Applied Textual Analysis with Python

Leveraging NLP for Financial Insights

Workshop Slides

Vitali Alexeev

Finance Department
UTS Business School
University of Technology Sydney, Australia

University of Otago, Dunedin, New Zealand
Department of Accountancy & Finance workshop
May 9, 2025

Table of Content

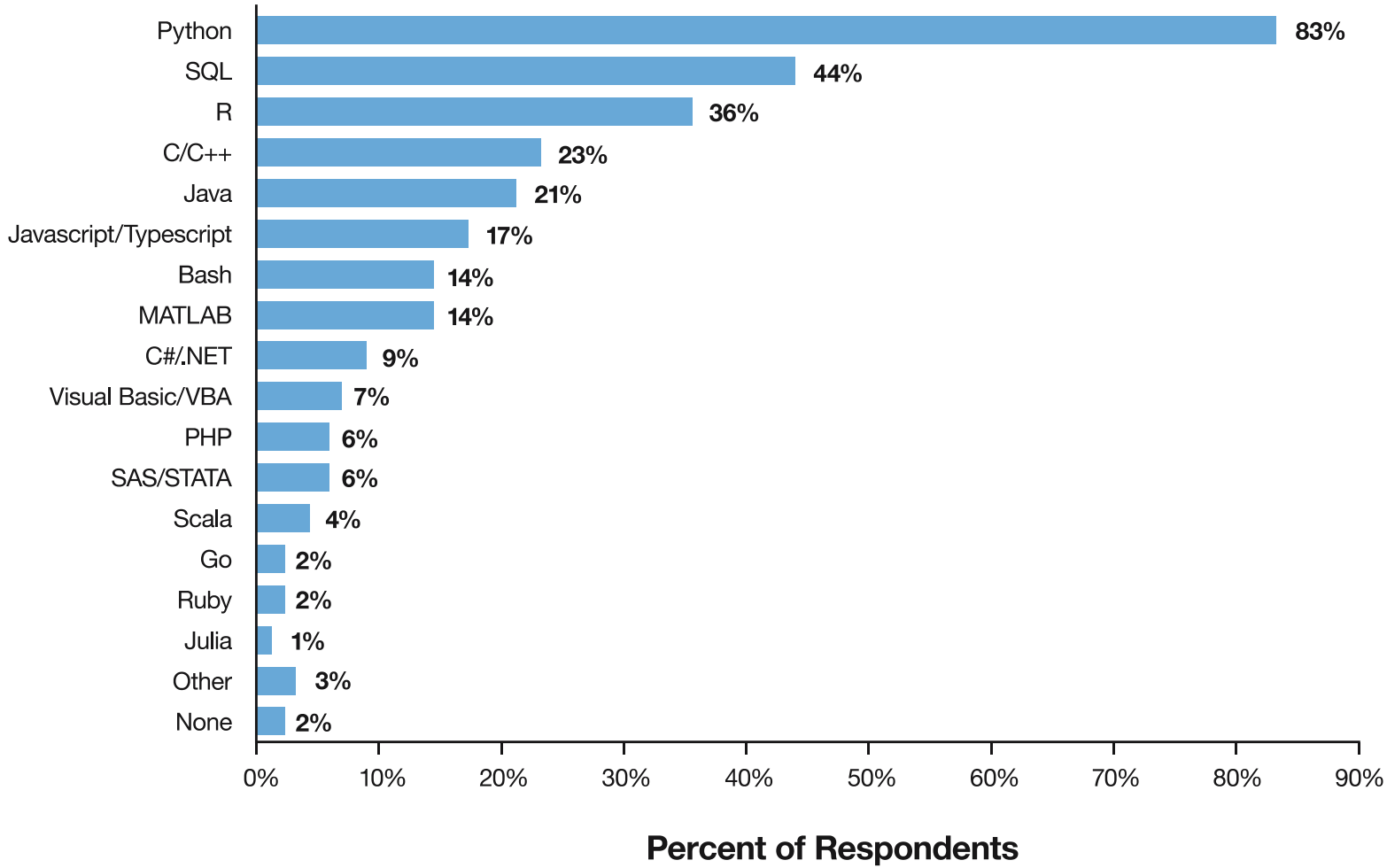
UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Resources

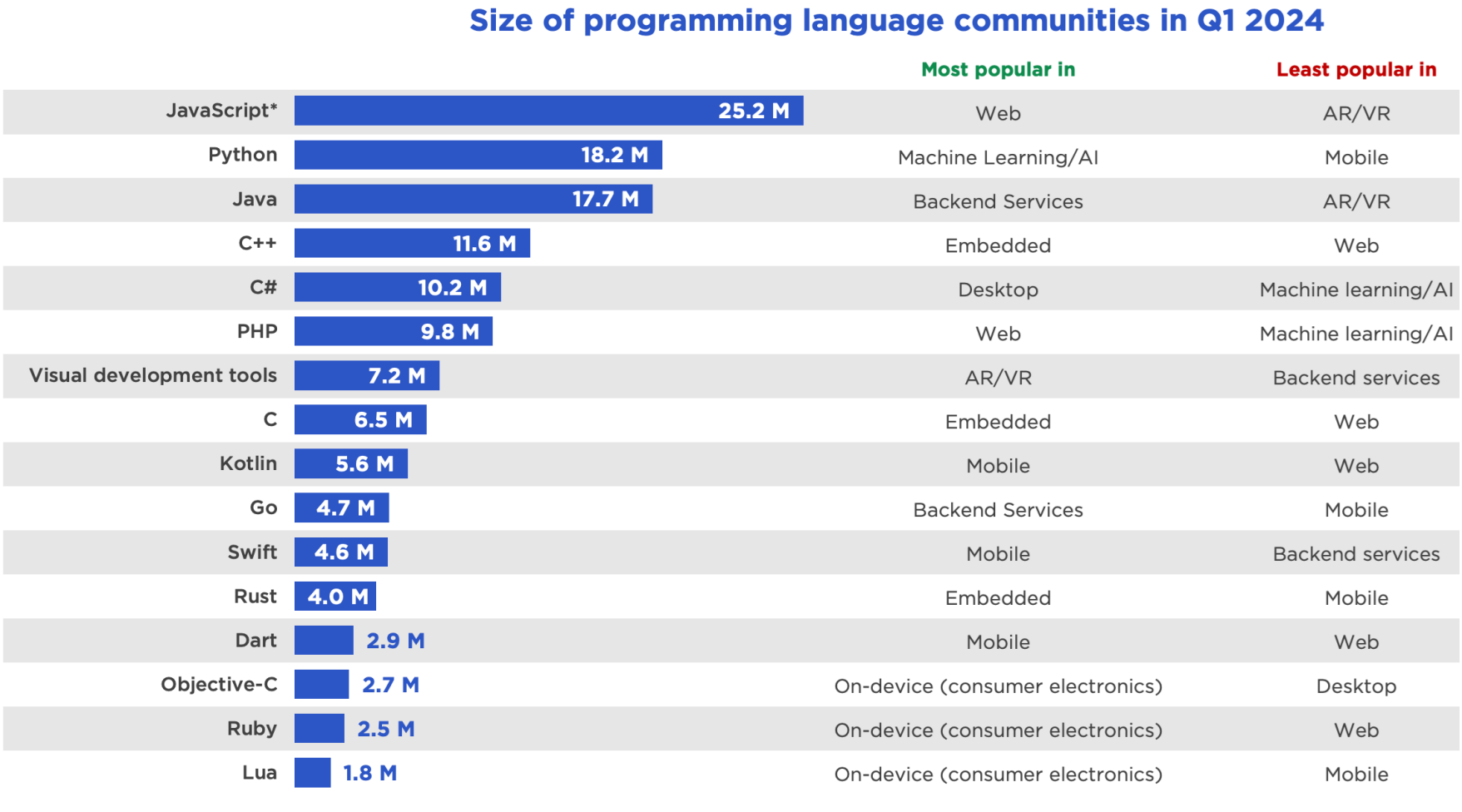
1. Repository of code, data and slides on ▸ Github
2. Python environment hosted on ▸ Binder
3. Latest transformer-based models on ▸ Hugging Face
4. GenAI for Coding: ChatGPT vs Claude ▸ Slides

The Programming Language Buffet

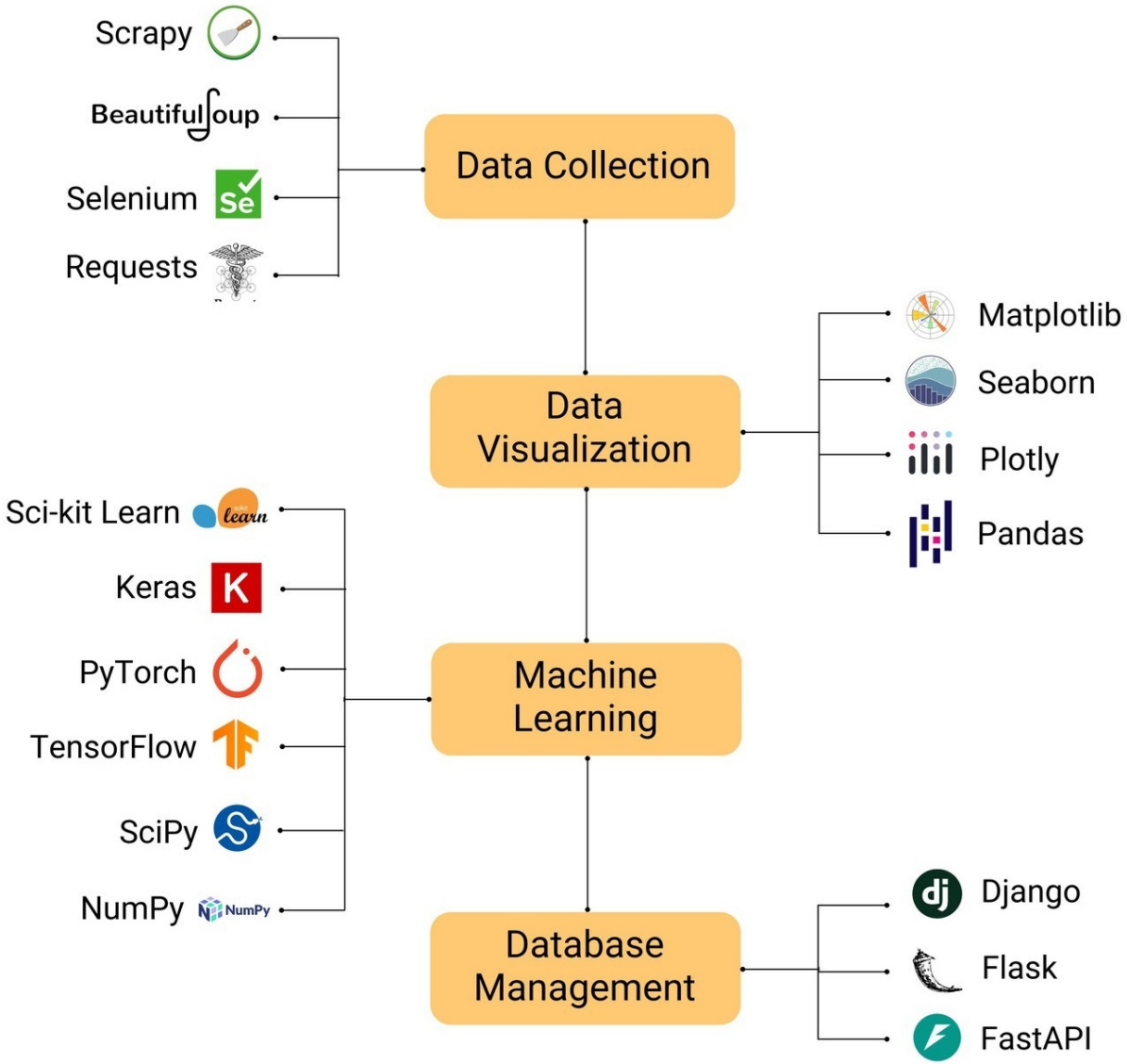
Data scientists have many options for programming languages to develop **machine learning** models. **Python** has become a popular choice.

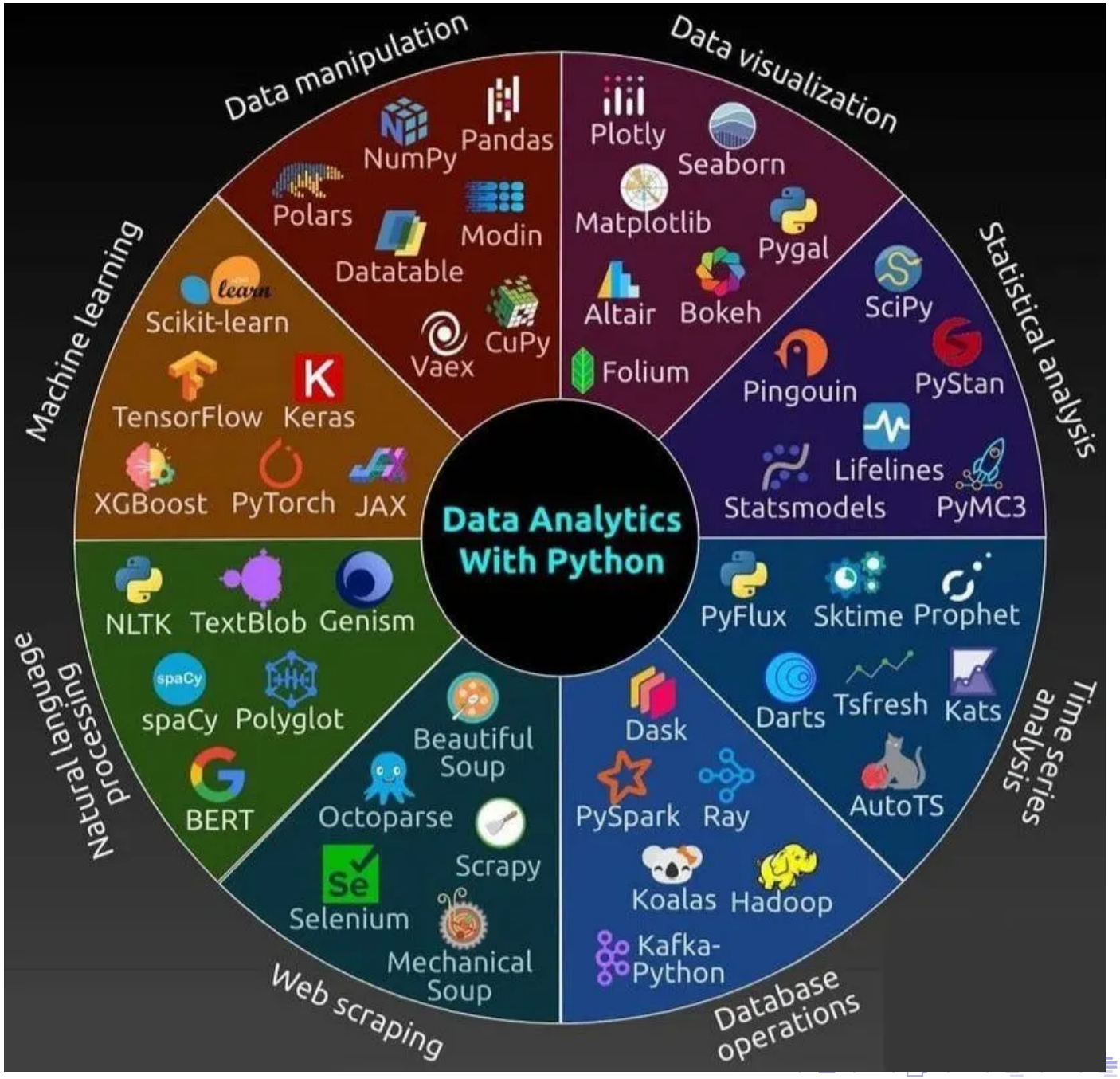


Programming Language Popularity by Application



Popular Python libraries




Data manipulation
NumPy
Pandas
Polars
Modin
Datatable
Vaex
CuPy
Data visualization
Plotly
Seaborn
Matplotlib
Pygal
Altair
Bokeh
Folium
Statistical analysis
SciPy
Pingouin
PyStan
Lifelines
Statsmodels
PyMC3
Time series analysis
PyFlux
Sktime
Prophet
Darts
Tsfresh
Kats
AutoTS
Database operations
Dask
PySpark
Ray
Koalas
Hadoop
Kafka-Python
Web scraping
Beautiful Soup
Octoparse
Scrapy
Selenium
Mechanical Soup
Natural language processing
NLTK
TextBlob
Genism
spaCy
Polyglot
BERT
Machine learning
Scikit-learn
TensorFlow
Keras
XGBoost
PyTorch
JAX
Data Analytics With Python

Running Python: Local vs. Cloud Environments

Local Installation

- **Pros**
  - Complete control over the environment and package versions.
  - No dependency on internet connectivity.
  - Ability to work with large datasets without upload/download limits.
- **Cons**
  - Requires setup and maintenance of the local environment.
  - Consumes local disk space and computational resources.
  - Potential for conflicts between packages or environments.

My recommendation:

ANACONDA.

The Cloud

- **Pros**
  - Easy sharing and collaboration on projects and notebooks.
  - No need to install Python locally; everything runs in the cloud.
  - Scalable resources depending on the plan.
- **Cons**
  - Limited free tier resources.
  - Dependent on internet connectivity.
  - May involve costs for premium features and resources.

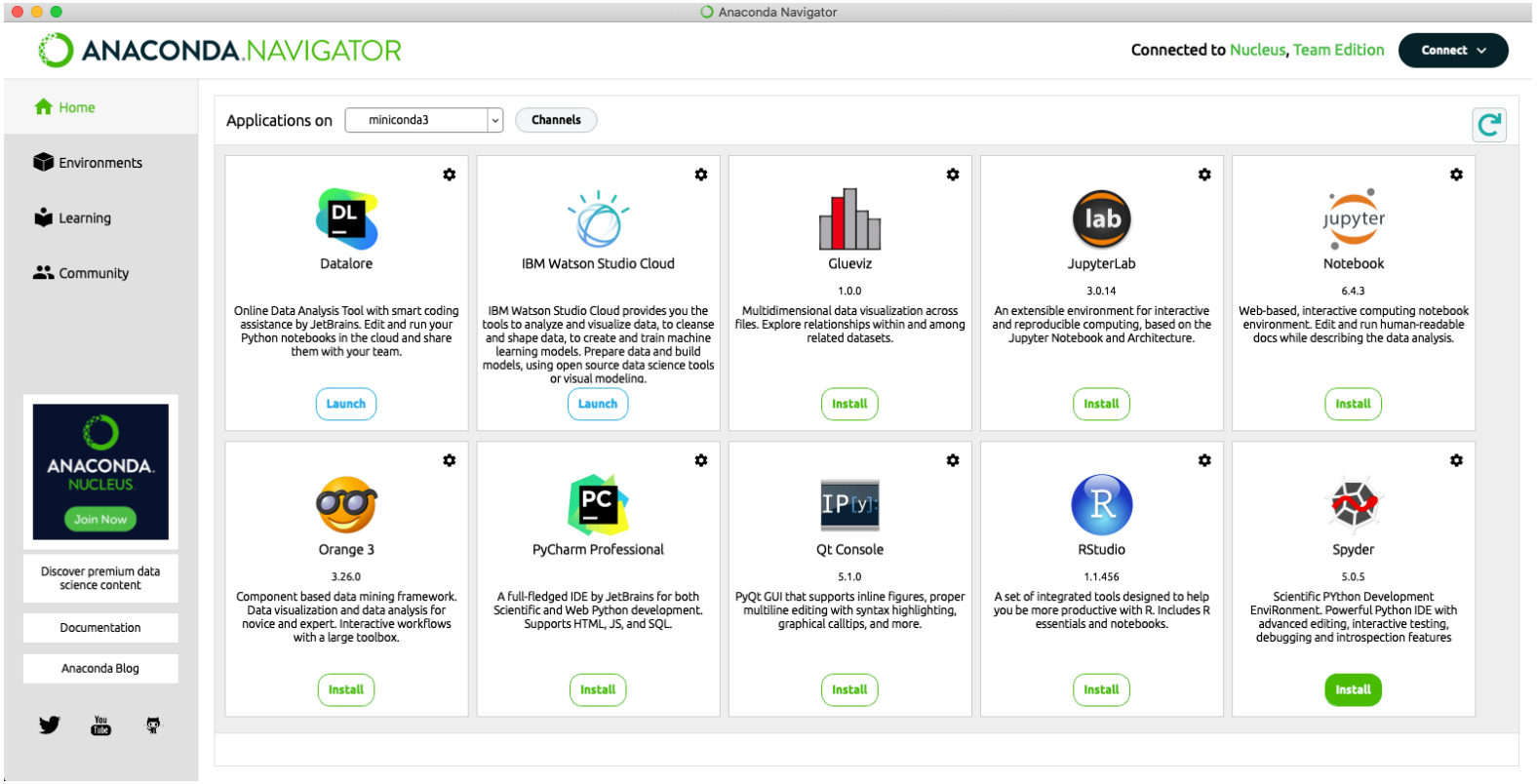
My recommendations:

ANACONDA CLOUD CO

UTS UNIVERSITY OF TECHNOLOGY SYDNEY

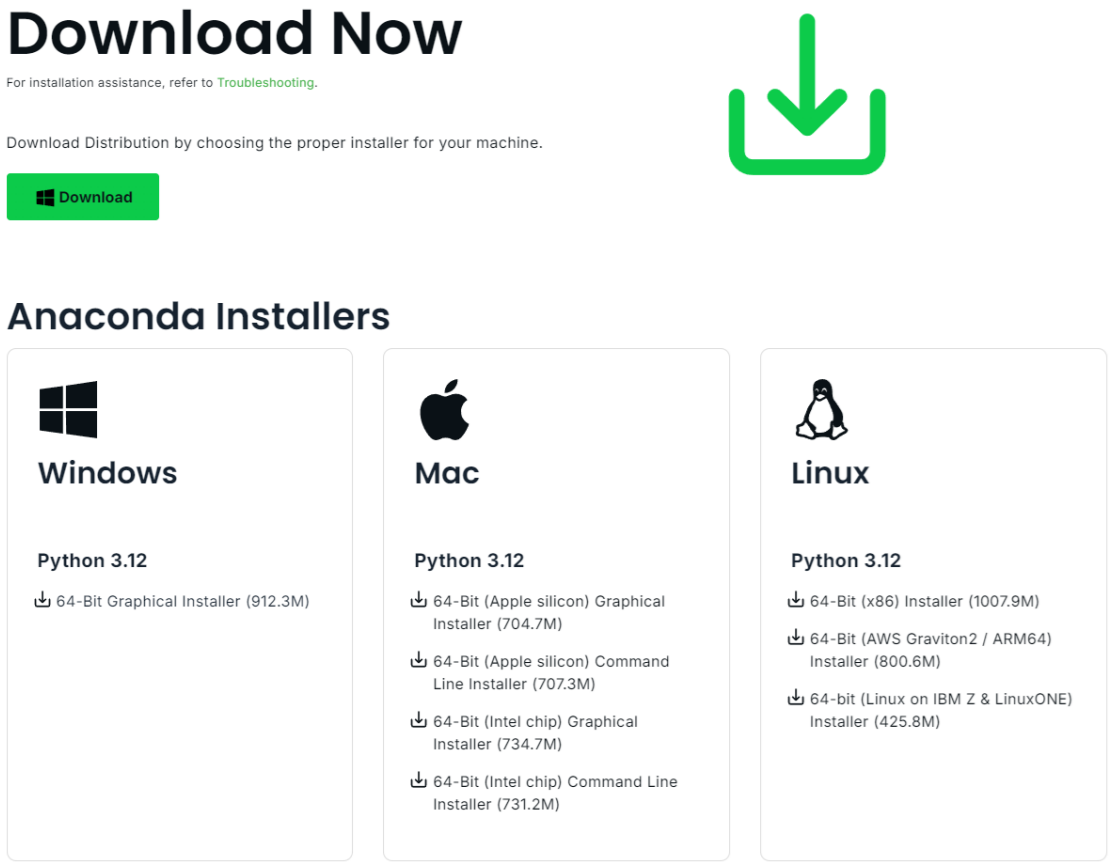
What is ANACONDA. ?

- Anaconda is a free and open-source distribution of Python and R.
- It simplifies package management and deployment.
- Commonly used in data science, machine learning, and scientific computing.
- We will use **Jupyter Notebook** or **JupyterLab**



Step 1: Download Anaconda

1. Visit the official Anaconda website: anaconda.com/products/distribution
2. Click on **Download** and select your operating system.

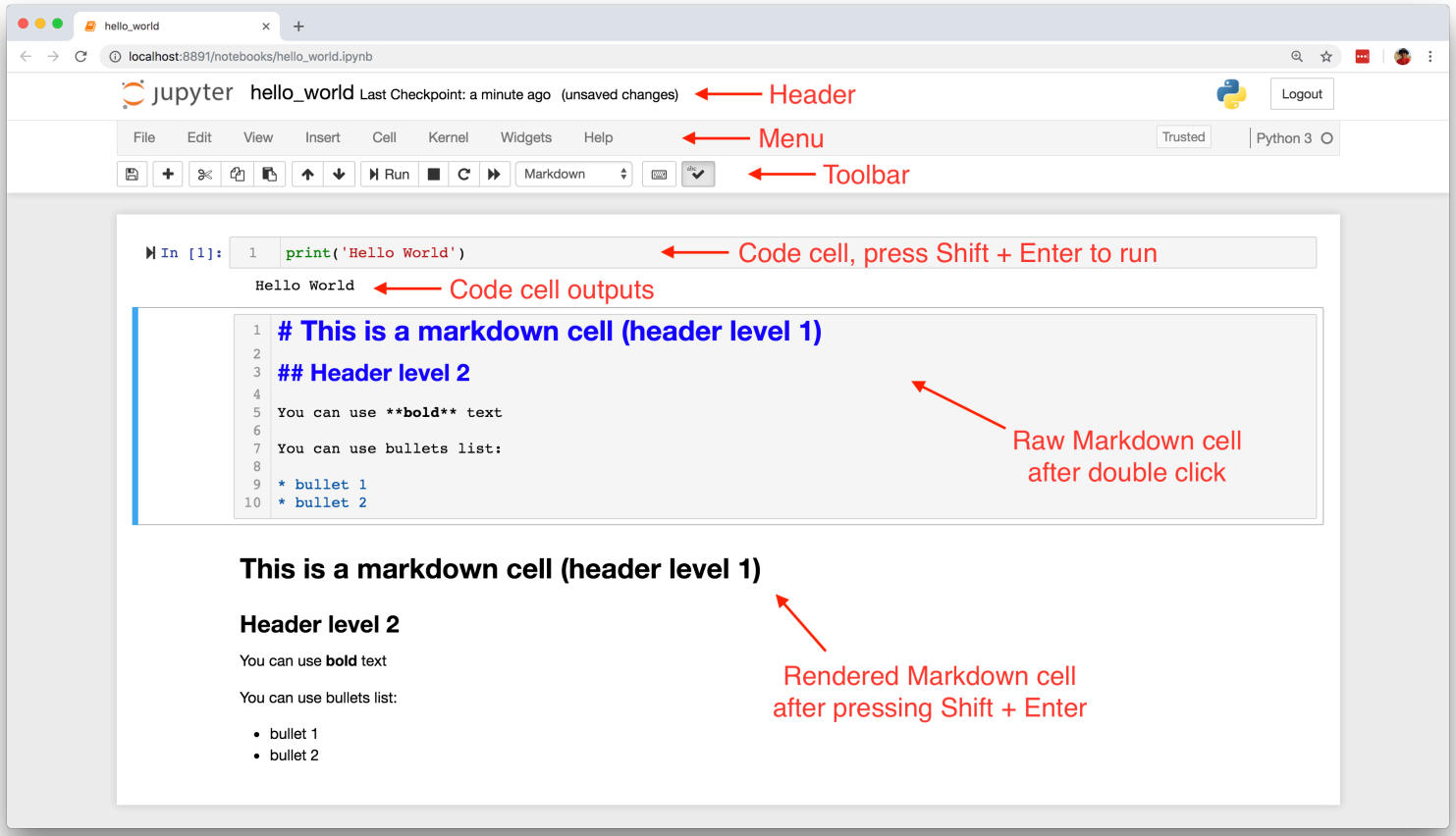


Step 2: Install Anaconda

1. Detailed installation instructions are available here and include Windows, macOS, and Linux.
2. Run the downloaded installer.
3. Follow the installation instructions on the screen.

Step 3: Using Anaconda

- Open the **Anaconda Navigator**.
- Launch **Jupyter Notebook** or **JupyterLab** for interactive computing.
- Use the **Anaconda Prompt** for command-line operations.



What is Anaconda Cloud?

- Anaconda Cloud is a robust platform for managing and sharing data and code.
- It supports Python with pre-built packages.
- Ideal for sharing environments with students.

Accessing Anaconda Cloud

1. Visit: Anaconda Cloud
2. Create an account or sign in.

What is Google Colab?

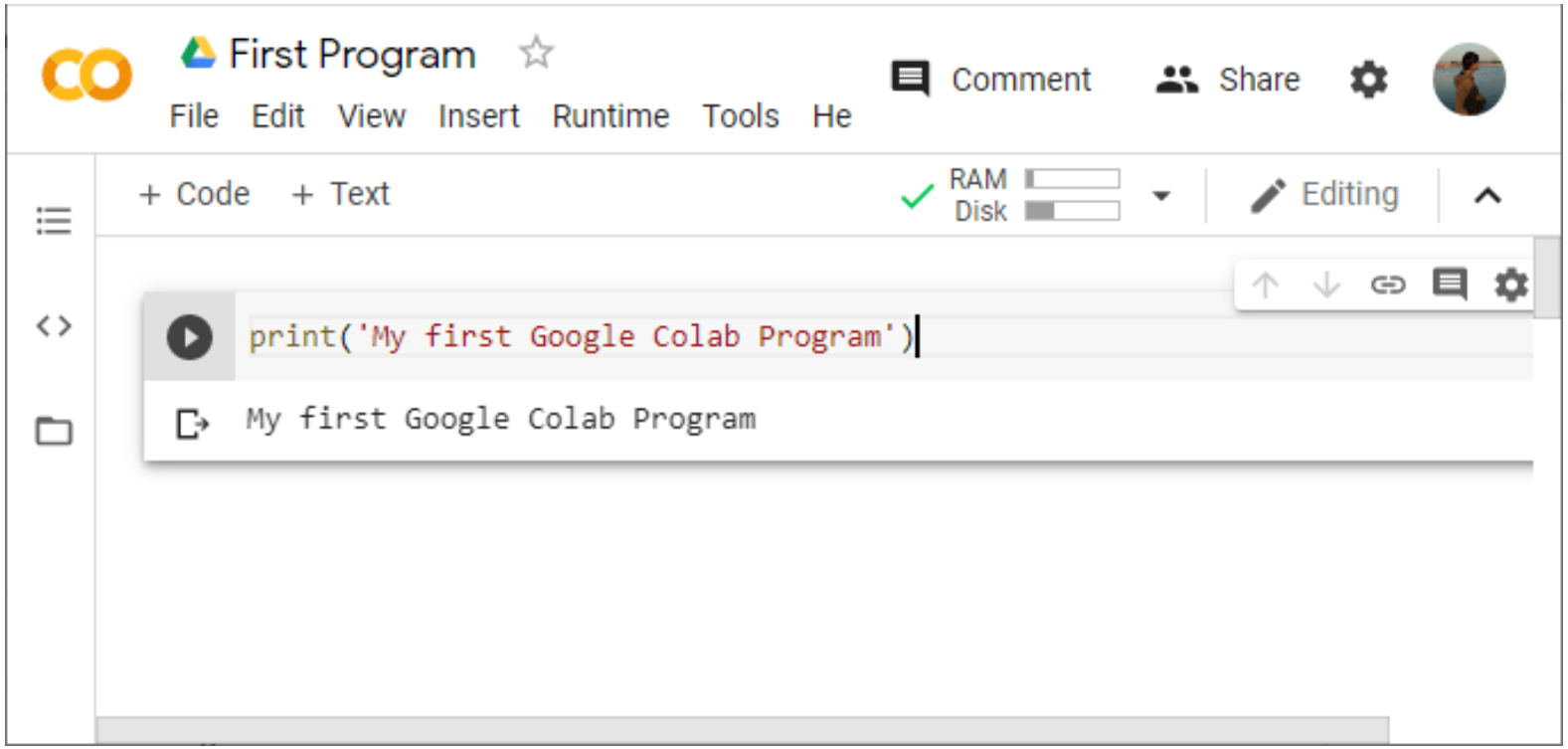
- Google Colab is a free cloud-based Jupyter notebook environment.
- It provides free GPU and TPU support.
- Ideal for collaborative work and sharing notebooks.

Accessing Google Colab

1. Visit: colab.research.google.com
2. Sign in with your Google account.
3. Create a new notebook by clicking **New Notebook**.

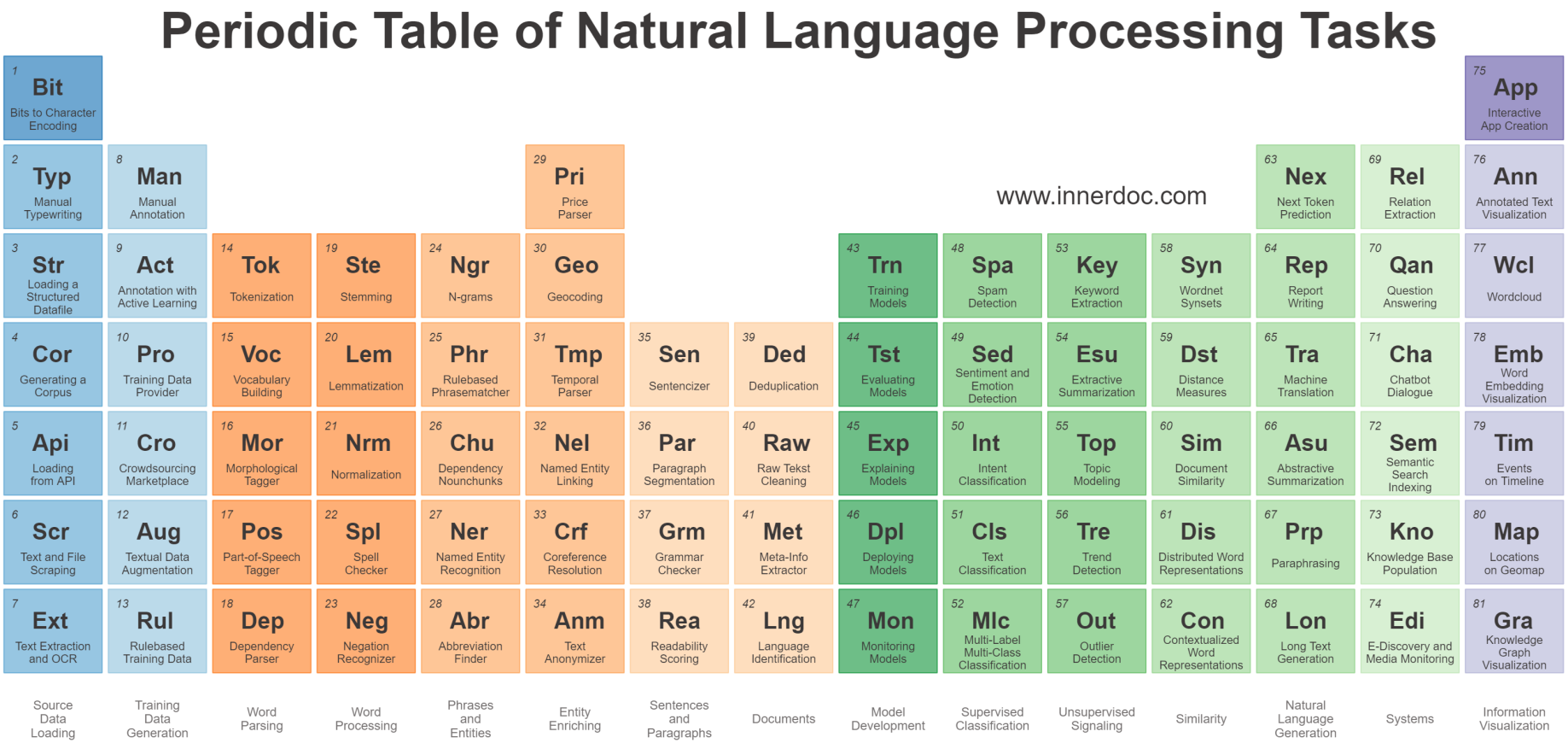
Running Code in Google Colab

- Google Colab uses the Python programming language.
- You can execute code cells by clicking the play button or pressing **Shift + Enter**.



Periodic Table of NLP Tasks

Source



1950s-1960s: Early Beginnings

- 1950s: Early attempts at computer-based linguistic analysis
- 1960s: Development of early language processing programs like ELIZA.

1970s-1980s: Rule-Based Systems

- 1970s: Rule-based expert systems for natural language understanding.
- 1980s: Introduction of early machine learning techniques for text analysis.

1990s: Sentiment Analysis

- 1990: Initial research in sentiment analysis, focusing on opinion polarity.

2000s: Rise of OSINT

- 2001: The 9/11 attacks highlight the need for Open Source Intelligence (OSINT).
- 2003: Launch of the OSINT-focused publication "Intelligence and National Security."

2000s: Web 2.0 and User-Generated Content

- 2004: Emergence of Web 2.0, leading to the proliferation of user-generated content.
- 2005: YouTube and Reddit are founded, platforms with significant textual data for analysis.

2010s: Sentiment Analysis and Social Media

- 2010: Expansion of sentiment analysis into social media data for brand monitoring.
- 2011: Introduction of sentiment analysis APIs by companies like AlchemyAPI (now IBM Watson).

2010s: Fact vs. Opinion

- 2014: Emergence of fact-checking organizations like PolitiFact and FactCheck.org.
- 2016: Fact-checking gains prominence during the U.S. presidential election.

2010s: Fake News and Misinformation

- 2016: Rise of fake news during the U.S. presidential campaign.
- 2017: Establishment of fact-checking initiatives and platforms to combat misinformation.

2010s: Deep Learning and NLP Advancements

- 2018: Advancements in deep learning, particularly in Natural Language Processing (NLP) with models like BERT.
- 2019: The introduction of OpenAI's GPT-2, which raised concerns about fake text generation.

2020s: Continued Growth

- 2020s: Ongoing development of AI-powered textual analysis tools, focusing on explainability, fairness, and privacy.
- 2021: Growing awareness of AI-generated deepfakes and their implications.

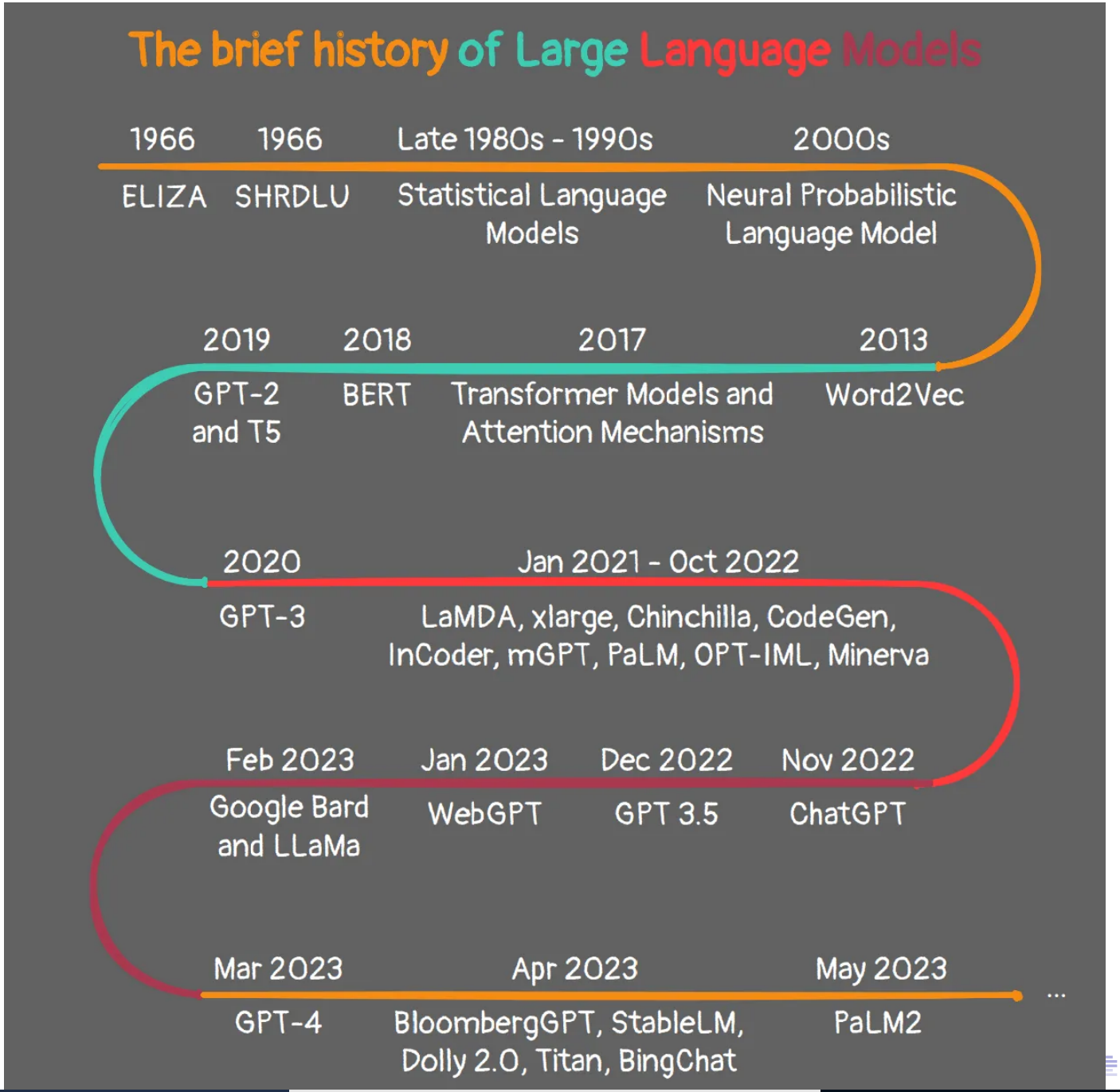
2020s: Ethical Considerations

- 2022: Focus on addressing ethical concerns in textual analysis, including bias mitigation and responsible AI practices.

Future Trends:

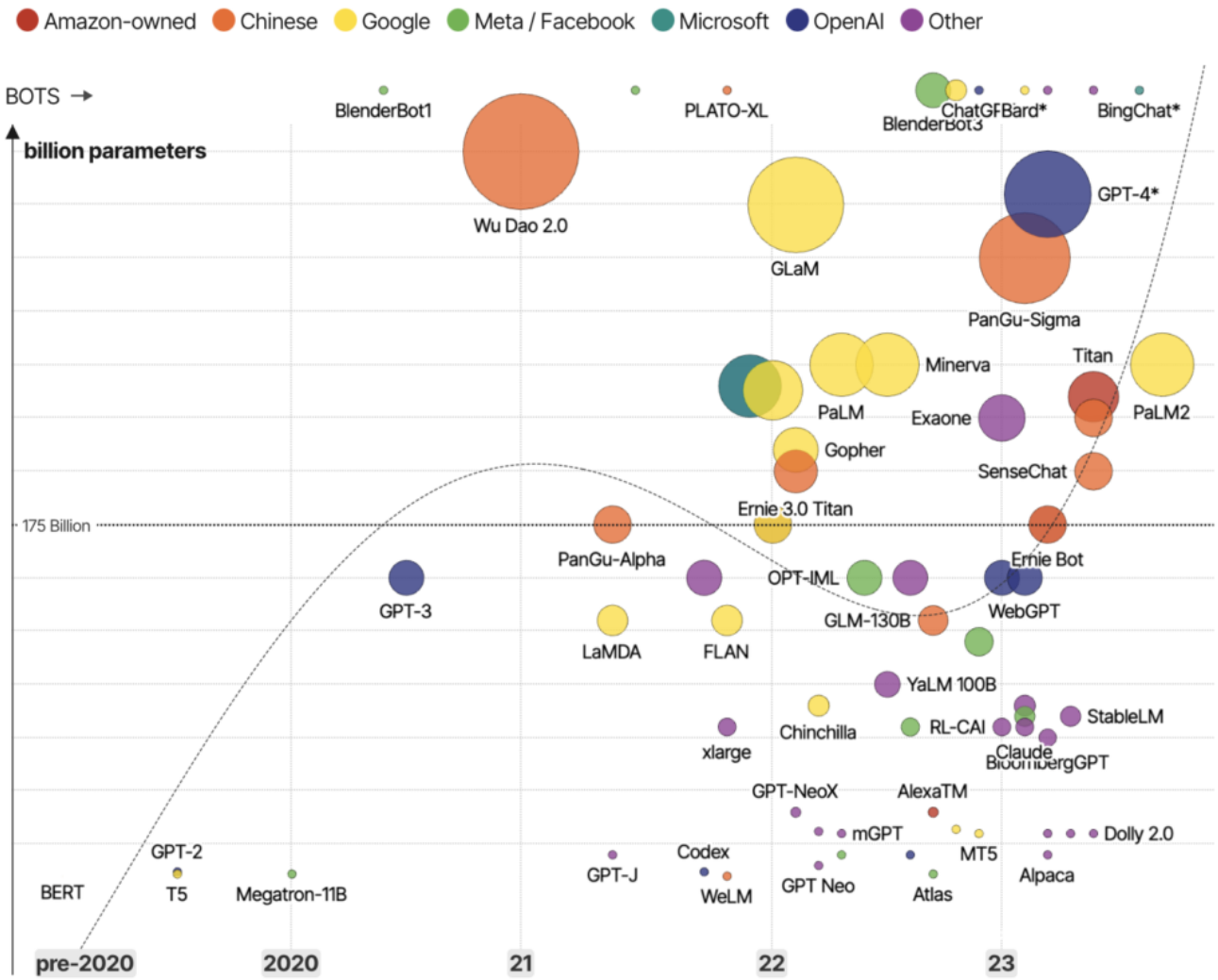
- Quantum computing's potential for enhancing textual analysis.

UTS UNIVERSITY OF TECHNOLOGY SYDNEY


The brief history of Large Language Models
1966
ELIZA
1966
SHRDLU
Late 1980s - 1990s
Statistical Language Models
2000s
Neural Probabilistic Language Model
2013
Word2Vec
2017
Transformer Models and Attention Mechanisms
2018
BERT
2019
GPT-2 and T5
2020
GPT-3
Jan 2021 - Oct 2022
LaMDA, xlarge, Chinchilla, CodeGen, InCoder, mGPT, PaLM, OPT-IML, Minerva
Nov 2022
ChatGPT
Dec 2022
GPT 3.5
Jan 2023
WebGPT
Feb 2023
Google Bard and LLaMa
Mar 2023
GPT-4
Apr 2023
BloombergGPT, StableLM, Dolly 2.0, Titan, BingChat
May 2023
PaLM2
...

Evolution of Large Language Models (LLMs)

▸ Explore Interactively



David McCandless, Tom Evans, Paul Barton
Information is Beautiful // May 2023

source: news reports, LifeArchitect.ai
* = parameters undisclosed // see the data

Lying CEOs

David F. Larcker and Anastasia A. Zakolyukina (2012). "Detecting Deceptive Discussions in Conference Calls". In: *Journal of Accounting Research* 50.2, pp. 495–540

- Analysed the language used by CEOs and CFOs talking to analysts in conference calls about earnings announcements.
- Contrasted with subsequent events:
  - discovery of accounting irregularities
  - restatements of earnings
  - changes of accountants
  - exit of CEO and/or CFO
- Retrospectively identified who was telling the truth or not.
- Linguistic-based classification model: A training corpus of transcripts which can be labelled as "true" or "deceptive".

Deceptive vs. Truthful text

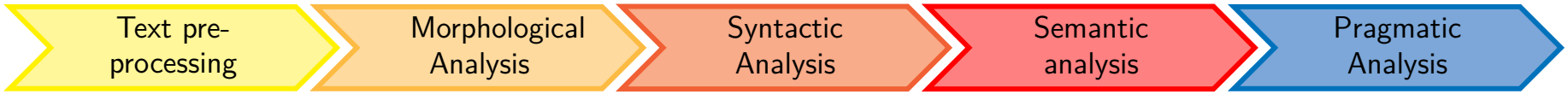
Liars

- Liars tend to use **more emotion** words, **fewer first person** ("I", "we"), **more negation** words, and **more motion** verbs ("lead", "go").
- Possible that liars **exaggerate** certainty and positive/negative aspects, and **do not associate** themselves closely with the content.

Truthtellers

- Truthtellers use **more self references**, **exclusive** words ("except", "but"), **tentative** words ("maybe", "perhaps") and **time related** words.
- Truthtellers are **more cautious**, accept **alternatives**, and **do associate** themselves with the content.

How Can You Tell When A CEO Is Lying? ▸ SMH news and ▸ NPR news



Different levels of language processing—each focuses on a different aspect of how language is structured and interpreted.

Analysis Type	Focus	Example
Text Preprocessing	Cleaning and normalizing raw text. Serves as a foundation for further linguistic processing.	*The stock market is going up!!!* → stock market going up
Morphological	Word structure (root words, prefixes, suffixes, inflections, and derivations)	*Running* → run + -ing
Syntactic	Sentence structure, grammar, parts of speech (nouns, verbs, etc.)	*She eats apples.* → Subject-Verb-Object structure
Semantic	Meaning of words & sentences, word sense disambiguation, synonymy, antonymy	*Apple* → a fruit or a company, based on context
Pragmatic	Contextual meaning, speaker intent, ambiguity, sarcasm, politeness	*It's cold in here.* → Could imply a request to close a window

Case Folding

Case folding is the process of converting all text into **lowercase** to ensure uniformity and reduce redundancy in NLP. It helps in normalizing words and improving text analysis efficiency.

- **Eliminates Redundant Variants**: "Apple", "apple", and "APPLE" are treated as the same word.
- **Reduces Vocabulary Size**: Helps models generalize better without treating uppercase and lowercase words separately.
- **Improves Search & Matching**: In tasks like information retrieval and text classification, words are compared case-insensitively.

When NOT to Use Case Folding?

1. Named Entity Recognition (NER) → "Apple" (company) vs. "apple" (fruit).
2. Sentiment Analysis → "GREAT" might indicate stronger sentiment than "great".
3. Acronyms & Abbreviations → "AI" (Artificial Intelligence) vs. "ai" (meaningless).

Stop-word Removal

Stop words removal is a text preprocessing technique where commonly used words (such as "the," "is," "and," "in," etc.) are removed from text data.

- These words typically do not carry significant meaning and are removed to reduce dimensionality and improve processing efficiency in NLP tasks.
- *The stock market is going up significantly in recent days.* → Stock market going up significantly recent days.

When Should You Keep Stop Words?

1. Sentiment Analysis → Words like "not," "never," and "no" can change meaning.
2. Chatbots & Conversational AI → Stop words contribute to natural dialogue.
3. Named Entity Recognition (NER) → Contextual words may be important.

Lowercasing and Stop-word Removal

Raw Data

Cleaned Data

Tokenization (Segmentation)

Definition: Tokenization is the process of breaking text into smaller units (tokens) for further NLP analysis.

Tokenization (Segmentation)

Types of Tokenization:

1. **Word Tokenization**: Splitting text into words.
2. **Sentence Tokenization**: Splitting text into sentences.
3. **Subword Tokenization**: Splits words into smaller meaningful subunits for handling rare or unseen words. unhappiness → "un" + "happiness"
4. **Character Tokenization**: Splitting into individual characters.

Why is Tokenization Important?

- **Facilitates Text Processing** → Converts unstructured text into structured tokens.
- **Helps in Feature Extraction** → Tokens serve as input for models.
- **Enables Efficient NLP Tasks** → Necessary for stemming, lemmatization, parsing, and embedding generation.
- **Handles Different Languages** → Different strategies for English (whitespace-based) vs. Japanese/Chinese (character-based)

Language dependent issues in NLP

Tokenizing in English is much simpler:
English: *"insurance companies that provide legal protection"*

It is harder to do in German:
German: *"Rechtsschutzversicherungsgesellschaften"*

Note: English has spaces, whereas Chinese, Japanese, and Thai lack them, requiring segmentation models.

Subword Tokenization: Using Byte Pair Encoding (BPE)/WordPiece

- Splits words into smaller meaningful subunits for handling rare or unseen words.
- unhappiness → "un" + "happiness"
- electroencephalography → "electro" + "encephalo" + "graphy"
- Rechtsschutzversicherungsgesellschaften → 'Rechtsschutz' (legal protection) + 'versicherungs' (insurance) + 'gesellschaften' (companies)]

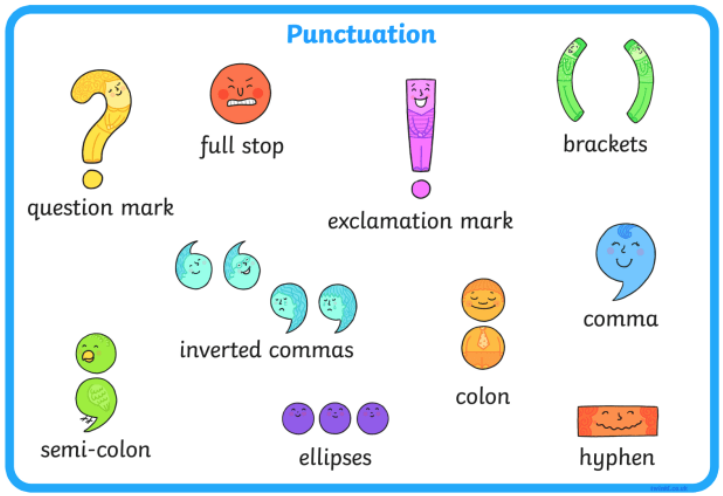
Longest name

Good luck tokenizing this!

Near Porangahau in Hawke's Bay, NZ is an unassuming hill known by the name in the photo, which translates into English as *"the place where Tamatea, the man with the big knees, who slid, climbed and swallowed mountains, known as 'landeater', played his flute to his loved one."* Locals simply call it Taumata Hill.

Punctuation Handling

Punctuation handling: removal, retention, or transformation of punctuation marks in NLP tasks. Helps clean and standardize text before further analysis.



When to Keep Punctuation?

- **Sentiment Analysis** → "Great!!!" vs. "Great." conveys different intensity.
- **Named Entity Recognition (NER)** → Punctuation can indicate abbreviations ("U.S.").
- **Conversational AI & Chatbots** → Punctuation helps understand tone and structure.
- **Legal & Financial Texts** → Some punctuation (like commas in numbers) is meaningful.

Stemming vs. Lemmatization

Stemming

· Reduces a word to its stem.

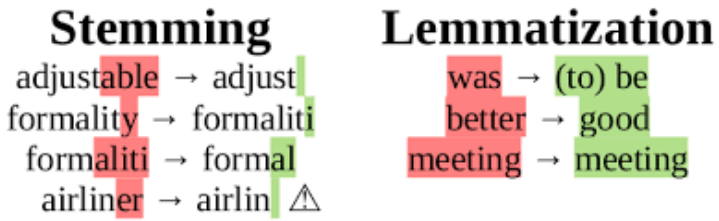
· Less readable by human but makes text more comparable across observations.

Lemmatisation

· Process of determining the lemma of a word based on its intended meaning.

· Identifies the intended part of speech and word meaning within the larger surrounding context.

Note: "**Meeting**" can be a **noun** or a **verb** ("to meet") depending on the context. E.g., "in our last **meeting**" or "We are **meeting** again tomorrow". Lemmatisation (unlike stemming) attempts to select the correct lemma depending on the context.



Feature	Stemming	Lemmatization
Approach	Rule-based	Dictionary-based
Speed	✓ Fast	✗ Slower
Accuracy	✗ Less accurate	✓ More accurate
Example	"Studies" → "studi"	"Studies" → "study"
Use Case	Search engines, large datasets	Chatbots, sentiment analysis

Part-of-Speech (POS) Tagging

▸ Interactive

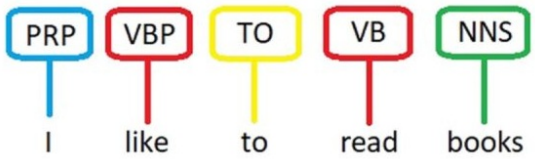
POS tagging assigns grammatical categories to words in a sentence, like nouns, verbs, adjectives, etc., helping NLP models understand syntax and structure.

- Helps disambiguate word meanings (e.g., "book" as a noun vs. a verb).

Common POS Tags:

▸ list of abbreviations

- **Noun (NN)**: *"dog," "car," "happiness"*
- **Verb (VB)**: *"run," "is," "calculate"*
- **Adjective (JJ)**: *"beautiful," "large"*
- **Adverb (RB)**: *"quickly," "very"*



POS Tagging Methods:

- **Rule-based Tagging**: Uses linguistic rules (if a word follows "the", it's likely a noun).
- **Statistical Tagging**: Uses probabilities from corpora (e.g., Hidden Markov Models).
- **Neural Tagging**: Deep learning models (e.g., LSTMs, Transformers) predict POS tags.

Chunking (Shallow Parsing)

Chunking, also known as **shallow parsing**, groups words into meaningful phrases (chunks) based on their Part-of-Speech (POS) tags.

- Identifies **phrases** such as noun phrases (NP), verb phrases (VP), and prepositional phrases (PP).
- **Common Chunk Types:**
  - **Noun Phrase (NP)**: *"The quick brown fox"*
  - **Verb Phrase (VP)**: *"jumps"*
  - **Prepositional Phrase (PP)**: *"over the lazy dog"*

Advantages:

- ✓ Provides higher-level structure than POS tagging.
- ✓ Useful for named entity recognition (NER) and syntactic parsing.
- ✓ Improves information extraction and question answering.

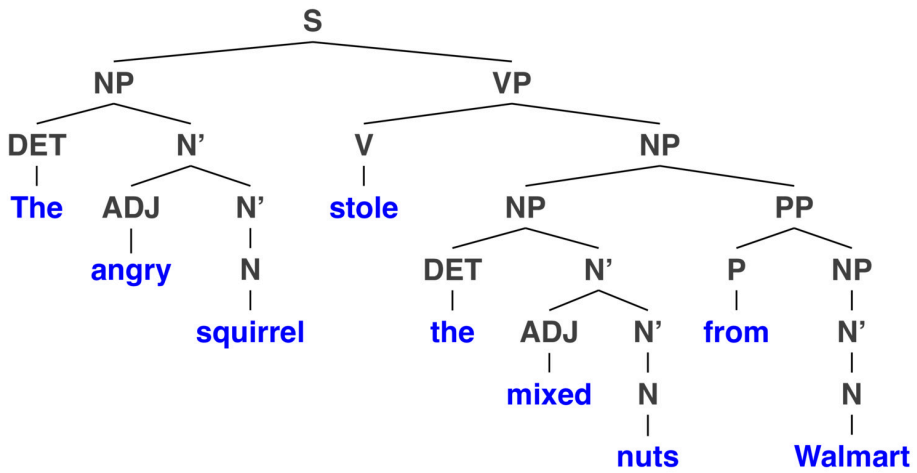
Limitations:

- ✗ Cannot handle long-range dependencies like full parsing.
- ✗ Rule-based chunking requires manual rule tuning.
- ✗ ML-based chunking needs labeled training data.

Constituency Parsing

Breaks sentence into nested phrases (syntax tree).

- Uses phrase structure rules.
- **Goal**: Identify how words group together into phrases (constituents) such as **noun phrases (NP)**, **verb phrases (VP)**, and **sentence (S)**.

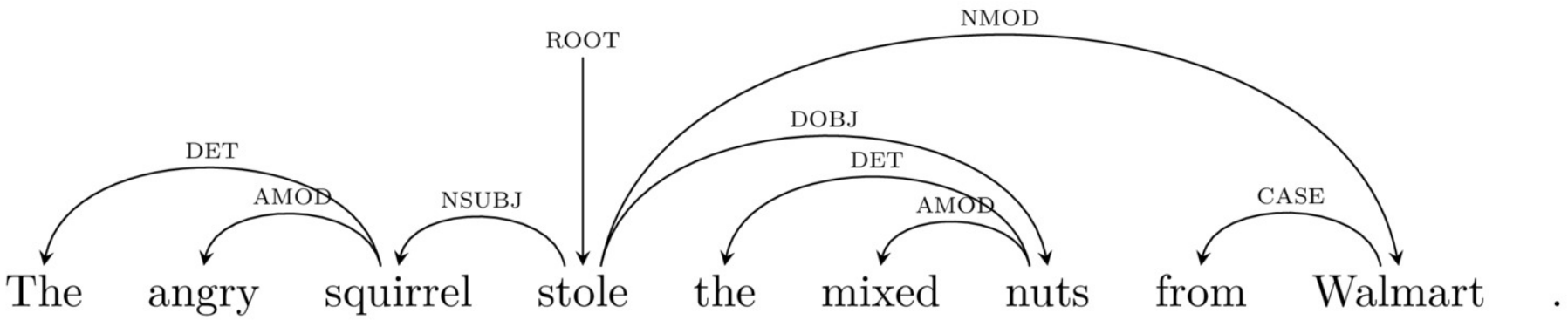


- **Machine Translation**: Helps preserve syntax when translating between languages.
- **Text Summarization**: Recognizes core sentence structures to extract important content.
- **Sentiment Analysis**: Parses sentiment-bearing phrases for better interpretation.

Dependency Parsing

Represents grammatical relationships between words

- Each word depends on a "head" word, forming a directed tree.



- NSUBJ - Nominal Subject: the 'do-er' of the verb
- DOBJ - Direct Object: object receiving the action
- 'Indirect Object' - Often used for recipients
- Modifiers - Words that make the meaning of each word more specific

Parsing: Constituency and Dependency

Parsing analyses **sentence structure** by breaking it down into components and their relationships.

- **Constituency Parsing:**
  - **Breaks sentence into nested phrases (syntax tree).**
  - Uses phrase structure rules.
- **Dependency Parsing:**
  - **Represents grammatical relationships between words.**
  - Each word depends on a "head" word, forming a directed tree.
- Parsing is crucial for **semantic analysis**, translation, and question answering.

Advantages:

- ✓ Captures sentence structure and relationships.
- ✓ Useful for **syntax-aware NLP tasks** like machine translation.
- ✓ Helps resolve **ambiguity** in language processing.

Limitations:

- ✗ Computationally expensive for large sentences.
- ✗ Struggles with ambiguity and informal text.
- ✗ Requires large annotated datasets for accuracy.

Word Sense Disambiguation (WSD)

Determines the correct meaning of a word based on its **context**.

- Resolves **lexical ambiguity** when a word has multiple meanings.
- **Examples:**
  - **"Bank"**: *"I deposited money at the bank"* vs. *"The boat reached the river bank"*.
  - **"Apple"**: *"I ate an apple"* vs. *"Apple released a new iPhone"*.
- **WSD Methods:**
  - **Dictionary-based**: Uses lexical databases (e.g., WordNet).
  - **Supervised Learning**: Trains models on labeled sense-annotated data.
  - **Unsupervised Learning**: Clusters word senses from raw text.
  - **Neural Methods**: Uses deep learning (e.g., BERT, Transformers) for contextual sense detection.

Advantages:

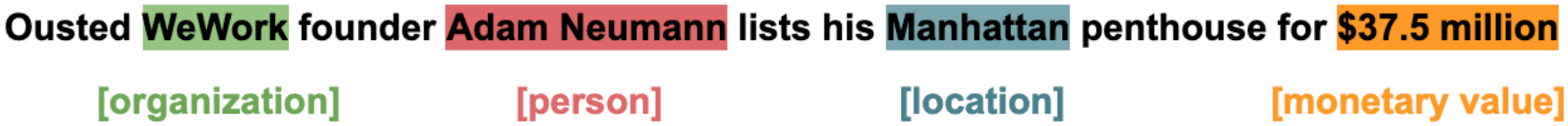
- ✓ Improves machine translation and information retrieval.
- ✓ Enhances chatbots and search engines.
- ✓ Provides better contextual understanding in NLP applications.

Limitations:

- ✗ Ambiguity is context-dependent and complex.
- ✗ Supervised methods require large labeled datasets.
- ✗ Unsupervised methods may lack accuracy.

Named Entity Recognition (NER)

NER identifies and classifies key entities in text into predefined categories.



- Detects and assigns labels like Person, Organization, Location, Date, etc.
- NER Methods:
  - **Rule-based**: Uses handcrafted rules and regular expressions.
  - **Machine Learning**: Trains classifiers on labeled entity datasets.
  - **Deep Learning**: Uses BERT and Transformers for contextual entity recognition.

Advantages:

- ✓ Automates information extraction from text.
- ✓ Useful for chatbots, search engines, and financial analysis.
- ✓ Helps in news classification, healthcare, and legal document processing.

Limitations:

- ✗ Struggles with ambiguous names (e.g., "Apple" as a company vs. fruit).
- ✗ Rule-based methods lack scalability.
- ✗ ML models require large labeled datasets for training.

Named Entity Recognition (NER)

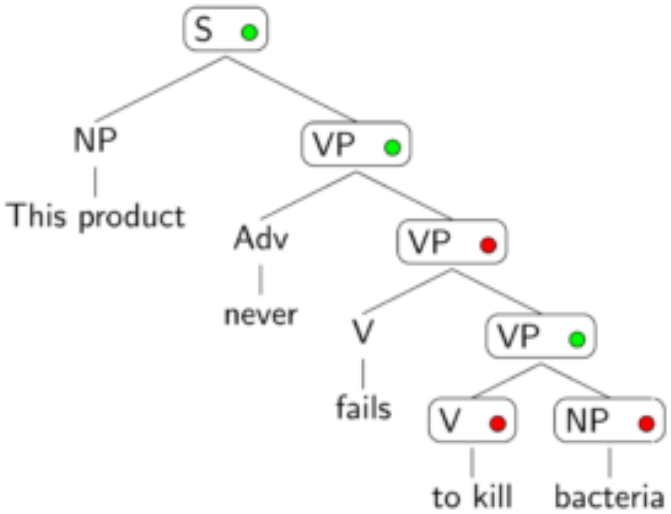
Approaches and Models

Jobs Person and Wozniak Person co-founded Apple Company in 1976 Date to sell Wozniak's Apple I Product personal computer Device. Together the duo gained fame and wealth a year later for the Apple II Product, one of the first highly successful mass-produced personal computers Device. Jobs Person saw the commercial potential of the Xerox Alto Product in 1979 Date, which was mouse-driven Device and had a graphical user interface Interface (GUI) Interface. This led to development of the unsuccessful Apple Lisa Product in 1983 Date, followed by the breakthrough Macintosh Product in 1984 Date, the first mass-produced computer Device with a GUI Interface.

Popular NER Models:

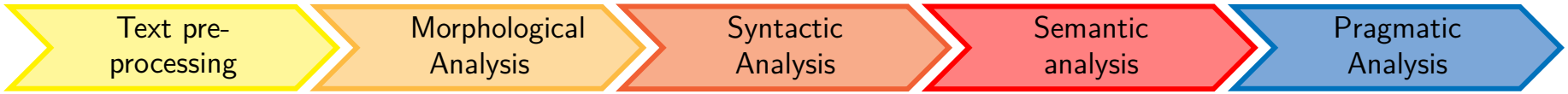
- **spaCy NER**: Fast and efficient, widely used in industry.
- **Stanford NER**: CRF-based, commonly used in academic research.
- **BERT-based NER**: Context-aware deep learning model for high accuracy.
- **Flair NER**: Character-level embeddings for multilingual NER.

Sentiment is compositional and requires linguistic analysis



Tokens	Most	large	cities	in	the	US	had	morning	and	afternoon	newspapers	.
POS Tags	JJS	JS	NNS	IN	DT	NNP	VBD	NN	CC	NN	NNS	.
Chunk IDs	B-NP	I-NP	I-NP	B-NP	B-NP	I-NP	B-NP	B-NP	I-NP	I-NP	I-NP	B-NP

1st chunk
2nd chunk
3rd chunk
4th chunk
5th chunk
6th chunk



Analysis Type	Focus	Applications
Text Preprocessing	Cleaning and normalizing raw text. Serves as a foundation for further linguistic processing.	Tokenization, stop-word removal, case folding, punctuation handling
Morphological	Word structure (root words, prefixes, suffixes, inflections, and derivations)	Stemming, Lemmatization, Morphological segmentation
Syntactic	Sentence structure, grammar, parts of speech (nouns, verbs, etc.)	Part-of-Speach (POS) tagging, parsing, chunking, sentence segmentation
Semantic	Meaning of words & sentences, word sense disambiguation, synonymy, antonymy	Named Entity Recognition (NER), Relationship Extraction, Word Sense Disambiguation (WSD), semantic parsing
Pragmatic	Contextual meaning, speaker intent, ambiguity, sarcasm, politeness	Coreference resolution, sentiment analysis, discourse analysis

Working with text data

Introduction

- A problem with modeling text is that it is messy.
- Text data is inherently unstructured and qualitative.
- It is composed of words, phrases, and sentences that convey meaning through language, which is complex and nuanced.
- ML algorithms cannot work with raw text directly and prefer well defined fixed-length inputs and outputs.
- The text must be converted into numbers. Specifically, **vectors of numbers**.

Working with text data

Converting textual data to numerical

In the context of NLP, **Vectorization** and **Feature Extraction** (or feature encoding) are closely related concepts, often used interchangeably.

- Both involve transforming text data into a numerical format that machine learning algorithms can process.

Vectorization happens first

· Converts text into a numerical format.

· Focuses on transforming words or sentences into vectors.

Examples:

· Bag of Words (BoW)
· TF-IDF
· Word Embeddings (Word2Vec, GloVe)
· Transformers (BERT, GPT)

Feature Extraction follows

· Derives meaningful features and insights from text for better model performance.

· Often involves vectorization but enhances it with linguistic or statistical techniques.

Examples:

· N-grams
· Named Entity Recognition (NER)
· Sentiment analysis
· Topic Modeling.

Bag of Words (BoW)

Concept:

- Represents text as a **word occurrence matrix** (simply counting how often each word appears).
- Each word is treated as an independent feature.
- The order of words is ignored.

Limitations:

- ✗ Ignores word meaning and context.
- ✗ High-dimensional representation.

Example:

- Doc 1: "Machine learning is great"
- Doc 2: "Deep learning is powerful"

Word	Doc 1	Doc 2
Machine	1	0
Learning	1	1
Deep	0	1
Great	1	0
Powerful	0	1

Bag-of-Words models

Bag of Words Example

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

Term	Document 1	Document 2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

Stopword List

for
is
of
the
to

TF-IDF (Term Frequency - Inverse Document Frequency)

▸ Video: Concept ▸ Video: Calculations

Enhances **BoW** by weighting words based on their importance.

- TF-IDF assigns weights to words based on:
  - **TF (Term Frequency)**: Frequent words in a document get higher scores.
  - **IDF (Inverse Document Frequency)**: Measures the rareness of a term. Common words across documents get lower scores.
- Words that are frequent in one document but rare in others get higher scores.

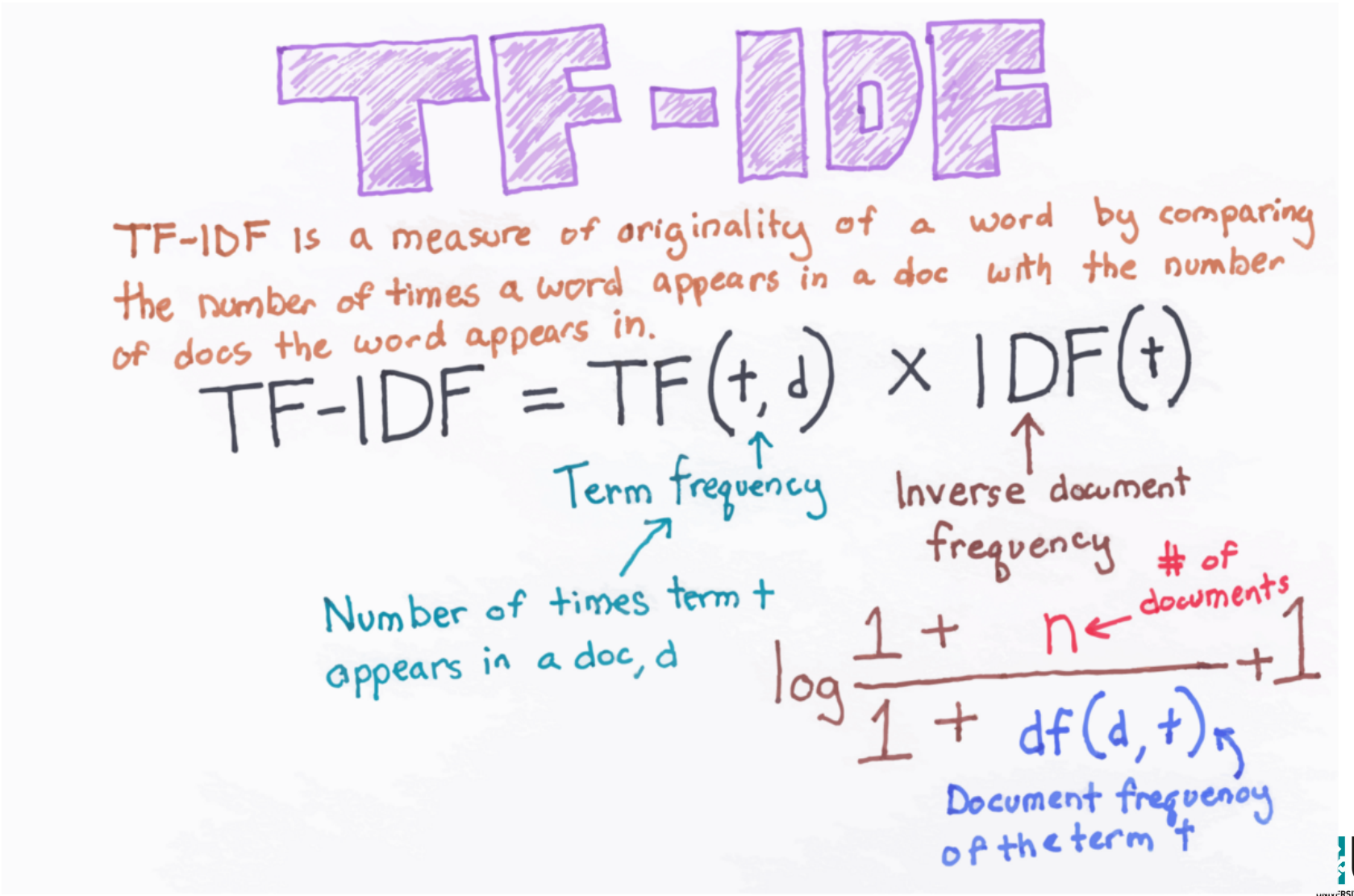
Advantages:

- ✓ Reduces noise from common words.
- ✓ Better for information retrieval.

Limitations:

- ✗ Ignores word meaning and context.
- ✗ High-dimensional and sparse representation.
- ✗ Cannot handle synonyms (e.g., "car" and "automobile" are unrelated).

TF-IDF (Word Originality)



Word Embeddings

Introduction

▸ 3D Embedding Projector ▸ Word Analogies

What are Word Embeddings?

- Word embeddings are vector representations of words that capture **semantic relationships**.
- Traditional methods like **TF-IDF** represent words based on frequency.
- Modern embeddings like **Word2Vec, GloVe, and Transformers** capture **context and meaning**.

Word Embeddings (Word2Vec, GloVe)

▸ Video: Concept

Word embeddings represent words as dense vectors, capturing their meaning based on context.

- **Word2Vec:** Uses Skip-gram and CBOW models.
  - **CBOW (Continuous Bag of Words)**: Predicts a target word from surrounding words.
  - **Skip-gram**: Predicts surrounding words from a target word.
- **GloVe (Global Vectors for Word Representation):** Learns word representations from co-occurrence matrices.

Advantages:

- ✓ Captures word relationships (semantic meaning).
- ✓ Words with similar meanings have similar vectors.
- ✓ Handles synonyms and semantic relationships (e.g., "car" and "automobile" are close in space).

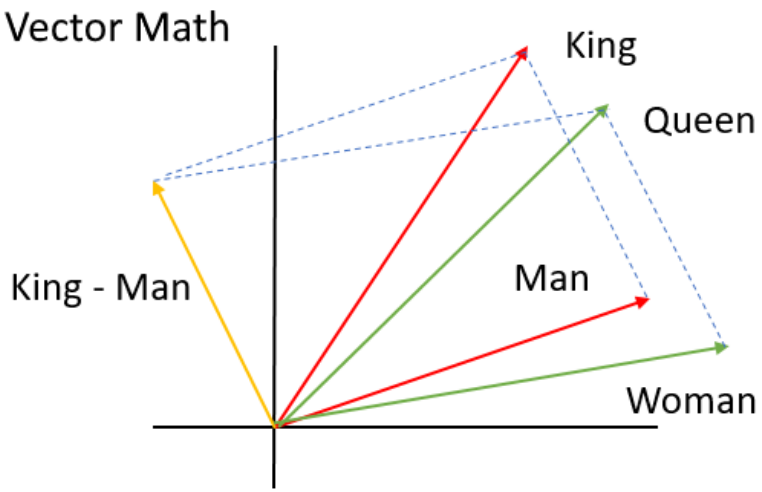
Limitations:

- ✗ Requires a large corpus for good results.
- ✗ Does not handle out-of-vocabulary words well.
- ✗ Context-independent (e.g., "bank" in finance vs. river).

UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Word2Vec in Action

Vector Maths



Example of Word Relationships:

- **King - Man + Woman = Queen**
- **Paris - France + Germany = Berlin**

Why It Works:

- Words in similar contexts have similar vectors.
- Word2Vec learns relationships from raw text.

Transformers: Contextual Word Embeddings

BERT, GPT, and beyond

▸ Video: Concept ▸ Video: Transformers Visually Explained

Transformers use **self-attention mechanisms** to capture long-range dependencies in text, enabling advanced NLP applications.

- Unlike **Word2Vec**, words have different embeddings depending on context.
- **BERT (Bidirectional Encoder Representations from Transformers)**:
  - **Bidirectional Training**: Considers both left and right context.
  - **Masked Language Model (MLM)**: Predicts missing words in a sentence.
- **GPT (Generative Pre-trained Transformer)**:
  - **Autoregressive Training**: Predicts next word in a sequence.
  - **Fine-tuned for text generation**: Used in ChatGPT, summarization, and conversational AI.

Advantages:

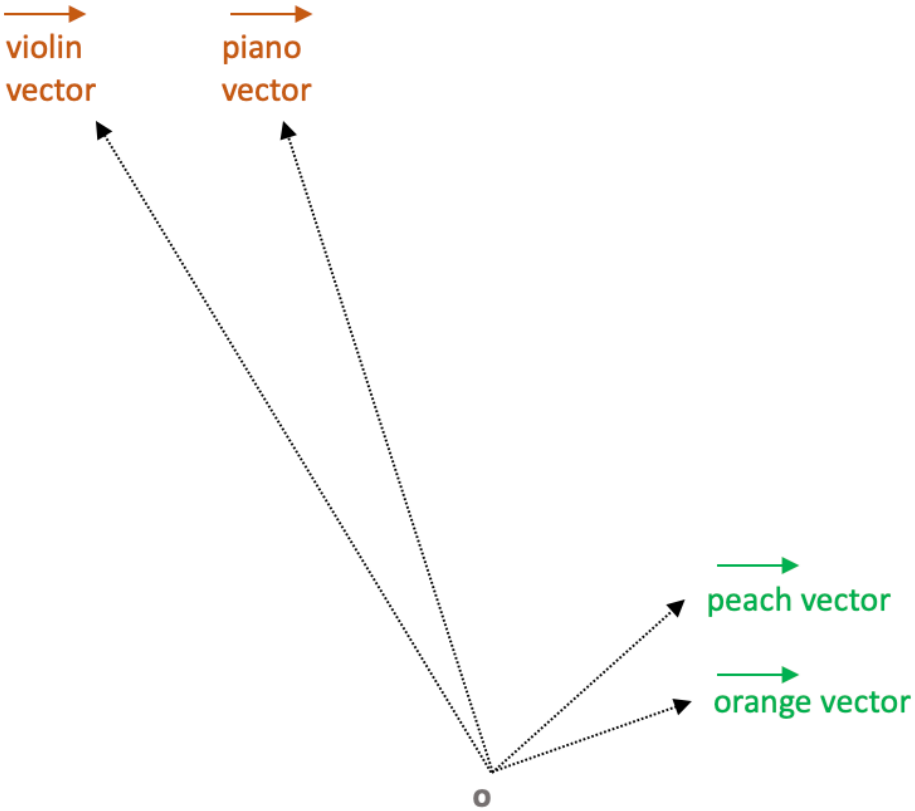
- ✓ Captures **contextual** meaning of words.
- ✓ Handles long-range dependencies in text.
- ✓ Powers state-of-the-art NLP applications (translation, Q&A, summarization).

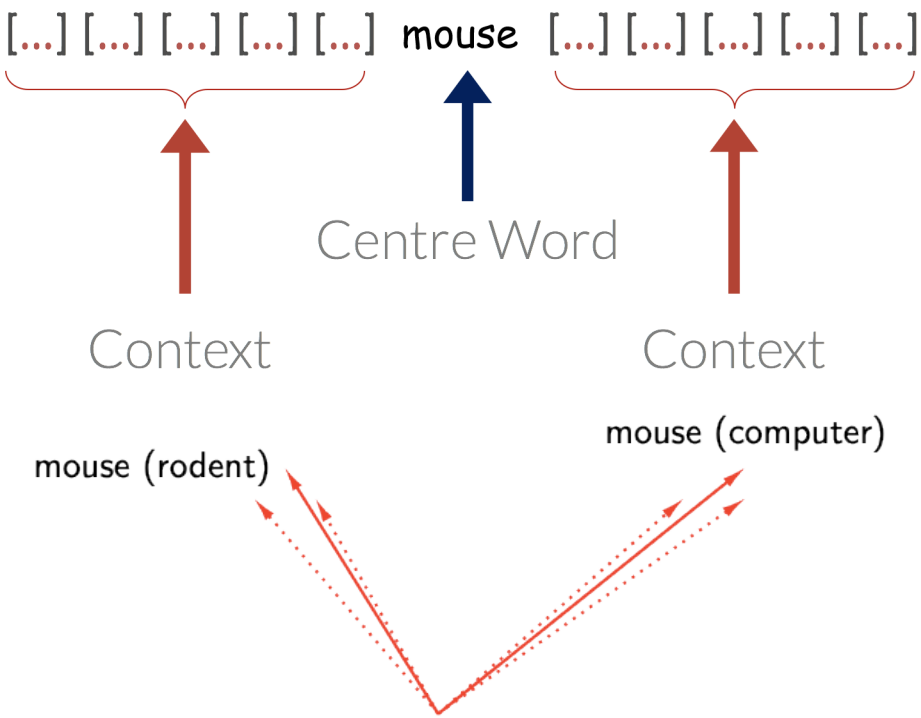
Limitations:

- ✗ Computationally expensive and requires large-scale hardware.
- ✗ May generate biased or misleading outputs.
- ✗ Requires extensive training data and fine-tuning.

Transformers: Contextual Word Embeddings

Words have **different embeddings** depending on context.





Example:

- "He went to the **bank** to withdraw money." → **bank** (finance).
- "The boat reached the **bank** of the river." → **bank** (geography).

Comparison of Feature Extraction Method

Feature	BoW	TF-IDF	Word2Vec	Sentence Emb.	Transformers
Captures Meaning?	✗	✗	✓	✓	✓
Context-Aware?	✗	✗	✗	✓	✓
Handles Synonyms?	✗	✗	✓	✓	✓
Handles OOV Words?	✗	✗	✓	✓	✓
Dimensionality	High	High	Low	Low	Medium
Computational Cost	Low	Low	Medium	Medium	High
Pre-trained Models Available?	✗	✗	✓	✓	✓
Handles Long Texts Efficiently?	✓	✓	✗	✓	✗

* **OOV** (Out-of-Vocabulary) words are words that were not seen during training.

- **TF-IDF** is simple but lacks meaning.
- **Word2Vec** learns semantic relationships but is not context-aware.
- **Transformers** (BERT, GPT) provide the most advanced word embeddings with context sensitivity.

Future Directions:

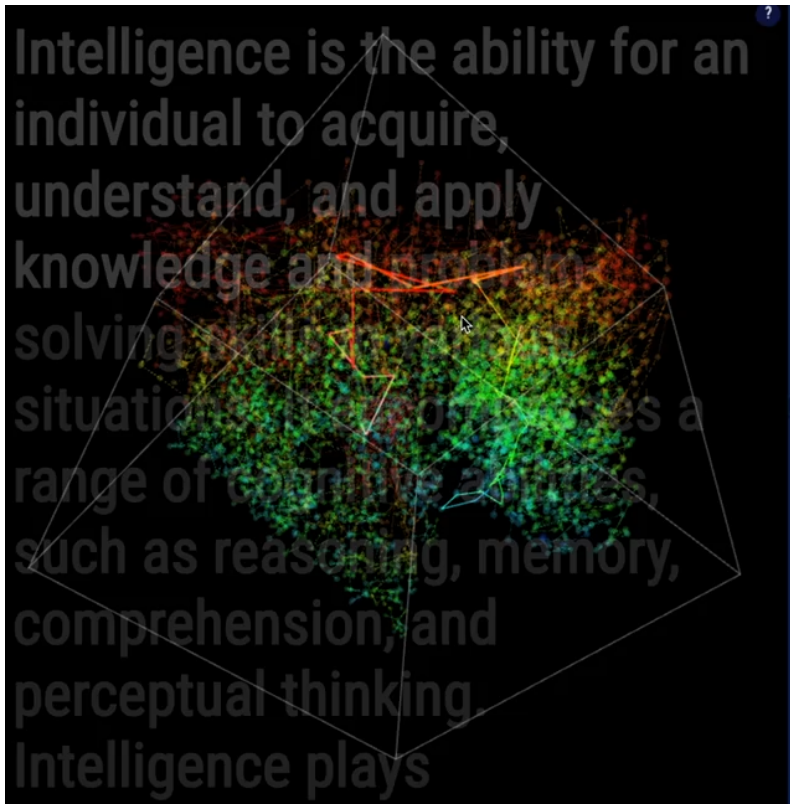
- Fine-tuning pre-trained transformers for domain-specific tasks.
- FinBERT Financial Sentiment Analysis: Hugging Face Demo on ▸ YouTube
- A fine-tuned version of FinBERT trained on Twitter Financial News Topic dataset ▸ Interactive

Linguistic Highways

▸ Video ▸ Interactive

Mapped ChatGPT's thought process:

- Run the prompt *Intelligence is* hundreds of times with a high temperature setting (1.6).
- Capture the model's diverse responses.
- Use PCA, to compress the 1536-dimensional word embeddings into a 3D visualization, showcasing how AI builds sentences word by word.

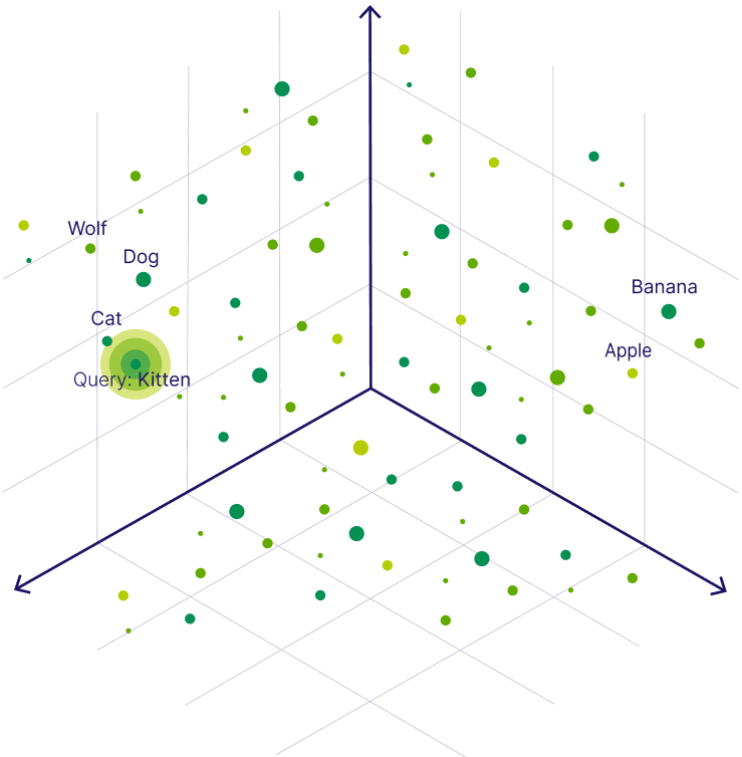


3D cube displaying bifurcating response paths

tree diagram illustrating word probability choices

Cosine Similarity in Word Embeddings

Concept



- Cosine similarity measures the similarity between two vectors by computing the cosine of the angle between them.
- In word embeddings, words are represented as high-dimensional vectors.
- Cosine similarity helps determine how similar two words are based on their vector representations.

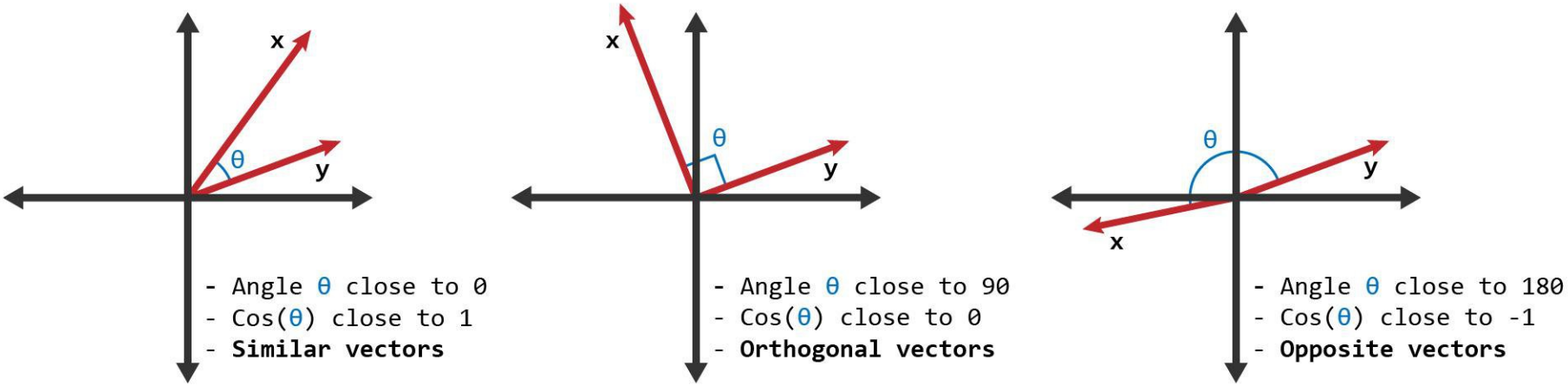
Mathematical Definition:

$$\cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

Cosine Similarity in Word Embeddings

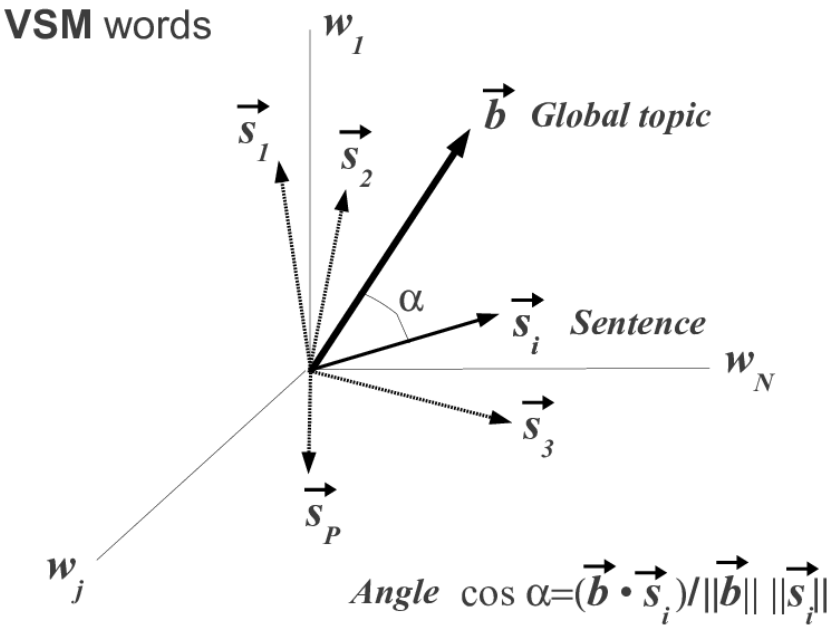
Intuition

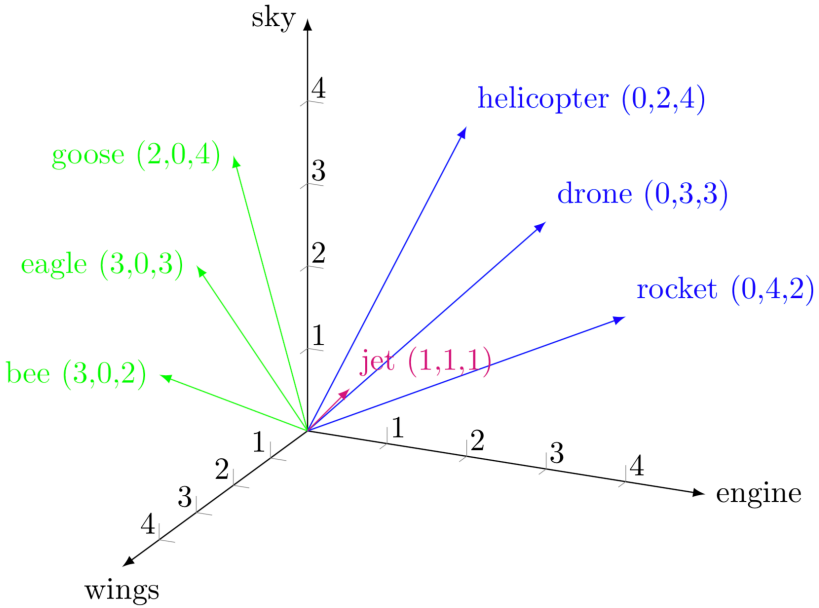
- If two words have similar meanings, their embeddings should be close in the vector space.
- Cosine similarity ranges from -1 to 1:
  - 1 indicates identical vectors (completely similar words).
  - 0 indicates orthogonal vectors (unrelated words).
  - -1 indicates completely opposite words (rare in word embeddings).
- Widely used in NLP for measuring similarity between words, sentences, and documents



Word embeddings

Cosine Similarity





Applications of Cosine Similarity in NLP I

Information Retrieval & Search Engines

- Used in **document ranking** by measuring similarity between a query and indexed documents.
- Helps in **semantic search** by finding documents with similar meanings.
- **Example**: Google Search retrieves relevant web pages based on keyword embeddings.

Text Similarity & Plagiarism Detection

- Measures similarity between **two sentences** or **documents**.
- Used in **plagiarism detection systems** to compare text submissions against an existing corpus.
- **Example**: Turnitin, Grammarly, and Copyscape detect copied content using cosine similarity.

Question Answering & Chatbots

- Identifies **similar questions** in FAQ-based chatbots.
- Enhances **intent detection** in conversational AI.
- **Example**: Customer support chatbots match user queries with predefined answers.

Applications of Cosine Similarity in NLP II

Recommender Systems

- Used in **content-based recommendation systems** for suggesting similar items.
- Compares **descriptions, user preferences, and historical interactions**.
- **Example**: Netflix movie recommendations, Amazon product suggestions.

Machine Translation & Paraphrase Detection

- Evaluates **semantic similarity** between translated sentences.
- Helps in **paraphrase detection** and **duplicate question identification**.
- **Example**: Quora's duplicate question detection and Google Translate evaluation.

Text Summarization

- Used in **extractive text summarization** to find the most relevant sentences.
- Helps **identify key phrases** that best represent the content.
- **Example**: Summarizing news articles by selecting key sentences.

Applications of Cosine Similarity in NLP III

Document Clustering & Topic Modeling

- Helps in **grouping similar documents together** in unsupervised learning tasks.
- Used in **news categorization, topic clustering, and recommendation systems**.
- **Example**: Clustering customer reviews by sentiment in e-commerce platforms.

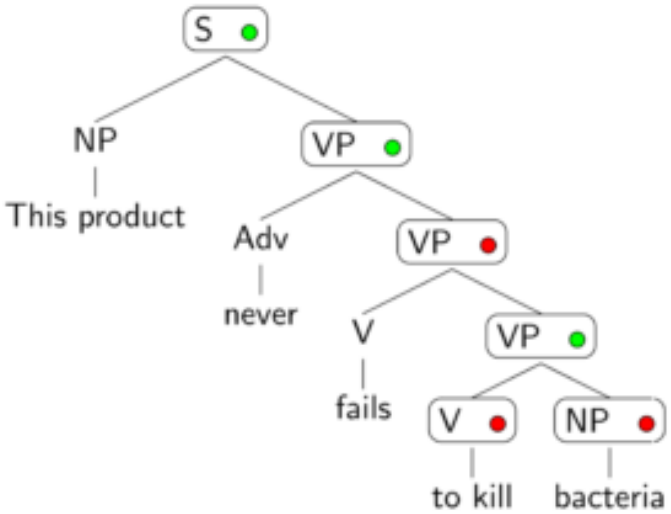
Sentiment Analysis & Opinion Mining

- Measures **similarity between user reviews or social media posts**.
- Helps **group similar sentiments together** in sentiment analysis.
- **Example**: Analyzing public opinion on social issues using Twitter embeddings.

Fake News & Misinformation Detection

- Detects **semantic similarity** between **fake** and **real news** articles.
- Identifies **patterns of misinformation by clustering similar articles**.
- **Example**: Detecting fake news on social media platforms like Twitter and Facebook.

Sentiment is compositional and requires linguistic analysis

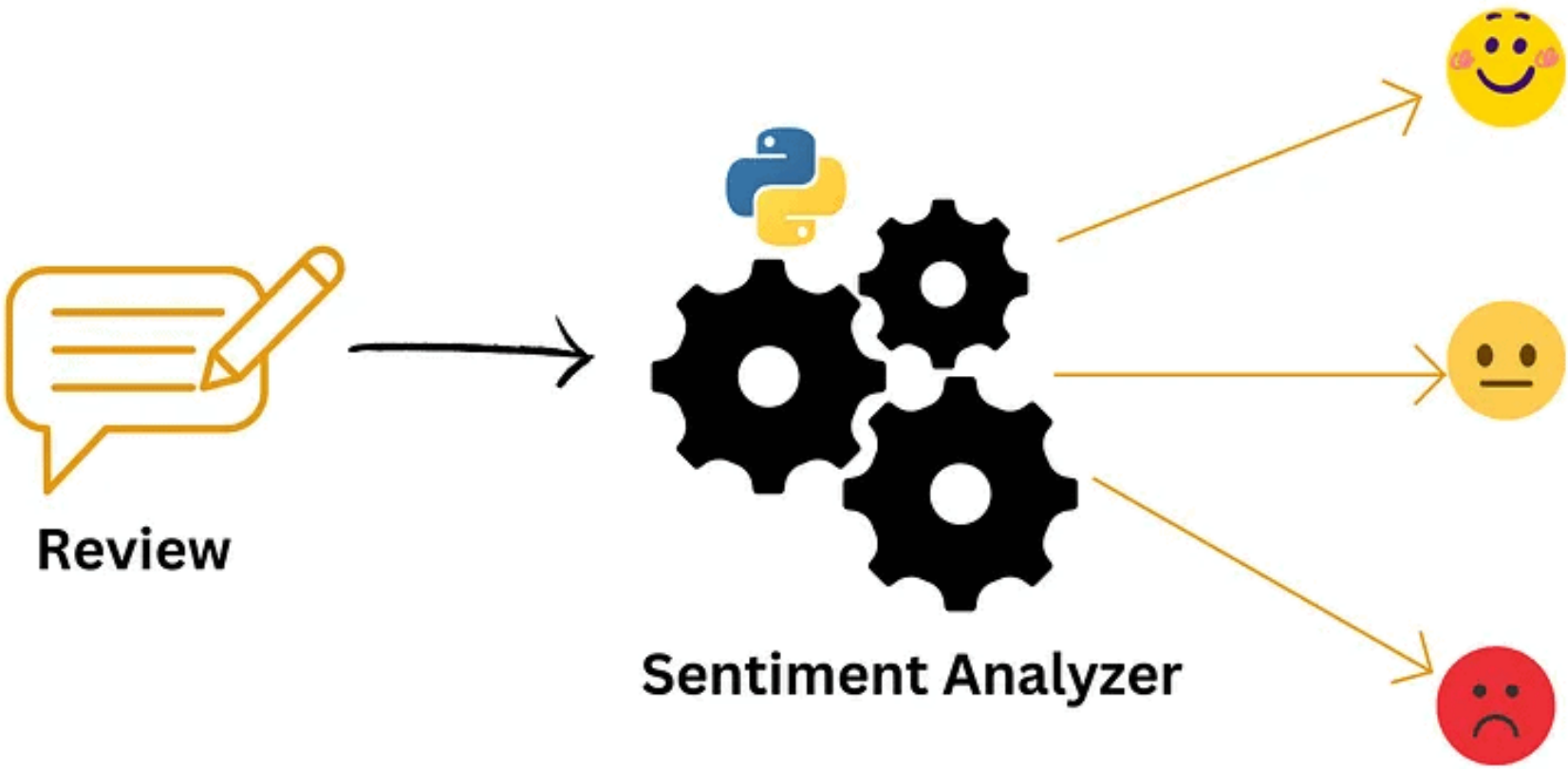


Tokens	Most	large	cities	in	the	US	had	morning	and	afternoon	newspapers	.		
POS Tags	JJS	JS	NNS	IN	DT	NNP	VBD	NN	CC	NN	NNS	.		
Chunk IDs	B-NP	I-NP	I-NP	B-NP	B-NP	I-NP	B-NP	B-NP	I-NP	I-NP	I-NP	B-NP		

1st chunk
2nd chunk
3rd chunk
4th chunk
5th chunk
6th chunk

Types of Sentiment Analysis in NLP

Sentiment Analysis (Opinion Mining) can be categorized into different types based on **granularity**, **classification**, **approach**, and **complexity**.



Based on Granularity (Level of Analysis)

- **Document-Level Sentiment Analysis**
  - Determines the overall sentiment of an entire document.
  - **Example**: Classifying a product review as "positive" or "negative".
- **Sentence-Level Sentiment Analysis**
  - Analyzes sentiment at the sentence level.
  - **Example**: "I love the phone, but the battery life is terrible" (mixed sentiment).
- **Aspect-Based Sentiment Analysis (ABSA)**
  - Identifies sentiment toward specific aspects of a product/service.
  - **Example**: "The camera is great, but the screen is too dim."
    Camera $\rightarrow$ *Positive, Screen* $\rightarrow$ *Negative*
- **Entity-Level Sentiment Analysis**
  - Extracts sentiment toward a specific named entity (e.g., brand, person, location).
  - **Example**: "Tesla's new model is amazing, but Apple's design is outdated."

Based on Classification Type

- **Binary Sentiment Analysis**
  - Classifies sentiment as **positive** or **negative**.
  - **Example**: "I enjoyed the movie" $\rightarrow$ *Positive*, "The movie was boring" $\rightarrow$ *Negative*.
- **Multi-Class Sentiment Analysis**
  - Categorizes sentiment into multiple levels (e.g., positive, neutral, negative).
  - **Example**: Some systems use a **5-star rating scale**.
- **Fine-Grained Sentiment Analysis**
  - Provides more detailed sentiment classification (e.g., very positive, slightly positive, neutral, slightly negative, very negative).
  - **Example**: **1 to 5 stars** where *1 = very negative, 5 = very positive*.

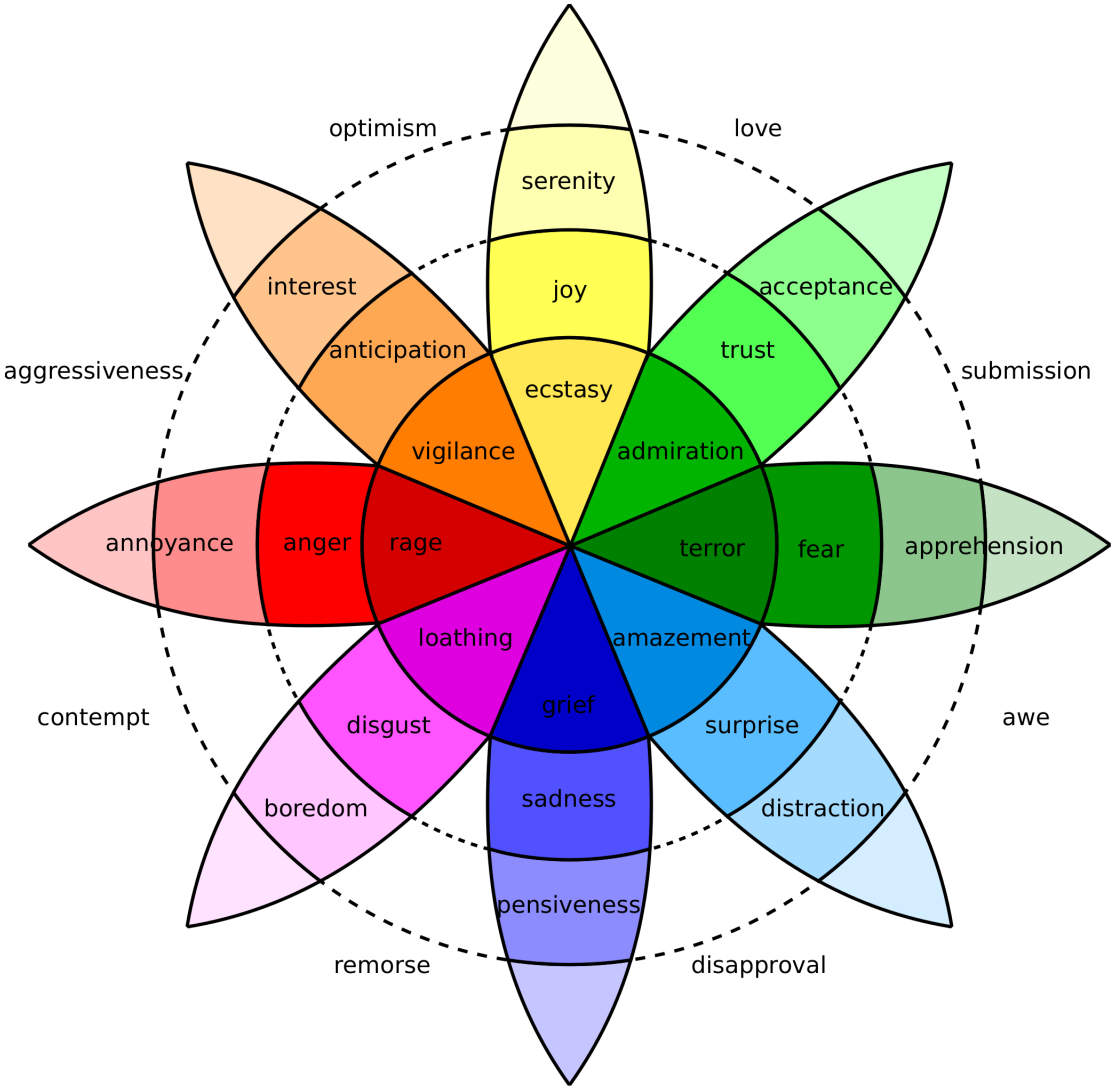
Based on Sentiment Approach

- **Lexicon-Based Sentiment Analysis**
  - Uses predefined lists of sentiment words (e.g., "happy," "amazing," "sad," "horrible").
  - **Example**: "Wonderful experience!" $\rightarrow$ Positive (based on the word "wonderful").
- **Machine Learning-Based Sentiment Analysis**
  - Uses supervised learning models trained on labeled datasets (e.g., Logistic Regression, SVM, LSTMs, Transformers).
  - **Example**: AI models trained on thousands of reviews to predict sentiment.
- **Hybrid Sentiment Analysis**
  - Combines lexicon-based and machine learning approaches.
  - **Advantage**: Balances interpretability and accuracy.

Based on Sentiment Context & Complexity

- **Emotion Detection**
  - Goes beyond positive/negative to detect emotions (happiness, sadness, fear, etc.).
  - **Example**: "I am devastated by this loss." → *Sadness*.
- **Sarcasm Detection**
  - Identifies irony or sarcasm that may mislead traditional models.
  - **Example**: "Great! Another Monday morning meeting..." → Likely *negative*, despite the word "Great!".
- **Aspect-Based Opinion Summarization**
  - Summarizes opinions on different aspects rather than just classifying sentiment.
  - **Example**: Aggregates customer reviews to summarize what users liked or disliked about a product.

Robert Plutchik's set of eight basic emotions



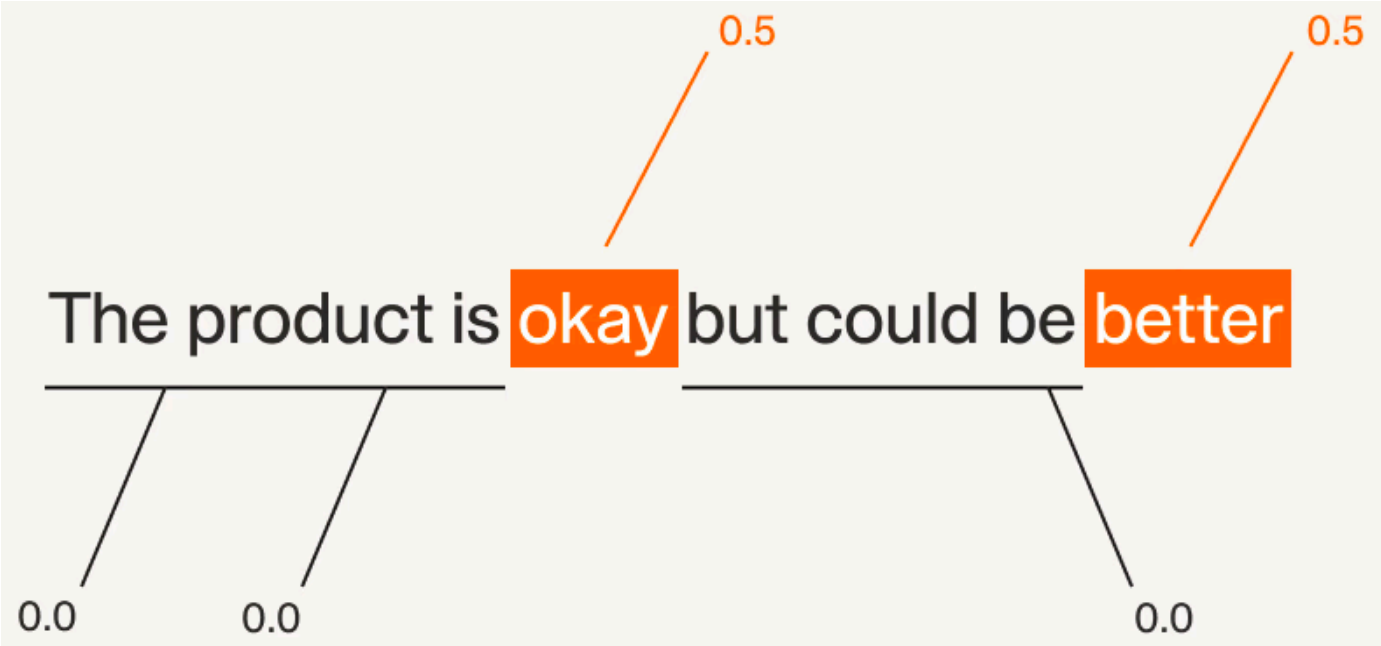
Sentiment Analysis

TextBlob: Simple and intuitive method

- Calculates the overall sentiment polarity by averaging the polarity scores of individual words, like verbs and adjectives, which carry sentiment information.
- Two additional parameters: confidence and intensity. These scores can modify the strength of the sentiment, reflecting how strongly a word expresses an emotion.

Sentiment Analysis

TextBlob: Struggles with nuanced language



Sentiment Analysis

VADER (Valence Aware Dictionary and sEntiment Reasoner)

Strengths:

- ✓ Performs well on social media text, handling slang, abbreviations, and emoticons.
- ✓ Does not require labeled training data, making it easy to implement.
- ✓ Provides intensity scores for sentiment (e.g., positive, negative, neutral, compound score).

Limitations:

- ✗ Struggles with long and complex sentences.
- ✗ Limited ability to understand sarcasm and deep contextual meanings.
- ✗ More effective in informal and social media-based contexts rather than formal text.

Input	neg	neu	pos	compound
"This computer is a good deal."	0	0.58	0.42	0.44
"This computer is a very good deal."	0	0.61	0.39	0.49
"This computer is a very good deal!!"	0	0.57	0.43	0.58
This computer is a very good deal!! :-)"	0	0.44	0.56	0.74
This computer is a VERY good deal!! :-)"	0	0.393	0.61	0.82

Comparison of Sentiment Analysers

The evolution of sentiment analysis techniques has seen significant advancements, from traditional methods such as NLTK, TextBlob, and VADER to more sophisticated approaches using AI and transformer-based models like BERT and ChatGPT-4.

Method	Pros	Cons
NLTK + TextBlob	Simple and effective for many basic applications.	Struggles with nuanced language, sarcasm, and context-specific meanings.
VADER	Excels with social media text, handles slang and emoticons well.	Limited in handling longer and more formal texts, basic contextual understanding.
ChatGPT-4/other LLMs	High accuracy, understands context, detects nuanced sentiment.	Using LLMs at scale can generate costs (token consumption). Requires lots of computational power if run locally.
V7 Go	Allows you to connect multiple LLMs to boost performance and get structured outputs, like tags, CSV reports, or JSON code.	It is very customizable, which means that in some cases the initial setup may require some experimentation.
BERT-based	Uses transformer architecture for context-aware sentiment analysis, excels in complex text understanding.	Requires large datasets for fine-tuning, computationally expensive.
RoBERTa	More robust version of BERT, trained on a larger dataset, achieving higher accuracy.	Needs significant fine-tuning for domain-specific tasks, high computational costs.
DistilBERT	Lighter version of BERT, retains most accuracy while reducing computational load.	Slightly lower accuracy than full-scale BERT models, not optimal for highly nuanced sentiment tasks.
LSTM-based	Effective at capturing sequential dependencies in text, handles long-term context.	Requires large labeled datasets, training can be slow compared to transformers.

Analysis of Earnings Call Transcripts

Earnings Call Transcripts Offer a Window to Otherwise "Hidden" Data

	Sections		People		
	Prepared Statements	Unscripted Q&A	Company Management	External Analysts	All Sections
Number of Sentences	18	189	55	152	219
Number of Words	1,098	10,305	7,804	3,599	11,403
Word Sentiment	0.67	-0.02	0.69	-0.18	0.28
Sentence Sentiment	0.64	-0.06	0.72	-0.23	0.32
Topic Sentiment	0.00	-0.13	0.17	-0.38	-0.11
Language Complexity	8.9	6.3	8.1	5.8	7.8

The earnings call narrative was dominated by company insider dialogue during the Q&A portion, suggesting long-winded answers from management may have been used to avoid answering questions directly

Sentiment during the Q&A section was mildly negative when compared to the prepared statements, suggesting a degree of linguistic vetting by company management

Management uses marginally more complex language than analysts, which could suggest obfuscation

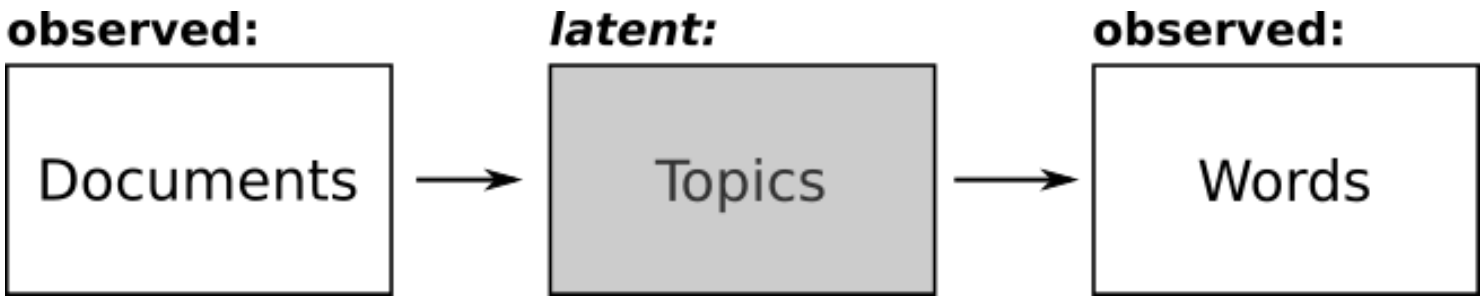
Alternative definitions of sentiment may yield contrasting views on a particular piece of text, helping to build a more comprehensive assessment of the language being used, generating richer analysis

Cook et al. 2019

What is Topic Modeling?

Topic modeling is a method for discovering **abstract topics** (hidden concepts) within a **collection of documents** (= **corpus**).

- Each corpus contains a **latent** (or "hidden") structure of topics.
- Some topics are more prominent in the whole corpus, some less.
- In each document there are multiple topics covered, each to a different amount.
- Assumes that words that are close in meaning will occur in similar pieces of text.



Types of Topic Modeling in NLP

Topic modeling techniques can be categorized into different types:

1. **Probabilistic Models**
   - Statistical approaches that model document-topic and topic-word distributions.
   - Example: **Latent Dirichlet Allocation (LDA)**
2. **Matrix Factorization-Based Models**
   - Linear algebra techniques that decompose document-word matrices to identify latent topics.
   - Example: **Latent Semantic Analysis (LSA)**
3. **Neural Network-Based Models**
   - Deep learning methods that use neural networks to learn topic distributions from text embeddings.
   - Example: **BERTopic**
4. **Hybrid and Alternative Approaches**
   - Combinations of different methods, such as graph-based topic modeling, which integrate clustering and probabilistic techniques.
   - Example: **TopicRank**

Choosing the Right Model

- Topic modeling is crucial in NLP for **identifying hidden themes** in large corpora.
- **Different methods** exist, ranging from probabilistic models to deep learning-based techniques.
- Choosing the right model **depends on the application requirements** (e.g., interpretability, scalability, efficiency).

Model	Best For	Limitations
LDA	General-purpose topic modeling	Requires tuning topics
PLSA	Information retrieval	Overfits with many topics
DTM	Tracking topic evolution	Computationally expensive
HDP	Auto-detecting topics	Complex inference
LSA	Word co-occurrence	Less interpretable
NMF	High interpretability	Non-negative constraints
BERTopic	Short texts, contextual topics	Requires embeddings
Top2Vec	Scalable topic discovery	High memory usage

What is LDA?

Latent Dirichlet Allocation (LDA) is a topic model that assumes:

- each document is made up from a **mixture of topics**
- each topic is a distribution over all words in the corpus

Assumptions:

- order of words in documents does not matte, e.g. Bag-of-Words model
- order of documents in a corpus does not matte
- no relations between words = no context! Topics are "neutral" (**but**: context of words within topics can be interpreted)
- documents can by anything: news articles, scientific papers, books, chapters of books, paragraphs **but**: each document should convey a **mixture of topics** - problematic with very short documents (tweets).
- the number of topics is known (has to be set in advance). Can infer from the data by evalating the quality of a model.

Latent Dirichlet Allocation (LDA)

Prerequisites:

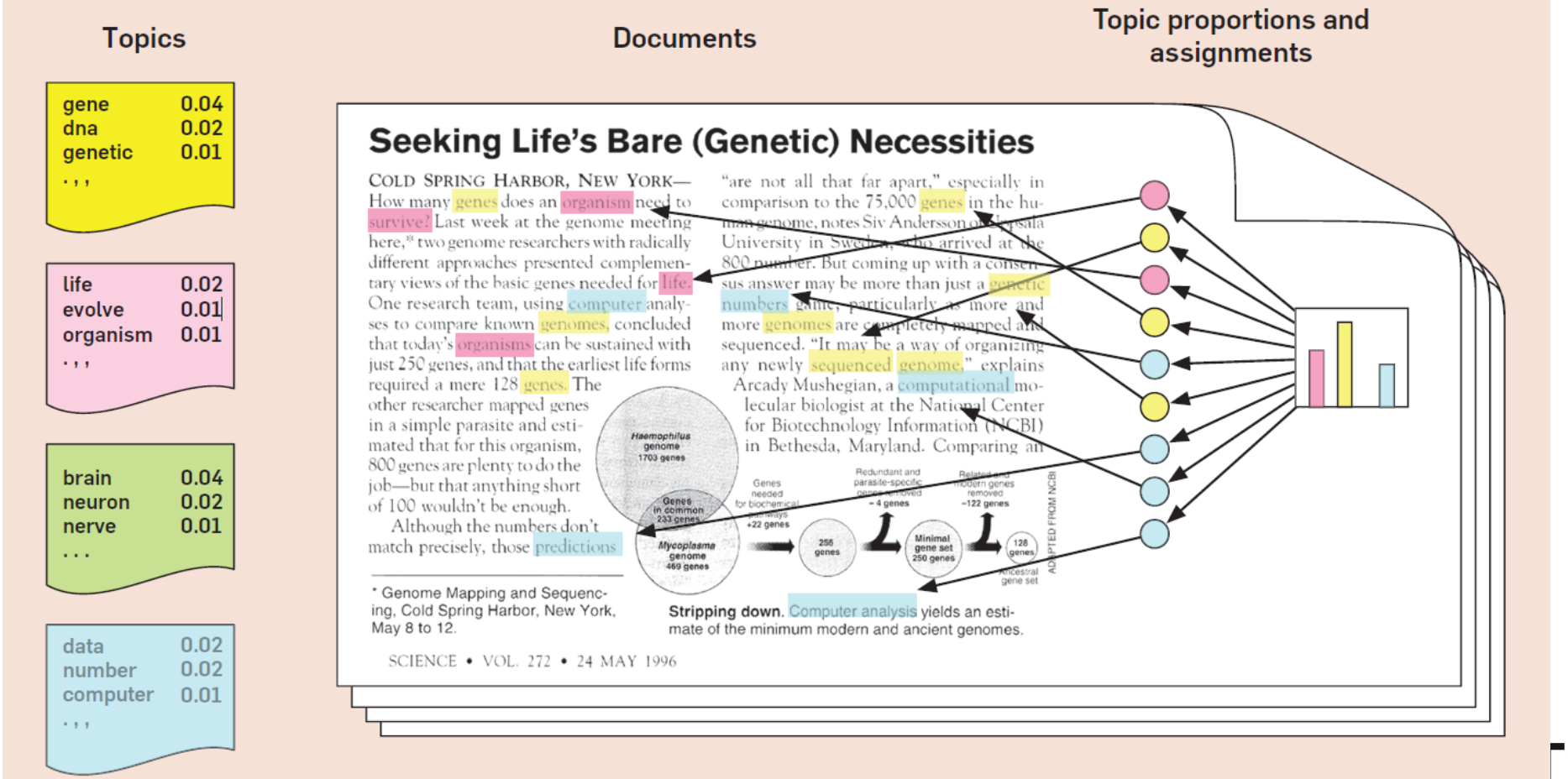
- Input text must be preprocessed
  - tokenize: split sentence into separte "tokens" (words)
  - normalize: stemming, lemmatization (find word stems)
  - filter: remove stopwords, punctuation, etc.

Before: "In the **first** **World Cup** **game**, **he** **sat** only on the **bank**."
After: **first**, **world**, **cup**, **game**, **he**, **sit**, **bank**

- Exact preprocessing steps (stopword removal, *n*-grams) are domain specific

Topic Modeling - LDA intuition

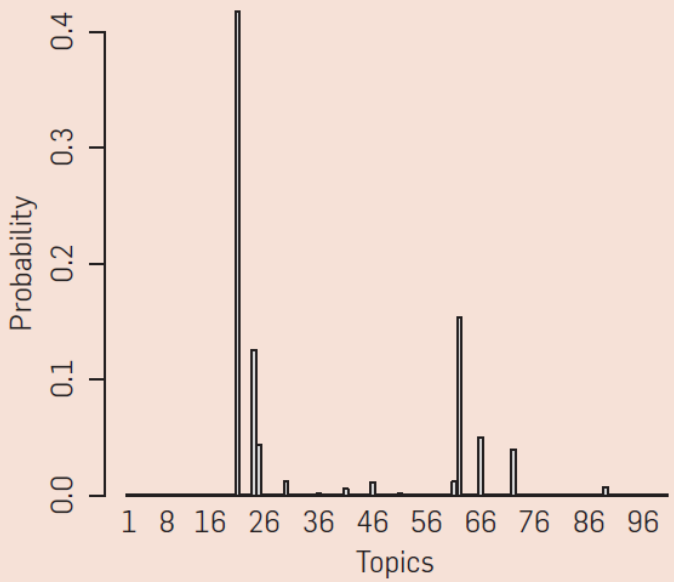
Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.



Source: Blei (2012)

Topic Modeling - LDA intuition

Figure 2. Real inference with LDA. We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left are the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

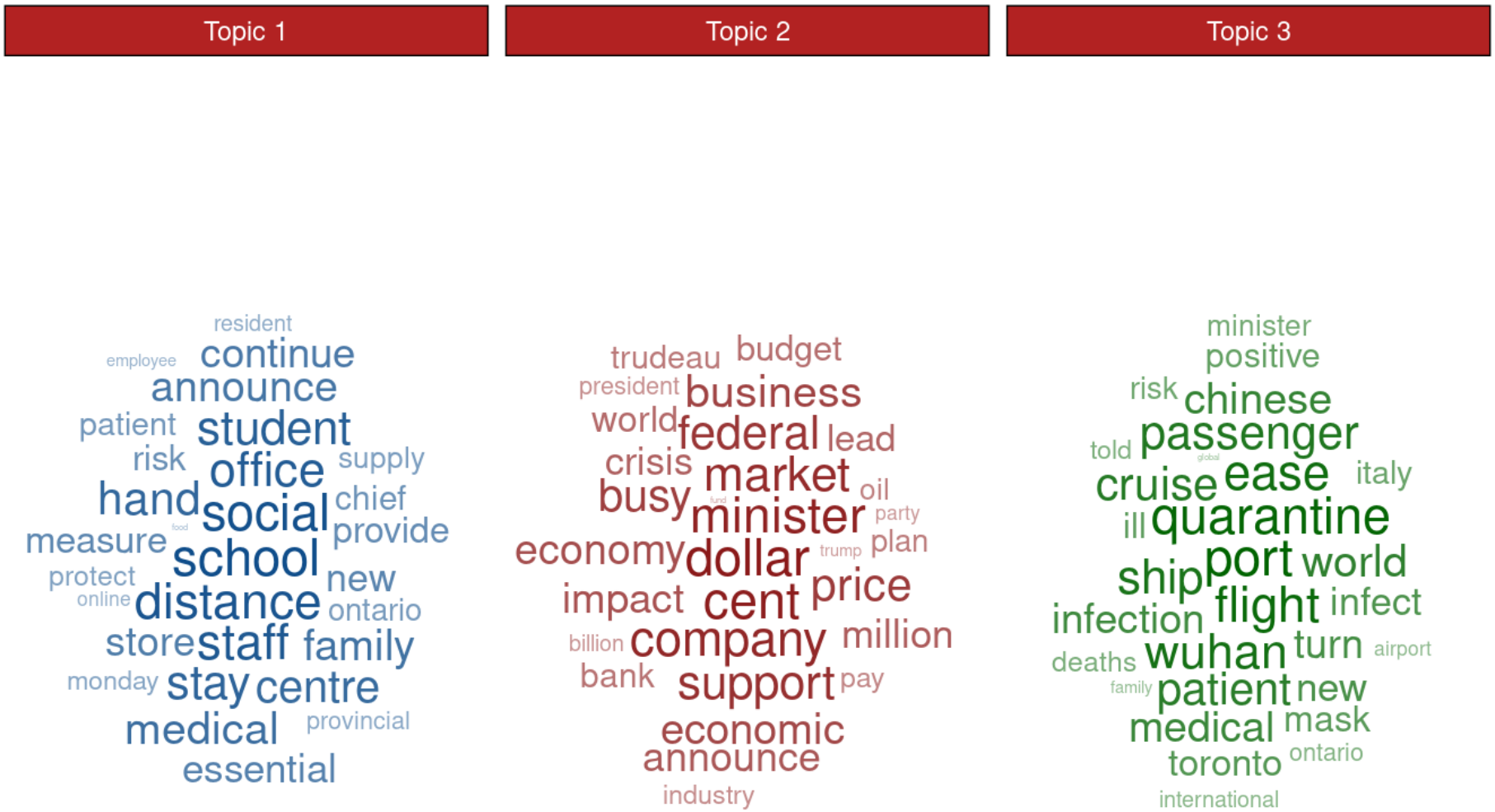
Source: Blei (2012)

Topic Modeling - LDA intuition

Figure 3. A topic model fit to the *Yale Law Journal*. Here, there are 20 topics (the top eight are plotted). Each topic is illustrated with its top-most frequent words. Each word's position along the *x*-axis denotes its specificity to the documents. For example "estate" in the first topic is more specific than "tax."

Source: Blei (2012)

Topic Modeling - LDA intuition

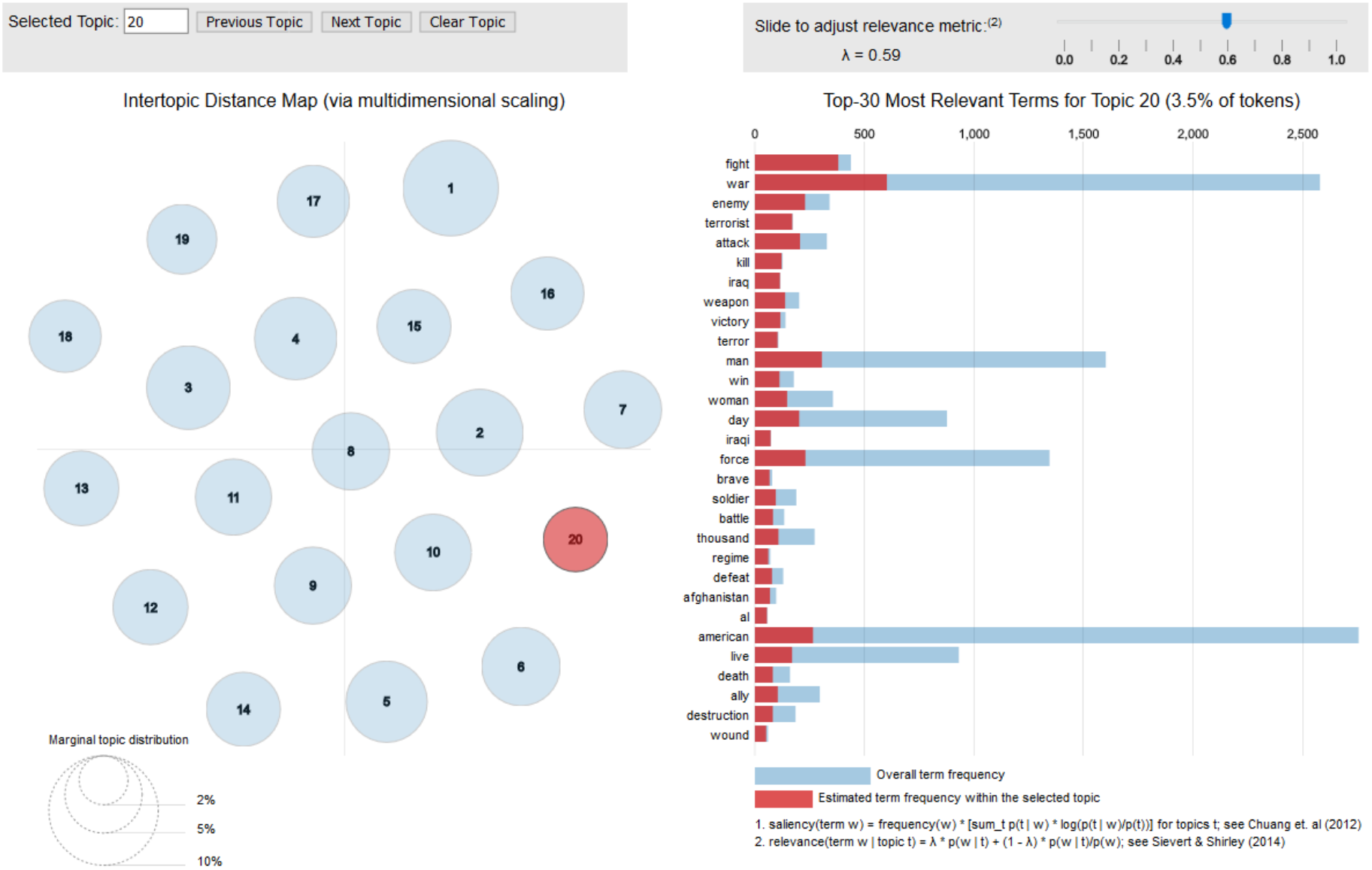


Topic Modeling - LDA implementation

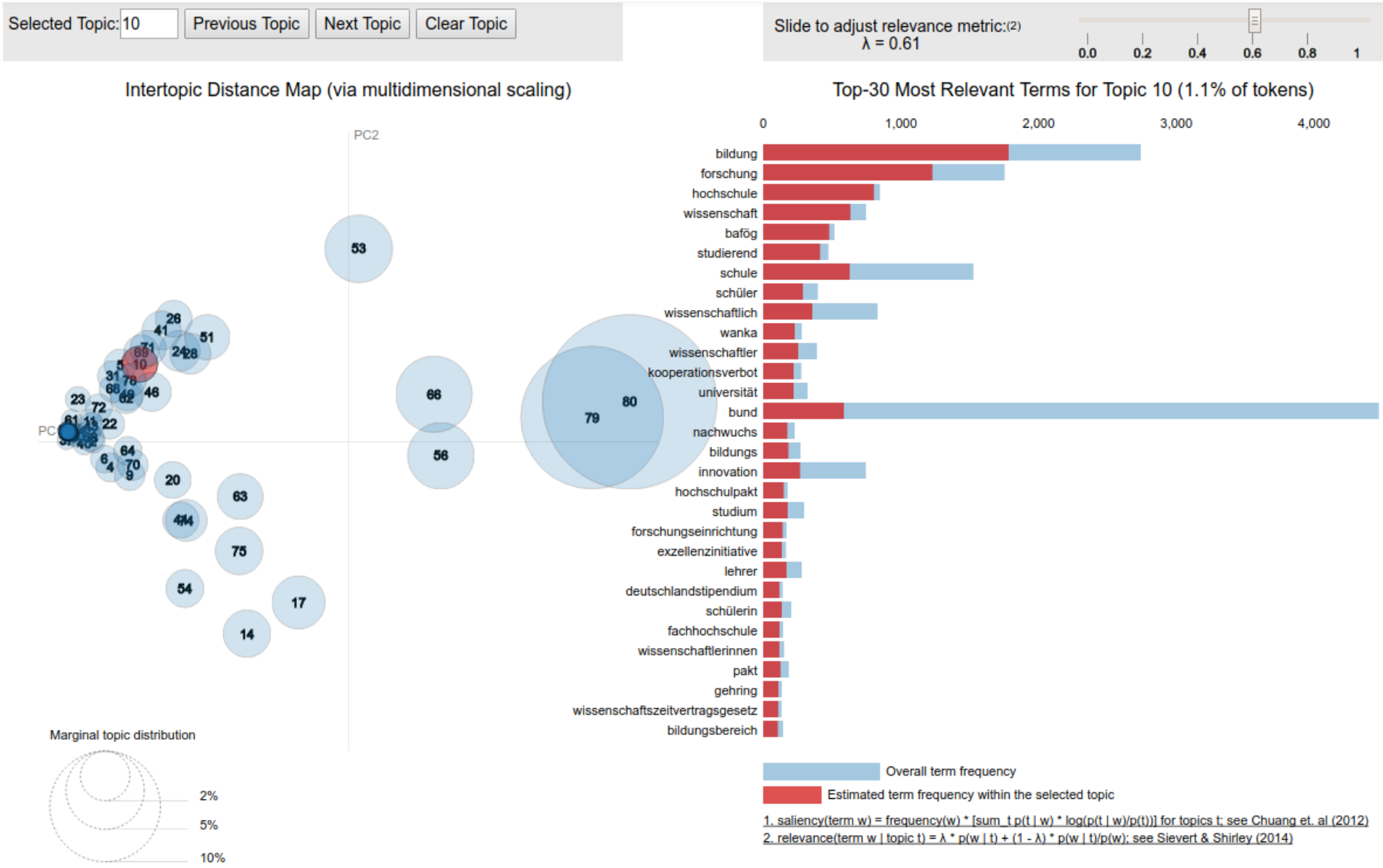
A Python package **PyLDAvis** allows an interactive visualization of a topic model.

- A brief incursion into LDA: ▸ details
- Visualization of topics and of its components plays a major role in interpreting the model ▸ PyLDAvis results (interactive)

PyLDAvis results: good LDA fit



PyLDAvis results: bad LDA fit



Text Classification vs Topic modeling

Text classification is a supervised machine learning problem, where a text document or article classified into a pre-defined set of classes.

Text classification examples in Python

Topic modeling is a form of unsupervised learning. Specifically, it is a form of dimensionality reduction but also clustering. It is the process of discovering groups of co-occurring words in text documents. These group co-occurring related words makes "topics". The set of possible topics are unknown.

Sentiment Analysis VS Sentiment Analytics

1. **Sentiment Analysis** (or opinion mining):
   - the process of determining the sentiment or **emotional tone expressed in text**, such as a social media post, customer review, or news article
   - involves analysing the language, words, and context to classify the sentiment as **positive, negative, or neutral**
   - focuses on extracting and quantifying the sentiment conveyed in the text
2. **Sentiment Analytics**:
   - a broad range of techniques and approaches to **analyse and interpret** sentiment data
   - applying advanced analytical methods to gain deeper insights, patterns, and trends from sentiment-related data
   - goes beyond simple sentiment classification and may involve **data visualization, trend analysis, predictive modeling, and sentiment-based decision-making**

Sentiment Data

More than 100 sentiment data feeds

- **Market Prophit** offers stock market sentiment analysis of social media conversations.
- **RavenPack**: provide FactSet clients with sentiment scores and abnormal sentiment.
- Interactive Data, **Sowa Labs** (Germany).
- **MarketPsych TRMI Sentiment Indexes** for Refinitiv.

Characteristics (... and challenges)

- A vast amount of data.
- High-dimensional.
- Irregularly spaced.
- Often sparse.

Refinitiv/LSEG MarketPsych data

MarketPsych offers three types of sentiment indicators:

1. **Emotional indicators** including *Sentiment*, *Anger*, and *Fear*
2. **Fundamental perceptions** such as *Long vs Short*, *Earnings Forecast*, and *Interest Rate Forecast*
3. **Buzz metric**, a measure indicative of how much activity are being generated and discussed.

Overall, TRMI offers a total of 35 emotional scores based on Social and News media (each).

Details in RL Peterson (2013). "Thomson Reuters MarketPsych Indices (TRMI) White Paper". In: *Inside the Mind of the Market*.

Textual content analyzed

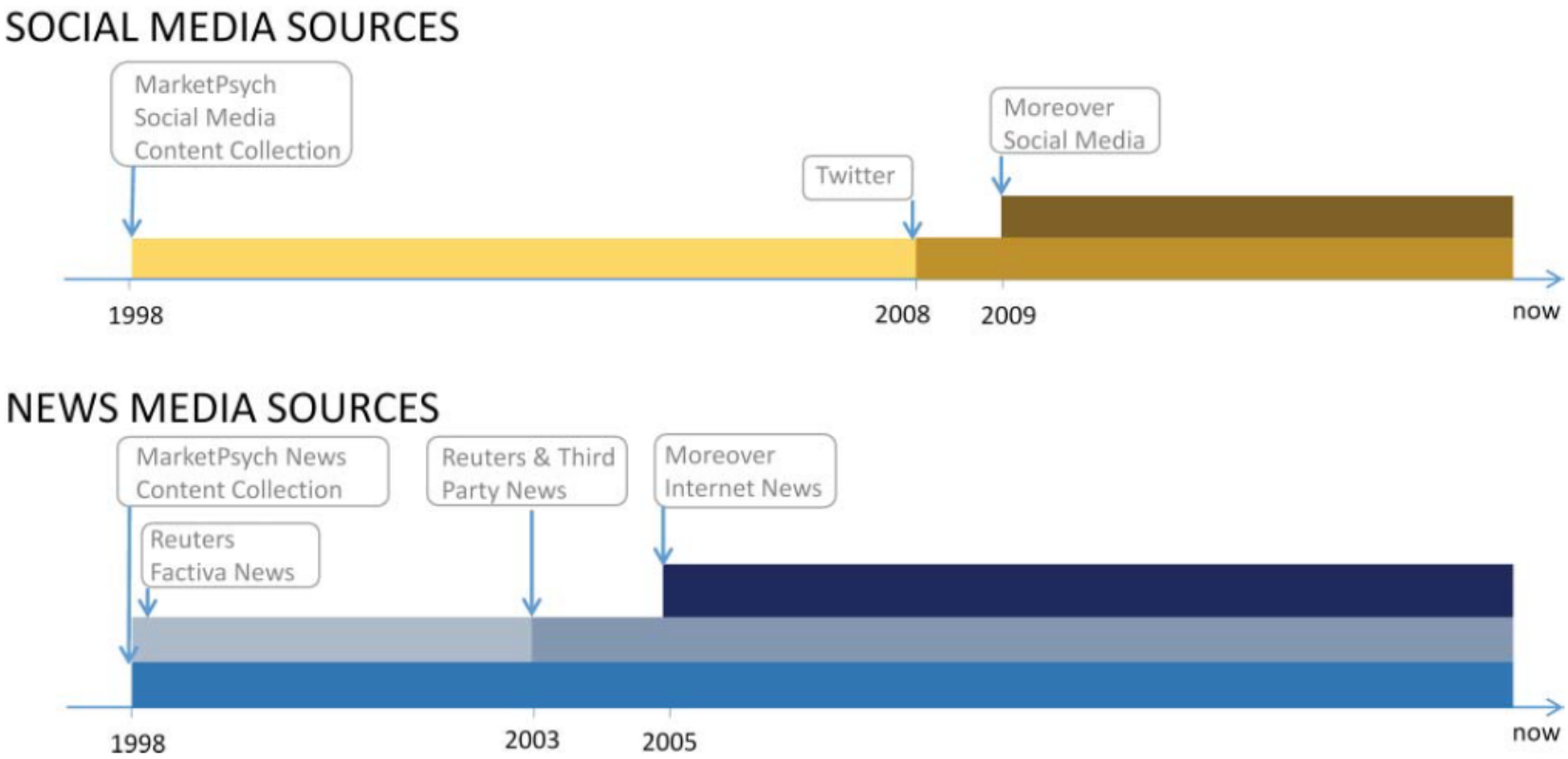


Figure: Timeline of textual content analyzed for the social and news media TRMI. **Moreover Technologies** offers media monitoring and news aggregation products; founded in 1998, acquired by LexisNexis in 2014.

Sentiment indices

Polarized Groups $[-1, 1]$	**Unidirectional Groups** $[0, 1]$	**Buzz** $[0, \infty)$
·sentiment	·anger	·buzz
·optimism	·fear	
·loveHate	·joy	
·trust	·gloom	
·conflict	·stress	
·timeUrgency	·surprise	
·emotionVsFact	·uncertainty	
·marketRisk	·violence	
·longShort	·volatility	
·longShortForecast	·debtDefault	
·priceDirection	·innovation	
·priceForecast	·laborDispute	
·analystRating	·layoffs	
·dividends	·litigation	
·earningsForecast	·mergers	
·fundamentalStrength	·cyberCrime	
·managementChange		
·managementTrust		

High Frequency Sentiment Data

▸ Apple's Intraday Sentiment Dynamics

Broad Market Index (S&P500 proxy) - Social media, 1min

Gan et al. (2019)

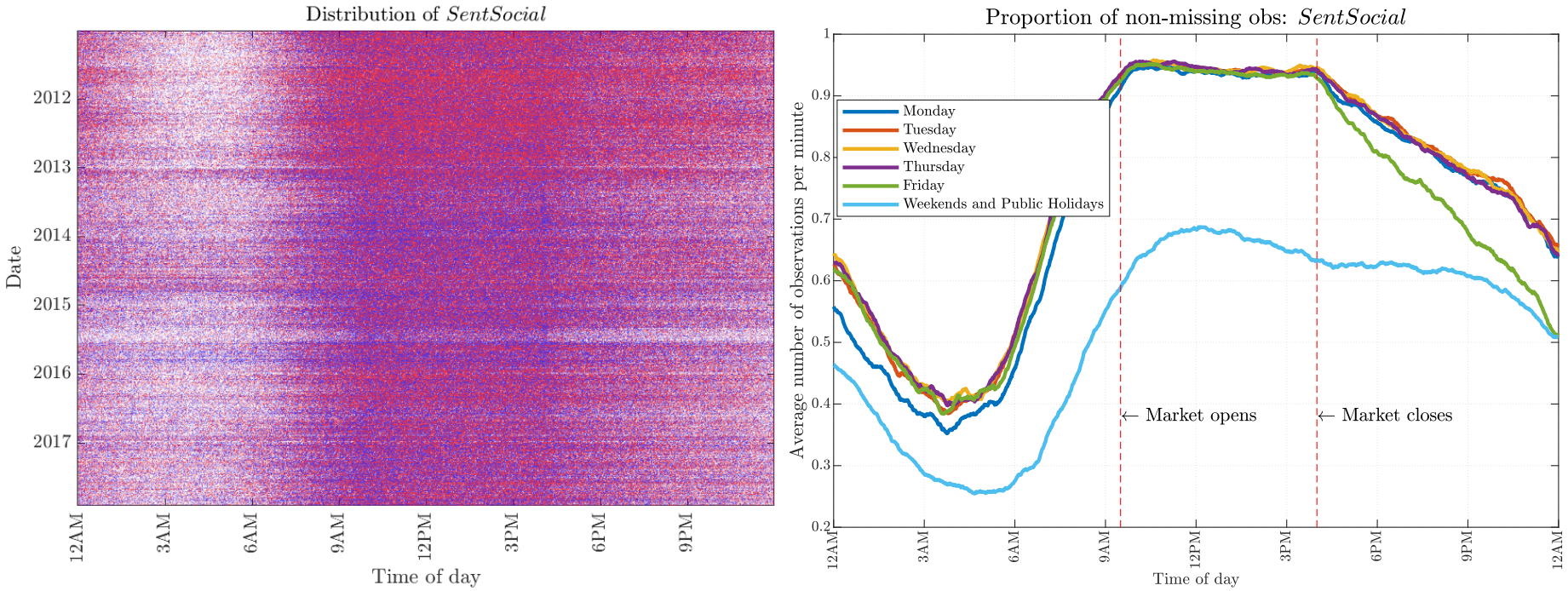


Figure: HEATMAP: Each pixel represents a single 1-minute observation. Positive values in red, negative values in blue, missing data as blank. A mixture of positive and negative sentiment scores brings out a purple hue, attesting to the frequent reversal in sentiment polarity.

High Frequency Sentiment Data

▸ News media surface ▸ Social media surface

Broad Market Index (S&P500 proxy) - News media, 1min

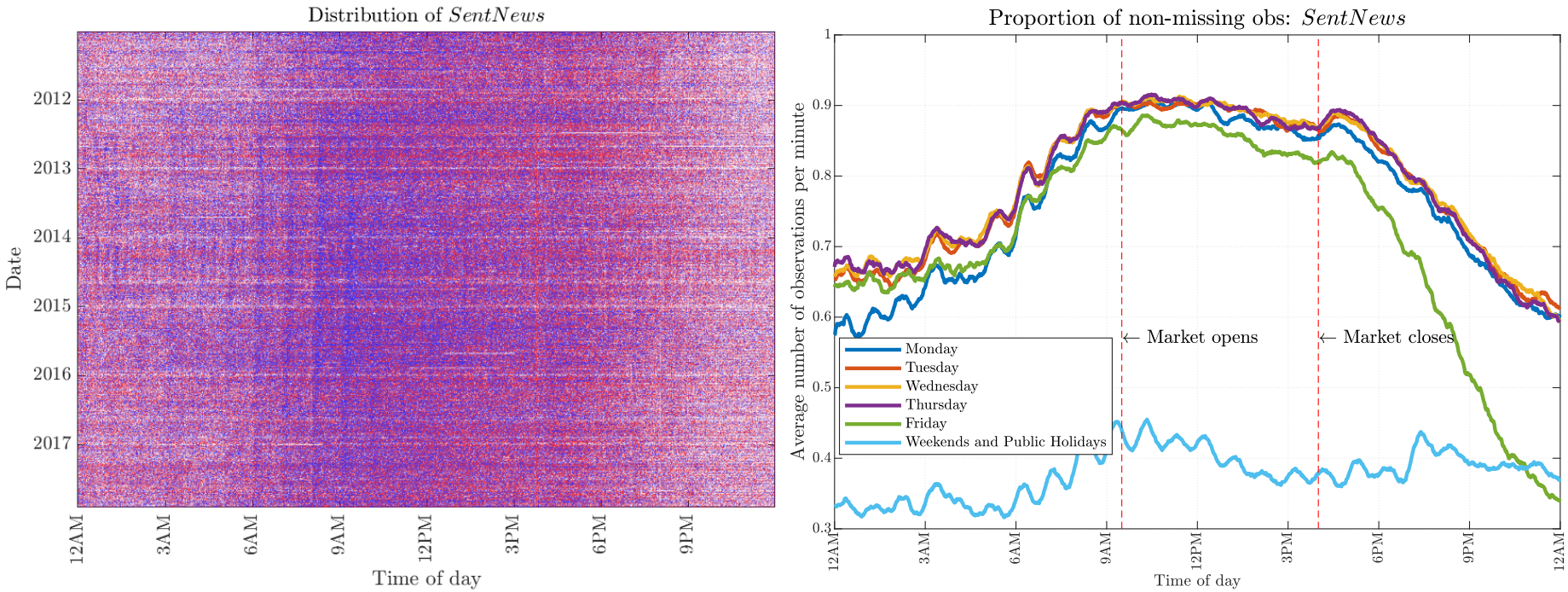
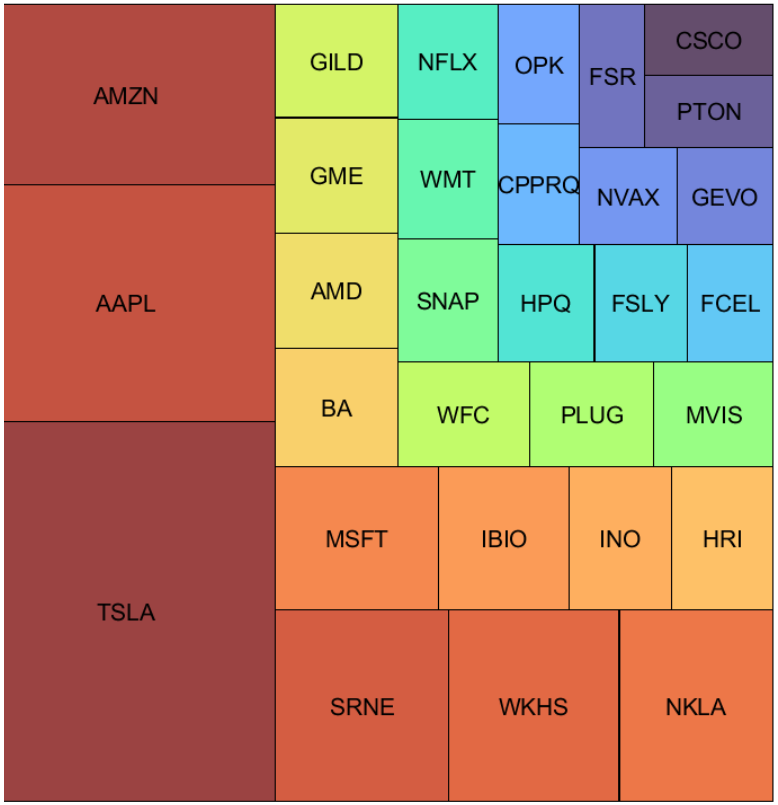


Figure: HEATMAP: Note pronounced threads weaving through mornings 'on-the-hour' (i.e., pronounced ridges at the 6:00, 7:00, 8:00 and 9:00 marks).

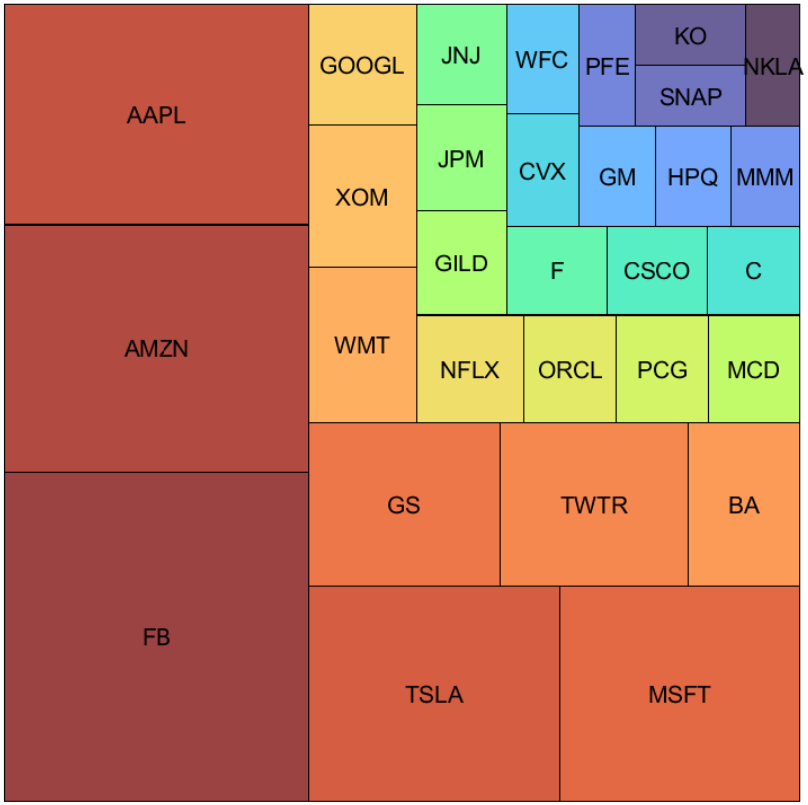
Most salient stocks

United States

Social Buzz: The Top 30 companies



News Buzz: The Top 30 companies



Sentiment Breakdown by Emotions

Social Media vs News

Gan et al. (2024)

MarketPsych Sentiment Indexes from Refinitiv/LSEG data

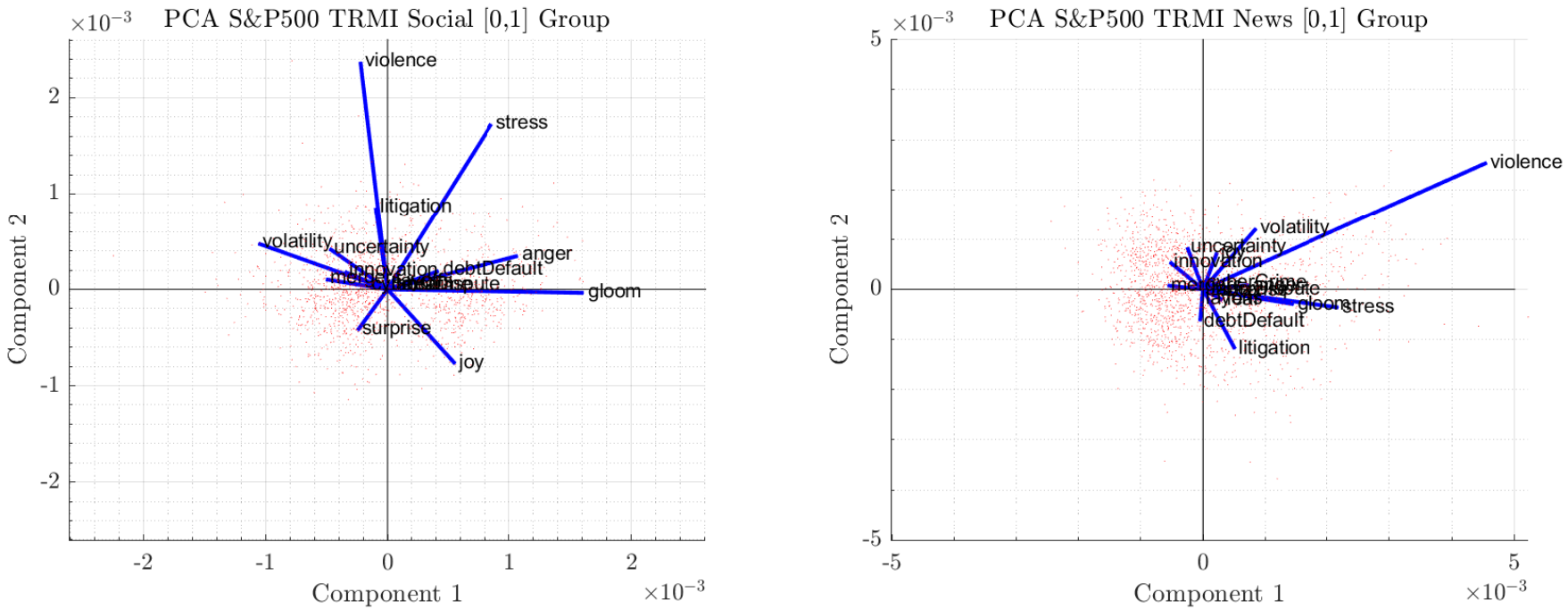


Figure: Principal Component Analysis (PCA) of the Sentiment Indices: Biplots of the first two principal component coefficients.

Growing importance of social media

- The US Securities and Exchange Commission (SEC) issued a guidance in 2008 admitting that **corporate websites can serve as an effective means for disseminating information** to investors.
- The SEC pointed out in its investigation report toward Netflix that "company **communications made through social media channels could constitute selective disclosures** and, therefore, require careful Regulation Fair Disclosure (Reg FD) analysis".
- In June 2015, the SEC further announced that "a start-up firm can post Twitter message about its stock or debt offering to gauge interest among potential investors", marking, for the first time, its **official acceptance of social media as information dissemination channel**.

Daily *Buzz*

Define 'dominating' or 'leading' series as follow:

- in an off-diagonal coefficients plot of a two-variable rolling-horizon VAR system,
- if one coefficient is significant, the other coefficient is insignificant,
- then the significant series 'leads' or 'dominates' the insignificant series.

VAR(1) model in scalar form:

$$\begin{aligned} Buzz_{S,t} &= \phi_{S,0} + \Phi_{1,1} Buzz_{S,t-1} + \Phi_{1,2} Buzz_{N,t-1} + \epsilon_{1,t}, \\ Buzz_{N,t} &= \phi_{N,0} + \Phi_{2,1} Buzz_{S,t-1} + \Phi_{2,2} Buzz_{N,t-1} + \epsilon_{2,t}. \end{aligned} \tag{2}$$

Same approach used in Victor DeMiguel et al. (2014). "Stock return serial dependence and out-of-sample portfolio performance". In: *The Review of Financial Studies* 27.4, pp. 1031–1073.

Daily *Buzz*

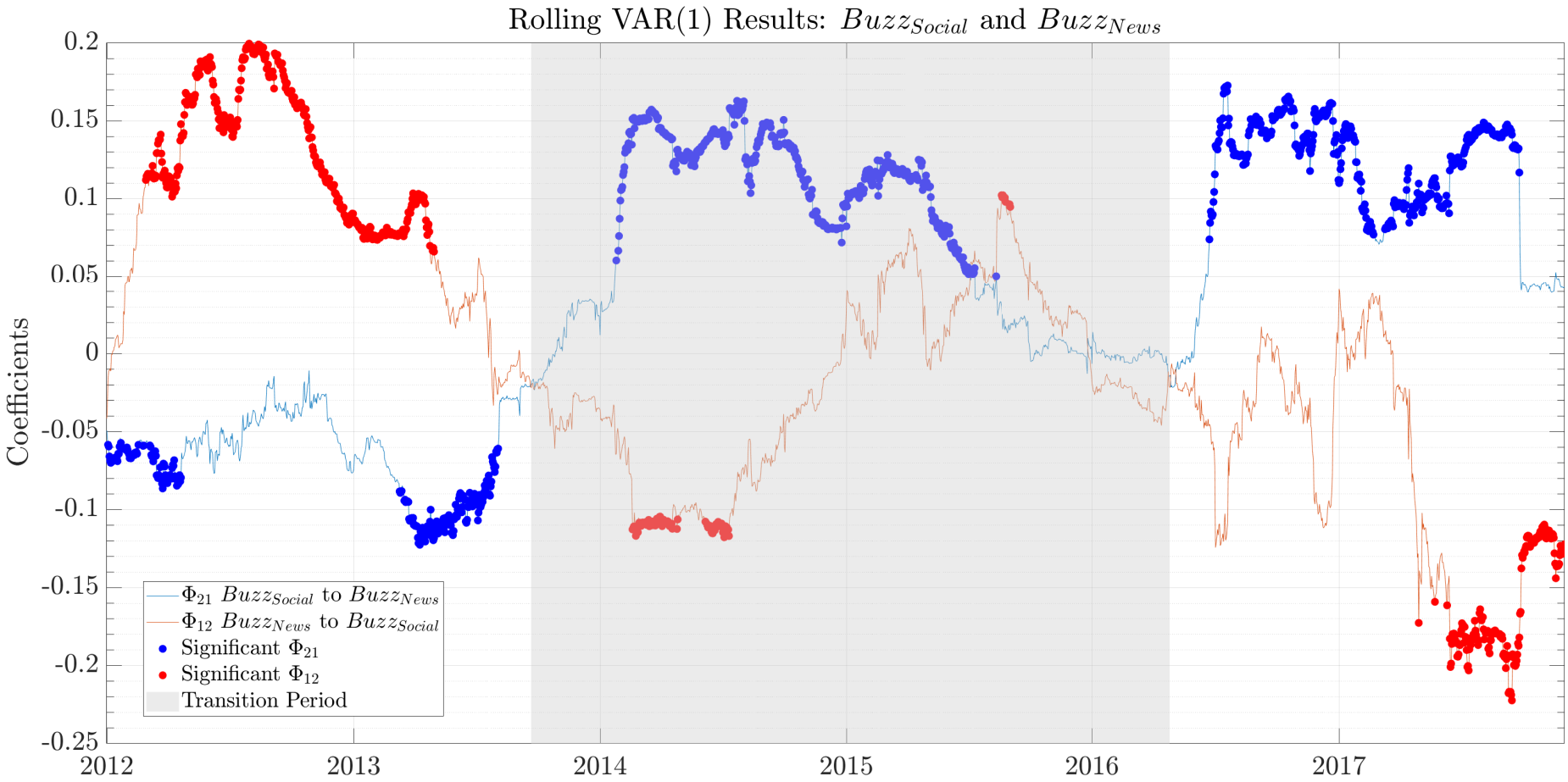
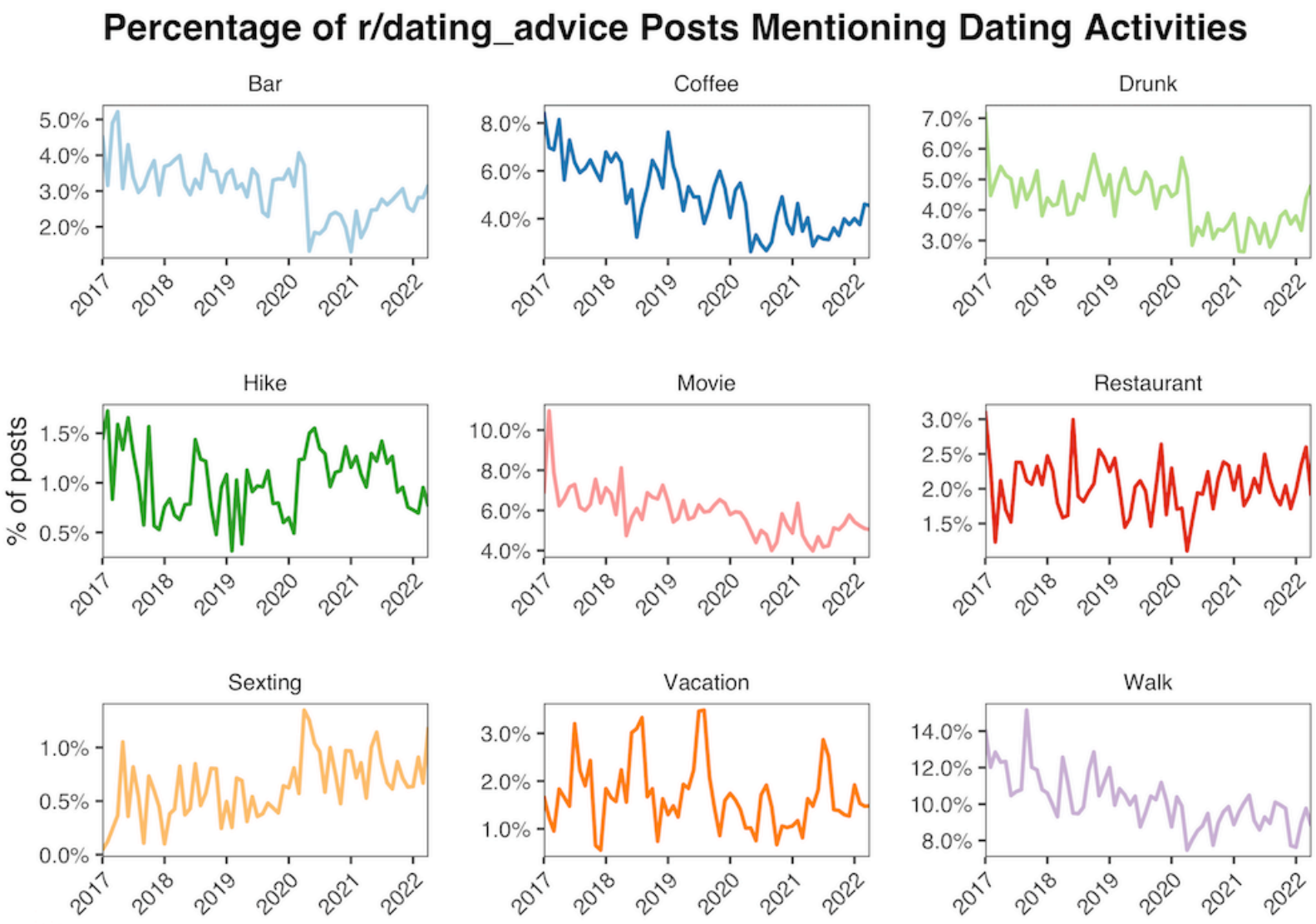


Figure: Rolling Window VAR(1) Off-Diagonal Elements - daily *Buzz*. Estimated effect of previous $Buzz_S$ on current $Buzz_N$ and estimated effect of previous $Buzz_N$ on current $Buzz_S$. For details, see Gan et al. (2020).

What We Talk About When We Talk About Dating

details



Why anomaly detection in textual data?

- Identifying fraudulent activities by analyzing **irregular patterns in communication** and records.
- Early detection of financial misconduct and **compliance breaches**.
- Aids in monitoring **market sentiment** and detecting unusual shifts, which can be indicative of market manipulation or insider information leaks.
- Enhances the accuracy of risk assessment models by **incorporating qualitative data** (from news and social media posts) to identify potential financial crises or company-specific issues.
- Supports the **automation** of compliance monitoring (less reliance on manual checks; increased efficiency of regulatory oversight).
- Potentially, maintains market **integrity** and investor **confidence**.

Anomaly Detection and Compliance Monitoring

- **Definition:** Anomaly detection in textual data involves identifying unusual patterns or outliers in text-based datasets, which may indicate errors, fraud, or emerging trends. ▸ Text Complexity
- **Importance:** Detecting anomalies in textual data helps organizations proactively address potential issues, mitigate risks, and make informed decisions. In finance, it can reveal suspicious transactions, insider trading, or market manipulation.
- **Role of Compliance Monitoring:** In industries like finance, healthcare, and telecommunications, compliance monitoring ensures adherence to regulatory standards and internal policies. It involves continuously analyzing data to detect deviations from the norm.
- **Contribution to Effective Compliance Monitoring:**
  1. *Early Detection:* Anomaly detection algorithms can identify potential compliance breaches before they escalate, allowing for timely intervention.
  2. *Cost Reduction:* Automated anomaly detection reduces the need for manual monitoring, lowering operational costs.
  3. *Enhanced Accuracy:* Machine learning models can process vast amounts of textual data with higher precision than human analysts, reducing the risk of oversight.
  4. *Adaptability:* Anomaly detection systems can be trained to adapt to evolving regulatory environments and detect new types of non-compliant behavior.

AI and LLMs for fraud detection

- In 2017, J.P. Morgan presented the first disruptive AI-based software for processing financial document called COiN (COntract Intelligence).
- In 2021, OECD opened the AI Observatory on Fintech focusing on opportunities and risks. ▸ AIFinanceOECD 2021
- Europe and Italy have also gone in this direction, and one of the 11 Italian priorities in the National Strategic Program on Artificial Intelligence launched in November 2021, is indeed AI for banking, finance and insurance.

Examples of NLP Applications in Finance I

- **Sentiment Analysis:** Assessing the market sentiment from news articles and social media to predict stock price movements.
  - Johan Bollen et al. (2011). "Twitter mood predicts the stock market". In: *Journal of Computational Science* 2.1, pp. 1–8. DOI: 10.1016/j.jocs.2010.12.007
- **Fraud Detection:** Analyzing textual data to identify patterns indicative of fraudulent activities in financial transactions or financial statements.
  - Khaled Gubran Al-Hashedi and Pritheega Magalingam (2021). "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019". In: *Computer Science Review* 40, p. 100402. DOI: 10.1016/j.cosrev.2021.100402
- **Regulatory Compliance:** Automatically monitoring communications for adherence to financial regulations and detecting non-compliant behavior.
  - David F. Larcker and Anastasia A. Zakolyukina (2012). "Detecting Deceptive Discussions in Conference Calls". In: *Journal of Accounting Research* 50.2, pp. 495–540
- **Risk Management:** Analyzing textual reports and documents to identify potential risks and vulnerabilities in financial operations.
- **Detecting Fake Reviews:** Analyzing online reviews to identify patterns indicative of fraudulent reviews or review manipulation.

Examples of NLP Applications in Finance II

- Maria Cristina Arcuri et al. (2023). "Does fake news impact stock returns? Evidence from US and EU stock markets". In: *Journal of Economics and Business* 125-126, p. 106130. DOI: 10.1016/j.jeconbus.2023.106130
- William Yang Wang (2017). *"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection*. arXiv: 1705.00648 [cs.CL]
- Fact-checking LLM output ▸ Google DeepMind

- **Crisis Detection on Social Media:** Monitoring social media platforms to detect early signals of events like natural disasters or public health emergencies.
  - Navya Makkena et al. (2024). "Urgency Detection in Social Media Texts Using Natural Language Processing". In: *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pp. 156–163. DOI: 10.1109/ICSC59802.2024.00031
  - Rabindra Lamsal et al. (2024). "CrisisTransformers: Pre-trained language models and sentence encoders for crisis-related social media texts". In: *Knowledge-Based Systems* 296, p. 111916. DOI: https://doi.org/10.1016/j.knosys.2024.111916
  - State-of-the-art contextual and semantically meaningful sentence embeddings for crisis-related social media texts ▸ HuggingFace: CrisisTransformers

- **Financial News for Market Prediction:** Analyzing financial news articles and reports to identify anomalies that may impact stock prices or market trends.
  - Shimon Kogan et al. (2022). "Social Media and Financial News Manipulation". In: *Review of Finance* 27.4, pp. 1229–1268. DOI: 10.1093/rof/rfac058

Opinion vs Fact Detection in NLP

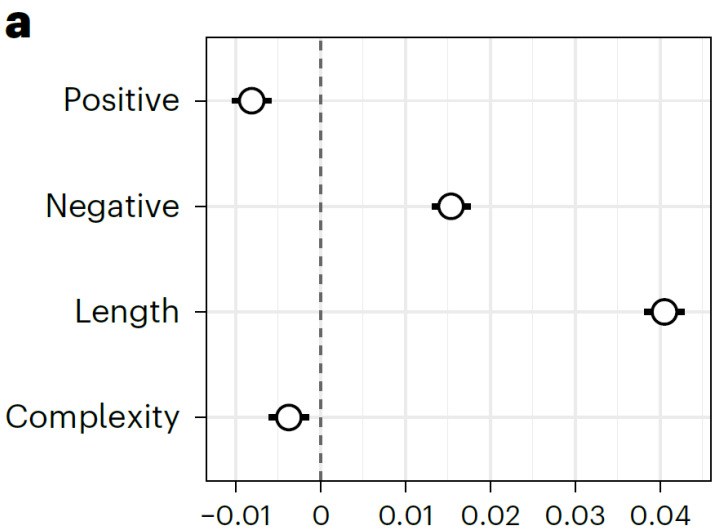
- The distinction is essential
  - News verification
  - Sentiment analysis
  - Factual accuracy in legal documents and compliance reports
- Challenges in Discerning Opinion and Fact
  - Subjectivity: Opinions are subjective and often express sentiments, emotions, or personal beliefs.
  - Objective Statements: Facts present verifiable information devoid of personal bias.
  - Ambiguity: Some statements can exhibit both subjective and factual elements.
- NLP Techniques for Opinion-Fact Classification
  - Text Classification: Supervised learning models to classify text as opinion, fact, or a mixture of both.
  - Sentiment Analysis: Detecting sentiment polarity (positive, negative, neutral) for opinion identification.
  - Linguistic Features: Analyzing linguistic cues like sentiment words, adjectives, and adverbs.
  - Contextual Analysis: Understanding the context surrounding a statement aids in discernment.

▸ Quiz: Fact or Opinion?
▸ ABC's Fact-Opinion Analysis

Negativity drives online news consumption

Claire E. Robertson et al. (2023). "Negativity drives online news consumption". In: *Nature Human Behaviour* 7.5, pp. 812–822.
DOI: 10.1038/s41562-023-01538-4. URL: https://doi.org/10.1038%2Fs41562-023-01538-4

Causal effect of negative and emotional words on news consumption using a large online dataset of viral news stories.



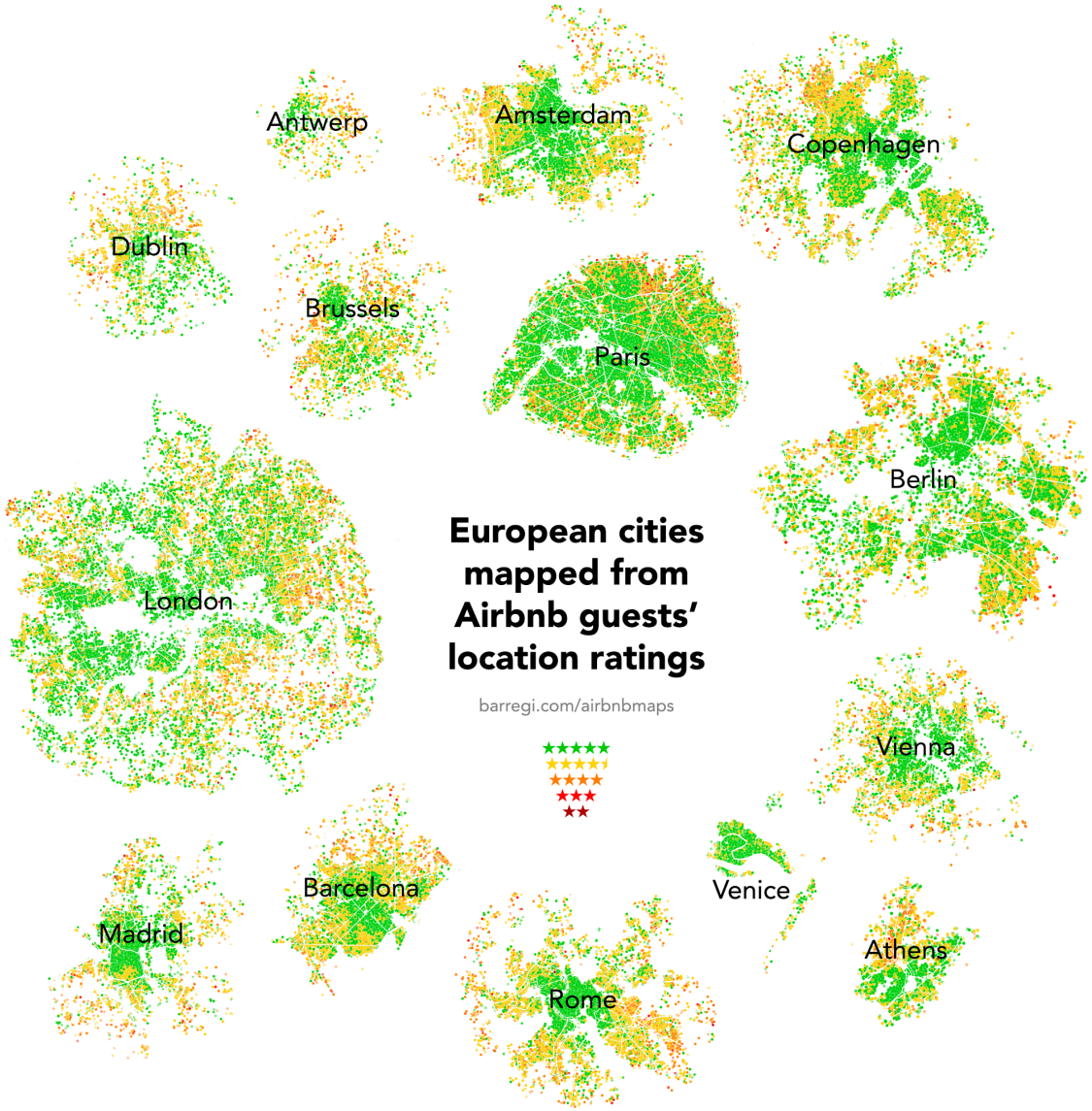
- Online media is important for society in informing and shaping opinions.
- Has a profound impact across marketing, finance, health and politics.
- Results allow better understanding of why users engage with online media.
- For a headline of average length, each additional negative word increased the click-through rate by 2.3%.

Method: Logit regression

Data: Publicly available structured data

City maps from Airbnb location ratings

▸ Interactive ▸ Data



Maps of tourists' **perceptions** of different urban areas through **sentiment analysis of reviews** retrieved from Airbnb.

The data reveals interesting geographic patterns and exposes subjective perceptions on safety, upkeep or convenience.

Many cities, for example, check out Sydney.

Airbnb Review Sentiments: NLP for Anomaly Detection

Objective: Detect underreported negative Airbnb experiences and identify critical service issues at a fine-grained geographic level in Rome. Frigau et al. (2024)

NLP Pipeline:

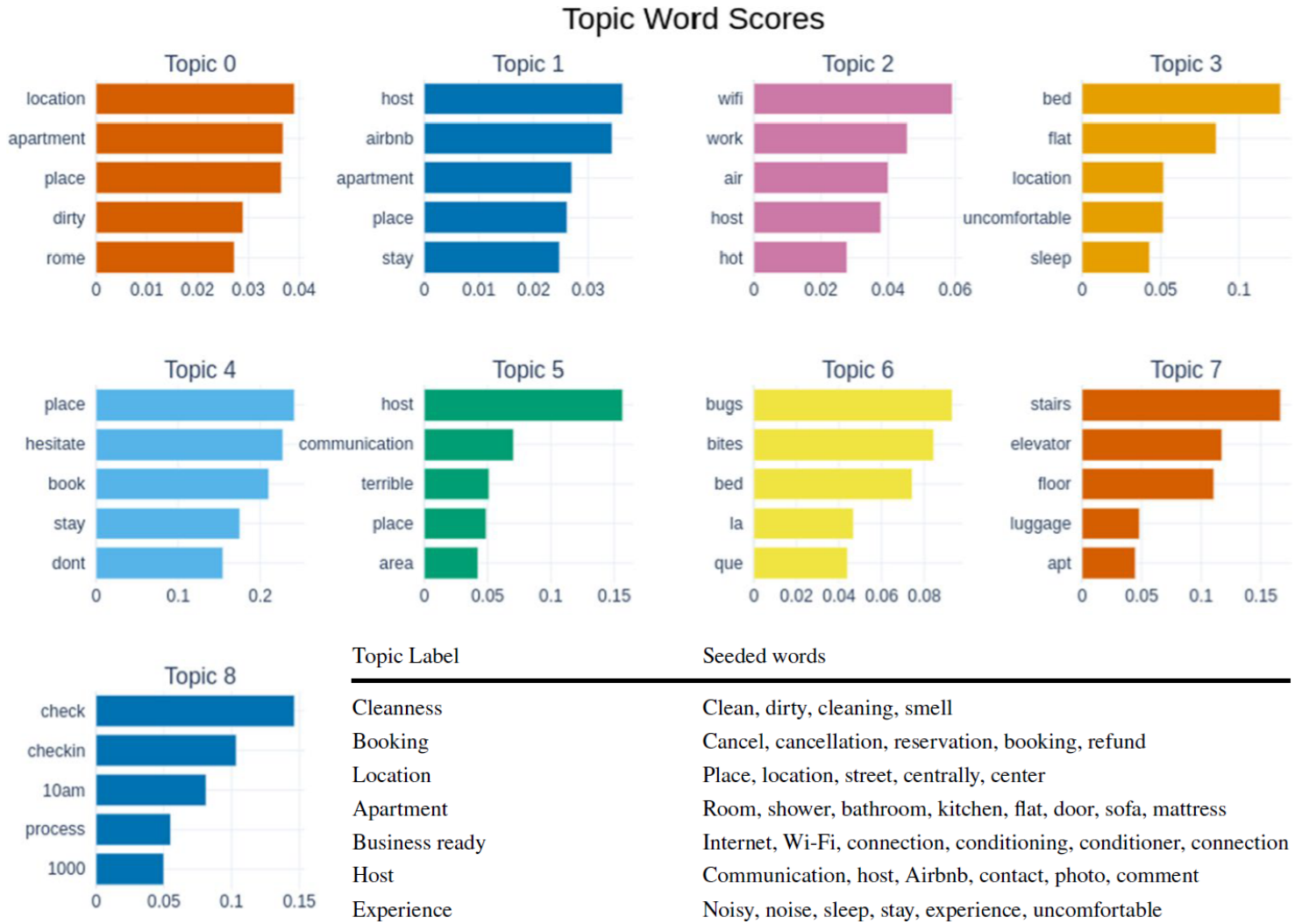
- **Sentiment Analysis**:
  - Fine-tuned `BERT` model to classify reviews as *positive* or *negative*.
  - Used review ratings from TripAdvisor as ground-truth labels.
- **Topic Modeling**:
  - Applied `BERTopic` to negative reviews.
  - Extracted interpretable complaint themes (e.g., Wi-Fi, cleanliness, noise, host communication).

Outcome:

- Estimated sentiment and topic prevalence across 226 Airbnb subdomains (geo-area $\times$ host type).
- Identified geographic hotspots of dissatisfaction — an NLP-driven anomaly detection approach.

Key Insight: NLP helps reveal critical user concerns and anomalies that traditional metrics miss.

Airbnb Review Sentiments: NLP for Anomaly Detection



Topic Label	Seeded words
Cleanness	Clean, dirty, cleaning, smell
Booking	Cancel, cancellation, reservation, booking, refund
Location	Place, location, street, centrally, center
Apartment	Room, shower, bathroom, kitchen, flat, door, sofa, mattress
Business ready	Internet, Wi-Fi, connection, conditioning, conditioner, connection
Host	Communication, host, Airbnb, contact, photo, comment
Experience	Noisy, noise, sleep, stay, experience, uncomfortable

Source: Frigau et al. (2024)

Redefining Sentiment in Finance

The Colour of Finance Words

▸ Garcia et al. (2023)

What's new? This paper rethinks sentiment analysis by creating a **data-driven sentiment lexicon** based on *market reactions*, not human-labelled dictionaries.

▸ YouTube: Warp Speed Price Moves: Jumps after Earnings Announcements

Key Contributions:

- Identify uni/bi-grams from financial texts (e.g., earnings calls) associated with positive or negative stock returns.
- Constructs sentiment dictionaries where word polarity is **inferred from return predictiveness**, not intuition.
- Highlights how some classic "positive" or "negative" words from existing dictionaries (e.g., Loughran-McDonald) may be **misclassified**.
- Demonstrates improved explanatory power of these ML-generated dictionaries.

Application:

- **Earnings calls**, 10-K filings, and news articles processed.
- Sentiment score derived from learned dictionaries significantly predicts abnormal returns.

Why it matters? Traditional lexicons lack context and misjudge nuanced financial language. This approach **aligns sentiment scoring with real financial impact**.

Measuring Textual Complexity

Readability scores

Why Measure Textual Complexity?

- Quantifying readability and complexity helps assess how accessible or opaque financial texts are.
- Crucial in compliance, regulatory disclosures, and corporate communication.
- Allows detection of intentional obfuscation or overly simplified narratives.
- Used in auditing, risk assessment, and detecting deception or manipulation.

Use in Manipulation and Anomaly Detection

- High complexity can indicate attempts to hide bad news or mislead stakeholders.
- Sudden changes in complexity may signal irregularities or a shift in disclosure strategy.
- Useful in detecting greenwashing, selective disclosure, and obfuscation in ESG reports.
- Applied in machine learning models as features for anomaly or fraud detection.

Applications in Finance Research

- Lower readability in 10-K filings linked to **lower firm valuation**.
  - Feng Li (2008). "Annual report readability, current earnings, and earnings persistence". In: *Journal of Accounting and Economics* 45.2, pp. 221–247. DOI: https://doi.org/10.1016/j.jacceco.2008.02.003
- **Domain-specific readability** and sentiment dictionaries for financial text.
  - Tim Loughran and Bill McDonald (2014). "Measuring Readability in Financial Disclosures". In: *The Journal of Finance* 69.4, pp. 1643–1671. DOI: https://doi.org/10.1111/jofi.12162
- Complex language in earnings calls linked to **lower analyst forecast accuracy**.
  - Brian J. Bushee et al. (2018). "Linguistic Complexity in Firm Disclosures: Obfuscation or Information?" In: *Journal of Accounting Research* 56.1, pp. 85–121. DOI: https://doi.org/10.1111/1475-679X.12179
- Fraud recognition is based on the text of the report rather than financial data.
- Although text of the report is usually ignored, text complexity improves the empirical effectiveness of fraud recognition and **predicts the probability of restatements and fraud**.
  - Yi Zhang et al. (2024). "Corporate fraud detection based on linguistic readability vector: Application to financial companies in China". In: *International Review of Financial Analysis* 95, p. 103405. DOI: https://doi.org/10.1016/j.irfa.2024.103405

Overview of Readability Metrics

Python implementation in TextStat package

Most metrics are based on: **sentence length**, **word length**, **syllables**, or known **word lists**.

Focuses on syllables and sentence complexity:

- **Flesch Reading Ease:** Higher scores = easier text. Based on sentence length and syllables per word.
- **Flesch-Kincaid Grade:** Converts ease score to U.S. school grade level.
- **SMOG Index:** Focused on polysyllabic words. Good for short texts.
- **Gunning Fog Index:** Based on sentence length and complex word proportion. Interactive

Character-based; useful when syllables are hard to compute:

- **Coleman-Liau Index:** Based on characters per word and words per sentence.
- **Automated Readability Index:** Similar to Coleman-Liau but uses characters per word and sentence length.

Vocabulary-based — highlight domain-specific complexity:

- **Dale-Chall Score:** Compares text words to a list of common words. Penalises unfamiliar terms.
- **Difficult Words:** Counts words not on common word lists.
- **Linsear Write Formula:** Designed for US Air Force manuals. Uses sentence and word difficulty.

Aggregate indices:

- **Text Standard:** Aggregates multiple indices to determine standard grade level.

UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Practical Advice

- Readability metrics are **useful signals** for anomaly, fraud, and complexity in financial texts.
- No single metric is perfect — use a **combination**.
- Always **interpret in context**: complex text may reflect technical detail, not deception.
- Use these metrics as features in NLP pipelines for predictive modelling.
- Use **state-of-the-art** context-aware **transformer-based models** like HuggingFace: Multilingual Readability Assessment

Manipulation of public perception

Full Story

Anomaly Detection in Textual Data: the #RacistBanksMustFall hashtag case

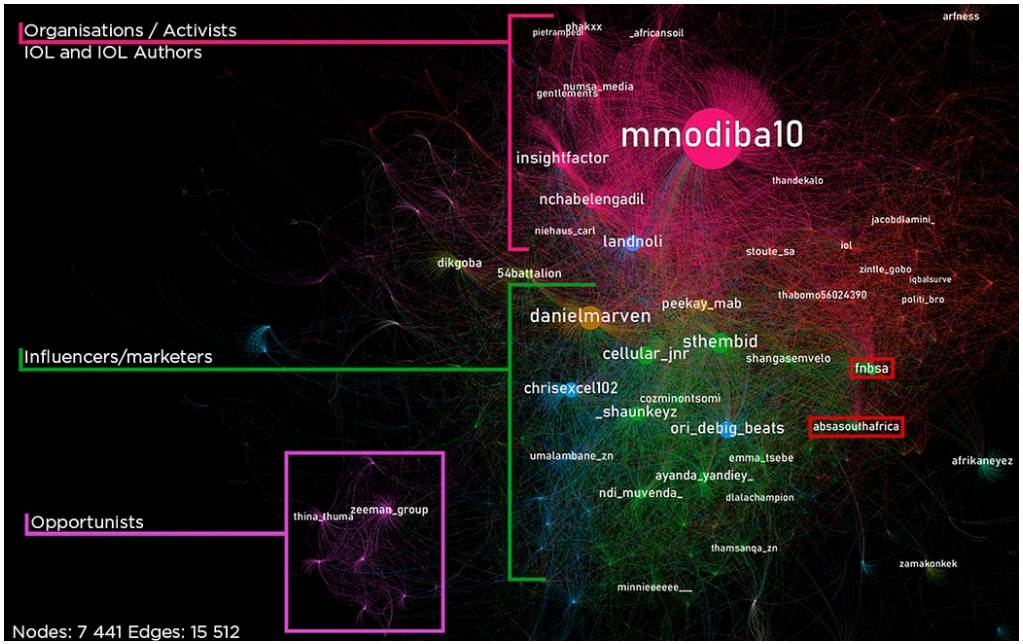
Problem: A coordinated social media campaign targeted South African banks in April 2021 under the guise of anti-racist activism. Paid influencers, sockpuppets, and digital marketers amplified hashtags to manipulate public perception.

How AI and Textual Analysis Helped:

- Detected unnatural engagement spikes on social media.
- Identified networks of fake or paid accounts.
- Tracked temporal patterns of hashtag activity.

Challenges:

- Distinguishing real activism from artificial amplification.
- Detecting sockpuppets using evasive strategies.



Manipulation of public perception

Methods for Text-Based Anomaly Detection

Techniques Used:

- **Network Analysis:** Identified clusters of interconnected influencers and paid accounts.
- **NLP & Sentiment Analysis:** Detected artificial sentiment shifts and non-organic engagement.
- **Time Series Analysis:** Tracked unnatural spikes in hashtag usage.
- **Machine Learning (ML) for Fake Account Detection:**
  - *Supervised Learning:* Classified sockpuppets using historical patterns.
  - *Unsupervised Learning:* Clustered anomalous user behavior.
- **Bot Detection Algorithms:** Analyzed metadata (creation dates, posting frequency, account interactions).

Outcome:

- Unveiled a network of fake influencers and manipulated activism.
- Showed that the campaign was commercially, not socially, motivated.

Complex mixture of facts and opinions

Predicting Gentrification with NLP and Social Media

Details

Traditional Challenge: Census data is often outdated and lacks real-time granularity to capture urban change like gentrification.

Solution: Digital media biases may help uncover hard-to-measure processes like gentrification.

NLP + Social Media = New Insights

- **Twitter + Foursquare** check-ins from **37,000 users** across **42,000 venues in London**.
- Applied **textual analysis** and **geo-social network modelling**.
- Analysed **500,000+ check-ins** to measure:
  - Brokerage, Serendipity, Entropy, Homogeneity (Social Diversity Metrics)
- Correlated these patterns with the **UK Index of Multiple Deprivation**.

Findings: Areas with low economic status + high social diversity showed clear signs of gentrification (e.g., Hackney, Lambeth).

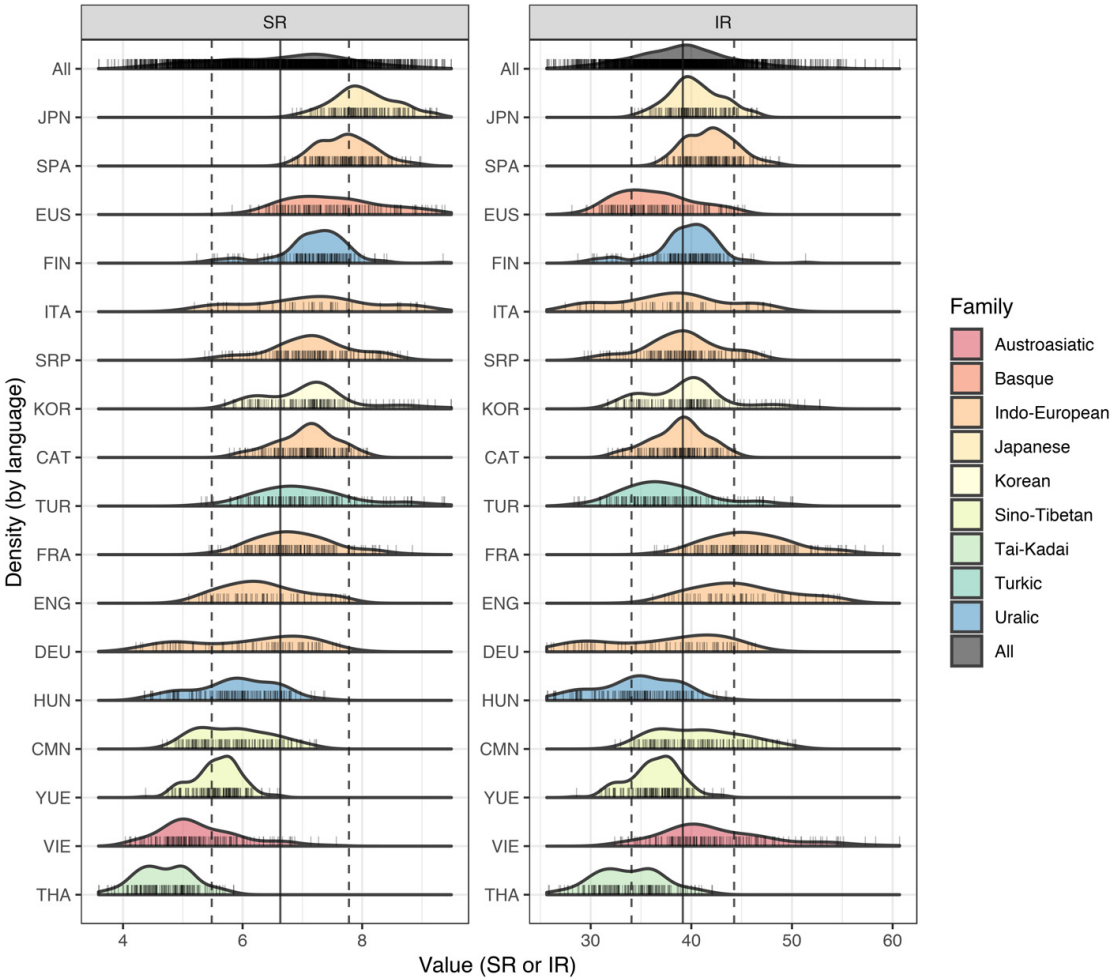
NLP Applications:

- Real-time urban trend monitoring
- Sentiment + location signals from digital footprints
- Classifying check-in language and content

Policy Implications:

- Helps identify *early signs of gentrification*
- Data-driven urban planning Details
- Tailored social protection strategies

Languages and Encoding Efficiency



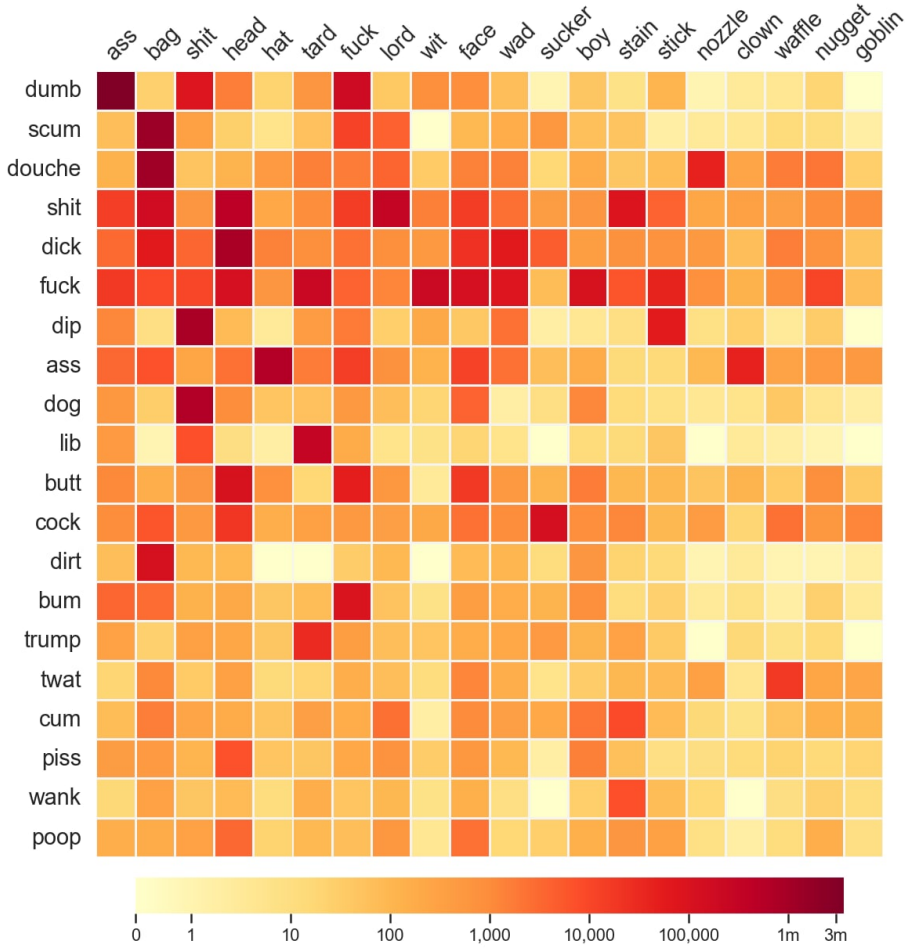
Christophe Coupe et al. (2019). "Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche". In: *Science Advances* 5.9, eaaw2594. DOI: 10.1126/sciadv.aaw2594

- Assess cross-linguistic variation
- 'Informativeness' using Shannon entropy
- SR = Speech Rate (in syllables/sec)
- IR = Information Rate (in bits/sec)

Method: NLP, linear mixed-effects regression (sex as fixed effect and text, language, and speaker as randomeffects)

Data: Publicly available oral and text corpora

Analysis of Compound Pejoratives



- **Textual complexity** is on the rise (even for derogatory terms).
- Corpus is based on **Reddit**, which has the virtue of being uninhibited in its profanity.
- Measures of statistical diversity using **Shannon entropy**.
- **Zipf's Law** visualisation of log-frequency vs log-rank.
- Includes analysis of insults based on political orientation.

Data and **Python Code**

Another interesting example from the same author: *The size of things*: an **ngram** experiment on the most common objects used in analogies of the form "the size of a ..." in English books in 1800s, 1900s, and 2000s.

Recommended Review Articles for Quick Insights

1 Common ML techniques:

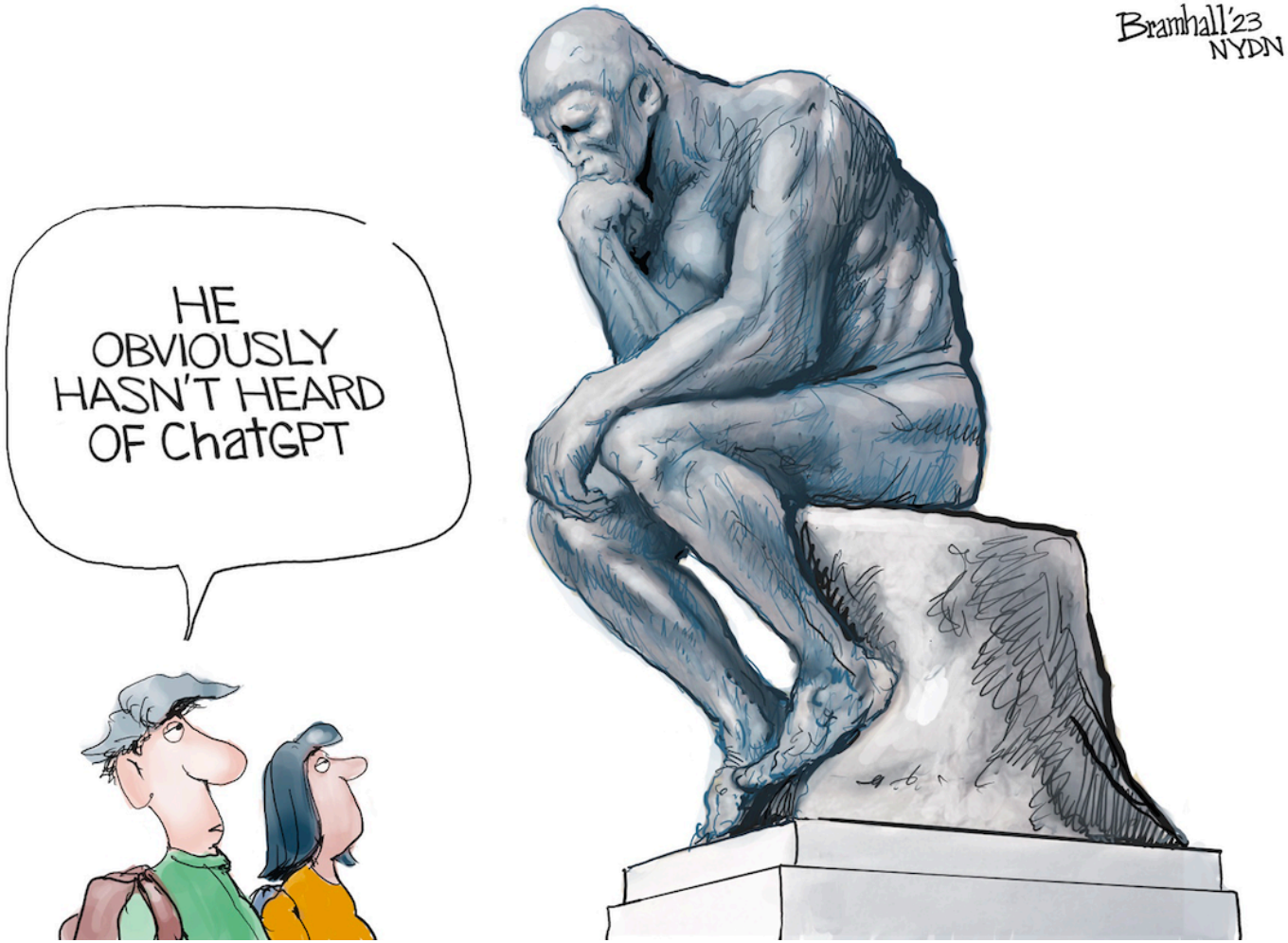
- Susan Athey and Guido W. Imbens (2019). "Machine Learning Methods That Economists Should Know About". In: *Annual Review of Economics* 11.1, pp. 685–725. DOI: 10.1146/annurev-economics-080217-053433
- Vaishak Belle and Ioannis Papantonis (2021). "Principles and Practice of Explainable Machine Learning". In: *Frontiers in Big Data* 4. ISSN: 2624-909X. DOI: 10.3389/fdata.2021.688969

2 NLP:

- Bethany Percha (2021). "Modern Clinical Text Mining: A Guide and Review". In: *Annual Review of Biomedical Data Science* 4.1, pp. 165–187. DOI: 10.1146/annurev-biodatasci-030421-030931
- Bijitaswa Chakraborty and Titas Bhattacharjee (2020). "A review on textual analysis of corporate disclosure according to the evolution of different automated methods". In: *Journal of Financial Reporting and Accounting* 18.4, pp. 757–777. DOI: 10.1108/jfra-02-2020-0047
- Tim Loughran and Bill McDonald (2020). "Textual Analysis in Finance". In: *Annual Review of Financial Economics* 12.1, pp. 357–375. DOI: 10.1146/annurev-financial-012820-032249

3 Fraud Detection:

- Shimon Kogan et al. (2022). "Social Media and Financial News Manipulation". In: *Review of Finance* 27.4, pp. 1229–1268. DOI: 10.1093/rof/rfac058
- Muhammad Atif Khan Achakzai and Peng Juan (2022). "Using machine learning Meta-Classifiers to detect financial frauds". In: *Finance Research Letters* 48, p. 102915. DOI: 10.1016/j.frl.2022.102915
- Khaled Gubran Al-Hashedi and Pritheega Magalingam (2021). "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019". In: *Computer Science Review* 40, p. 100402. DOI: 10.1016/j.cosrev.2021.100402


Bramhall'23
NYDN
HE OBVIOUSLY HASN'T HEARD OF ChatGPT

References I

Achakzai, Muhammad Atif Khan and Peng Juan (2022). "Using machine learning Meta-Classifiers to detect financial frauds". In: *Finance Research Letters* 48, p. 102915. DOI: 10.1016/j.frl.2022.102915.

Al-Hashedi, Khaled Gubran and Pritheega Magalingam (2021). "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019". In: *Computer Science Review* 40, p. 100402. DOI: 10.1016/j.cosrev.2021.100402.

Arcuri, Maria Cristina et al. (2023). "Does fake news impact stock returns? Evidence from US and EU stock markets". In: *Journal of Economics and Business* 125-126, p. 106130. DOI: 10.1016/j.jeconbus.2023.106130.

Athey, Susan and Guido W. Imbens (2019). "Machine Learning Methods That Economists Should Know About". In: *Annual Review of Economics* 11.1, pp. 685–725. DOI: 10.1146/annurev-economics-080217-053433.

Belle, Vaishak and Ioannis Papantonis (2021). "Principles and Practice of Explainable Machine Learning". In: *Frontiers in Big Data* 4. ISSN: 2624-909X. DOI: 10.3389/fdata.2021.688969.

Blei, David M. (2012). "Probabilistic topic models". In: *Commun. ACM* 55.4, pp. 77–84. DOI: 10.1145/2133806.2133826.

Bollen, Johan et al. (2011). "Twitter mood predicts the stock market". In: *Journal of Computational Science* 2.1, pp. 1–8. DOI: 10.1016/j.jocs.2010.12.007.

Bushee, Brian J. et al. (2018). "Linguistic Complexity in Firm Disclosures: Obfuscation or Information?" In: *Journal of Accounting Research* 56.1, pp. 85–121. DOI: https://doi.org/10.1111/1475-679X.12179.

References II

Chakraborty, Bijitaswa and Titas Bhattacharjee (2020). "A review on textual analysis of corporate disclosure according to the evolution of different automated methods". In: *Journal of Financial Reporting and Accounting* 18.4, pp. 757–777. DOI: 10.1108/jfra-02-2020-0047.

Cook, Michael et al. (2019). *The Ubiquity of Data: Challenges and Opportunities for Asset Managers*. Lazard Asset Management. URL: https://www.lazardassetmanagement.com/us/en_us/research-insights/perspectives/the_ubiquity_of_data.

Coupe, Christophe et al. (2019). "Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche". In: *Science Advances* 5.9, eaaw2594. DOI: 10.1126/sciadv.aaw2594.

DeMiguel, Victor et al. (2014). "Stock return serial dependence and out-of-sample portfolio performance". In: *The Review of Financial Studies* 27.4, pp. 1031–1073.

Frigau, Luca et al. (2024). "Gauging Airbnb review sentiments and critical key-topics by small area estimation". In: *Statistical Methods & Applications* 33.4, pp. 1145–1170. DOI: 10.1007/s10260-024-00764-y.

Gan, Baoqing et al. (2019). *Investor Sentiment Under the Microscope*. Working Paper. University of Technology Sydney. URL: https://acfr.aut.ac.nz/__data/assets/pdf_file/0005/328928/B-Gan-Investor-Sentiment-Under-a-Microscope_27Nov2019.pdf.

Gan, Baoqing et al. (2020). "Sensitivity to sentiment: News vs social media". In: *International Review of Financial Analysis* 67, p. 101390.

Gan, Baoqing et al. (2024). "Tweets versus broadsheets: Sentiment impact on stock markets around the world". In: *Journal of Financial Research* 47, pp. 601–633.

References III


Garcia, Diego et al. (2023). "The colour of finance words". In: *Journal of Financial Economics* 147.3, pp. 525–549. DOI: https://doi.org/10.1016/j.jfineco.2022.11.006.


Kogan, Shimon et al. (2022). "Social Media and Financial News Manipulation". In: *Review of Finance* 27.4, pp. 1229–1268. DOI: 10.1093/rof/rfac058.


Lamsal, Rabindra et al. (2024). "CrisisTransformers: Pre-trained language models and sentence encoders for crisis-related social media texts". In: *Knowledge-Based Systems* 296, p. 111916. DOI: https://doi.org/10.1016/j.knosys.2024.111916.


Larcker, David F. and Anastasia A. Zakolyukina (2012). "Detecting Deceptive Discussions in Conference Calls". In: *Journal of Accounting Research* 50.2, pp. 495–540.


Li, Feng (2008). "Annual report readability, current earnings, and earnings persistence". In: *Journal of Accounting and Economics* 45.2, pp. 221–247. DOI: https://doi.org/10.1016/j.jacceco.2008.02.003.


Loughran, Tim and Bill McDonald (2014). "Measuring Readability in Financial Disclosures". In: *The Journal of Finance* 69.4, pp. 1643–1671. DOI: https://doi.org/10.1111/jofi.12162.


— (2020). "Textual Analysis in Finance". In: *Annual Review of Financial Economics* 12.1, pp. 357–375. DOI: 10.1146/annurev-financial-012820-032249.


Makkena, Navya et al. (2024). "Urgency Detection in Social Media Texts Using Natural Language Processing". In: *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pp. 156–163. DOI: 10.1109/ICSC59802.2024.00031.

References IV

Percha, Bethany (2021). "Modern Clinical Text Mining: A Guide and Review". In: *Annual Review of Biomedical Data Science* 4.1, pp. 165–187. DOI: 10.1146/annurev-biodatasci-030421-030931.

Peterson, RL (2013). "Thomson Reuters MarketPsych Indices (TRMI) White Paper". In: *Inside the Mind of the Market*.

Robertson, Claire E. et al. (2023). "Negativity drives online news consumption". In: *Nature Human Behaviour* 7.5, pp. 812–822. DOI: 10.1038/s41562-023-01538-4. URL: https://doi.org/10.1038%2Fs41562-023-01538-4.

Wang, William Yang (2017). *"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection*. arXiv: 1705.00648 [cs.CL].

Zhang, Yi et al. (2024). "Corporate fraud detection based on linguistic readability vector: Application to financial companies in China". In: *International Review of Financial Analysis* 95, p. 103405. DOI: https://doi.org/10.1016/j.irfa.2024.103405.