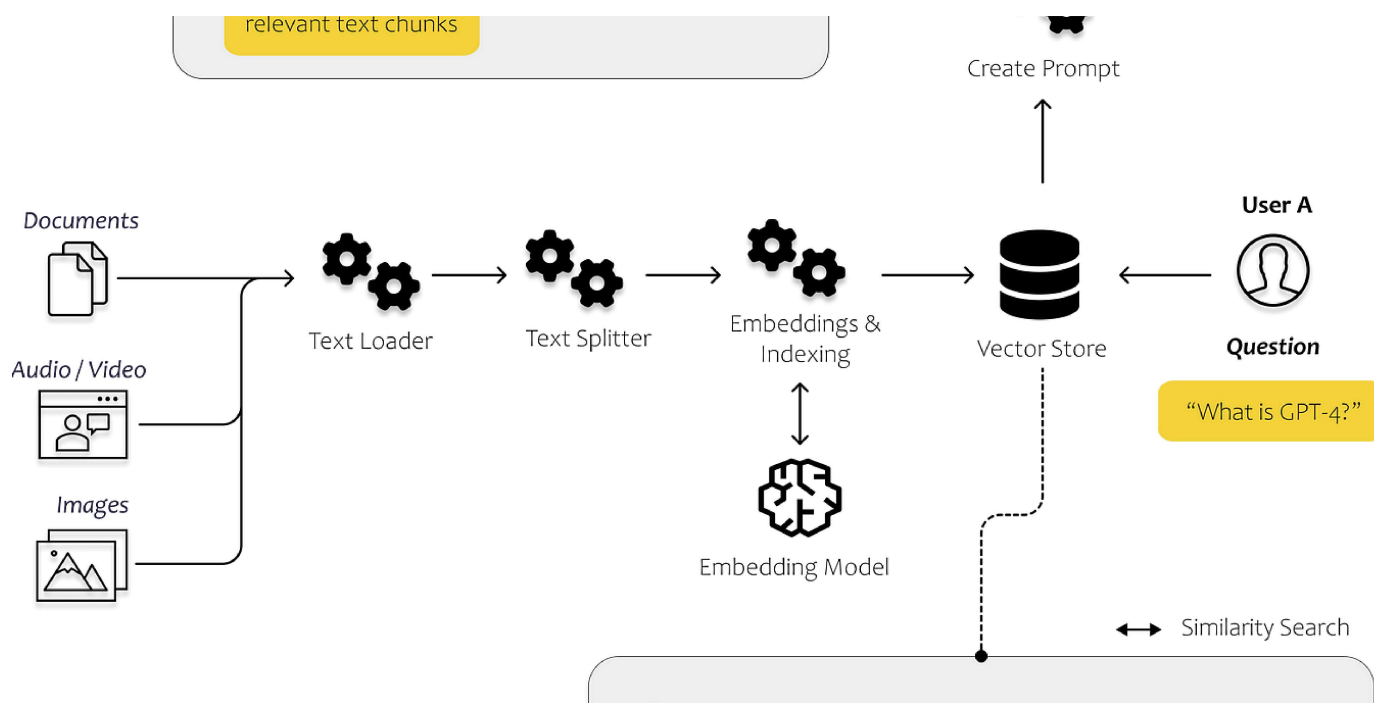
 Jerry Liu in Better Programming


LlamaIndex and Deep Lake for Financial Statement Analysis

(co-authored by Davit Buniatyan, co-founder/CEO of Activeloop, and Jerry Liu, co-founder/CEO of LlamaIndex)

9 min read · Apr 27

 267  3



 Dominik Polzer in Towards Data Science

All You Need to Know to Build Your First LLM App

A step-by-step tutorial to document loaders, embeddings, vector stores and prompt templates

★

• 26 min read

• Jun 21

 4.8K

 43



Lists



Predictive Modeling w/ Python

20 stories • 336 saves



Practical Guides to Machine Learning

10 stories • 380 saves



Natural Language Processing

568 stories • 188 saves

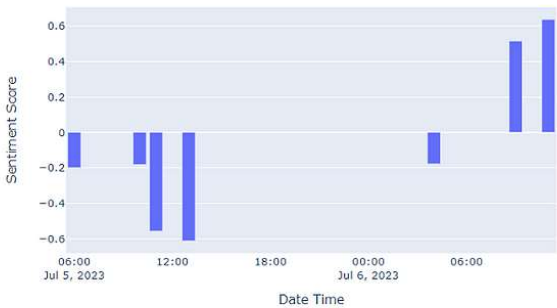


New_Reading_List

174 stories • 92 saves

Stock Sentiment Analysis for AAPL

AAPL Hourly Sentiment Scores



AAPL Price



The above chart averages the sentiment scores and price of AAPL stock. The table below gives each of the most recent headlines of the stock and the negative, neutral, positive and an aggregated sentiment score. The news headlines are obtained from Mboum Finance API.

Headline		Description	neg	neu	pos	Sentiment Score
Date Time						
2023-07-06 12:16	12 Best Major Stocks to Buy Now	In this piece, we will take a look at the 12 best major stocks to buy now. If you want to skip our analysis of the market that sees how major and mega cap stocks have dominated this year, head on over to 5 Best Major Stocks to Buy Now. 2023 has	0.000	0.588	0.412	0.6369



David Shilman in GoPenAI

Building News Sentiment and Stock Price Performance Analysis NLP Application with Python

4 min read · Jul 6



8



Wei-Meng Lee in Level Up Coding

Training Your Own LLM using privateGPT

Learn how to train your own language model without exposing your private data to the provider



· 8 min read · May 19

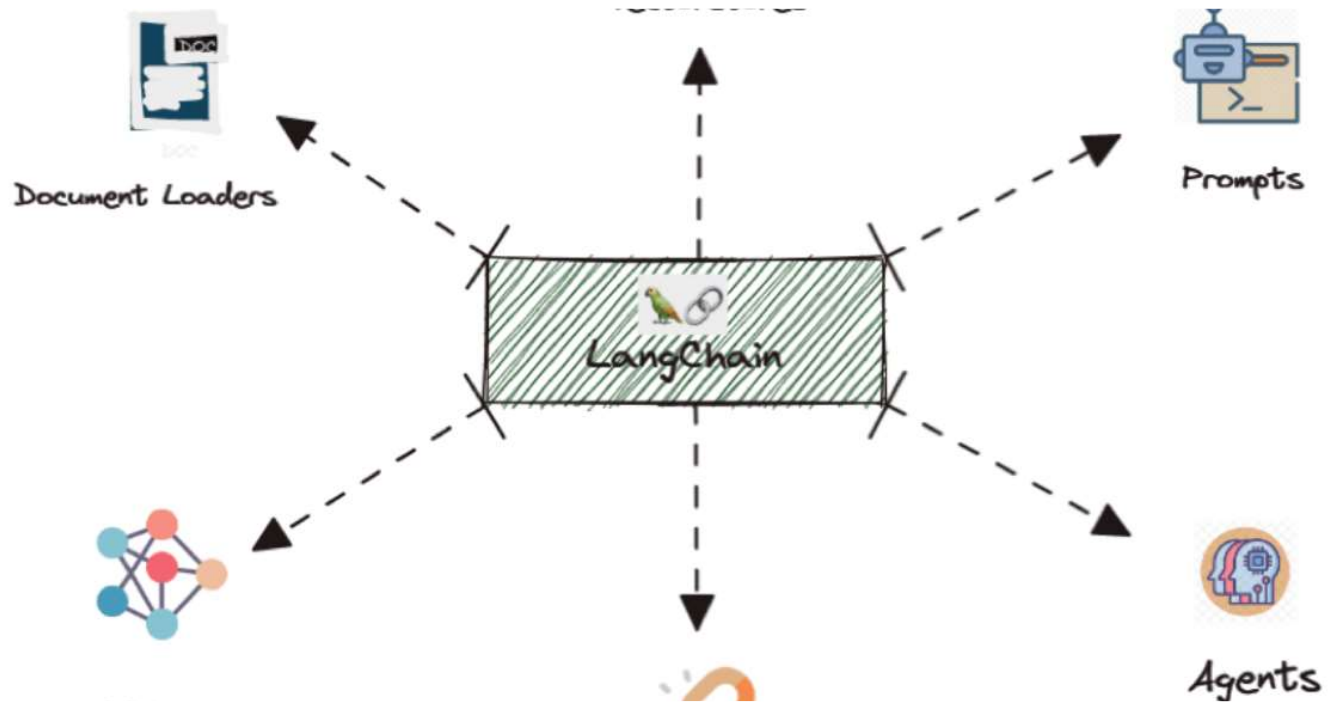


1.3K



12





 Zeeshan Malik

Connecting ChatGPT with your own Data using Llama Index and LangChain

In the last three months, there has been a rapid increase in the use of Large Language Models (LLMs) for a variety of applications, such as...

5 min read · Jun 11



35



2





Maximilian Vogel in MLearning.ai

The ChatGPT list of lists: A collection of 3000+ prompts, examples, use-cases, tools, APIs...

Updated Aug 20, 2023. Added prompt design courses, masterclasses and tutorials.

10 min read · Feb 7



7.5K



83



See more recommendations