JMAN GROUP

Natural Language Processing

14 March 2025

jmangroup.com

FT FINANCIAL TIMES
statista
UK'S LEADING MANAGEMENT CONSULTANTS 2022

Content

1 NLP Introduction

2 Applications of NLP

3 Levels of NLP

4 Core NLP Tasks

5 NLP Techniques

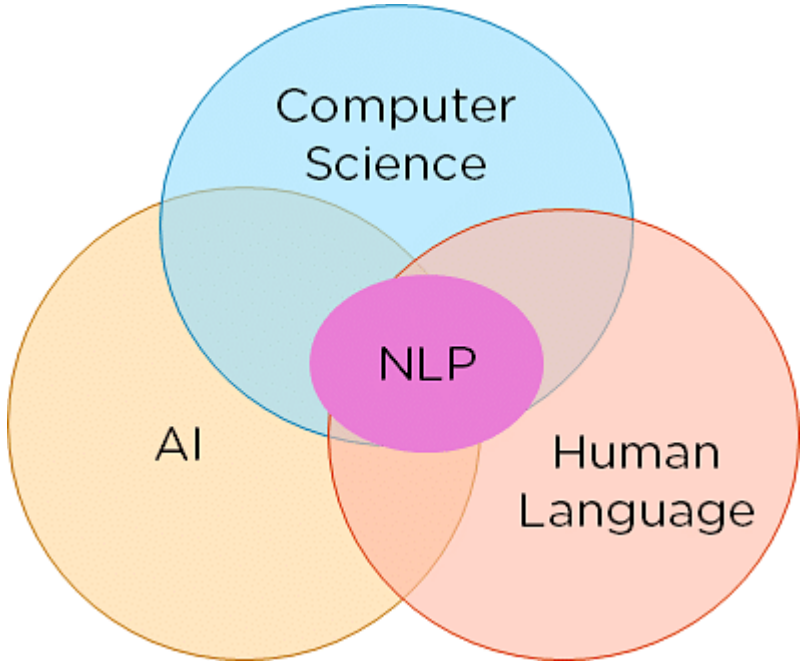
6 Working of NLP

What is NLP?

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that enables computers to understand, interpret, and respond to human language. It combines linguistics, machine learning, and computational techniques to process and analyze text and speech.

Examples of NLP around us-

- Virtual Assistants
- Chatbots
- Spell Check & Auto-Correction
- Speech-to-Text Conversion
- Search Engines
- Machine Translation
- Spam Filters



Applications of NLP

NLP is transforming the way machines understand human communication. Its applications are diverse, impactful, and essential to today's AI-driven solutions. NLP powers many real-world applications that bridge the gap between human language and machine understanding. Its capabilities are widely used across industries for automating tasks, extracting insights, and enhancing user experiences.

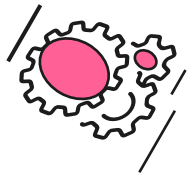
NLP is Specialized in multiple fields such as Healthcare, Finance, Education etc…

Sentiment Analysis

Determines the emotional tone of a text, whether it is positive, negative, or neutral.

Applications:

- Analysing customer feedback and reviews
- Social media monitoring for brand sentiment
- Assessing public opinion on political events



Named Entity Recognition

Identifies and categorizes named entities in text, such as people, organizations, locations, dates, and more.

Applications:

- Extracting company names from financial reports
- Identifying key entities in legal documents
- Automated news categorization

Text Summarization

Generates concise summaries of long texts while retaining key information.

Applications:

- Summarizing news articles for quick reading
- Generating executive summaries for business reports
- Reducing lengthy research papers into key points

Machine Translation

Automatically translates text from one language to another while preserving meaning and context.

Applications:

- Translating websites and news content
- Assisting global businesses in multilingual communication
- Enabling cross-language medical or legal documentation

Importance of NLP in Tech Services and PE

The tech service industry deals with massive volumes of unstructured text (reports, emails, contracts), NLP automates analysis, extracting key information and patterns which in turns faster data processing, reduced manual effort, and quicker turnaround times as it enables data-driven insights for strategic decisions.

Automation

NLP automates customer service with chatbots. Virtual assistants improve efficiency.

Data Insights

NLP enhances data analysis from unstructured text. It reveals valuable insights.

Search Improvement

NLP improves search algorithms. Results are faster and more relevant.

Due Diligence

NLP analyzes news and social media. It assesses target companies.

Portfolio Monitoring

NLP tracks market sentiment. It identifies emerging risks.

Deal Sourcing

NLP discovers potential investments. Analysis of industry trends helps.

NLP provides the tools to gain a deeper understanding of target companies and market dynamics, leading to more informed investment decisions.

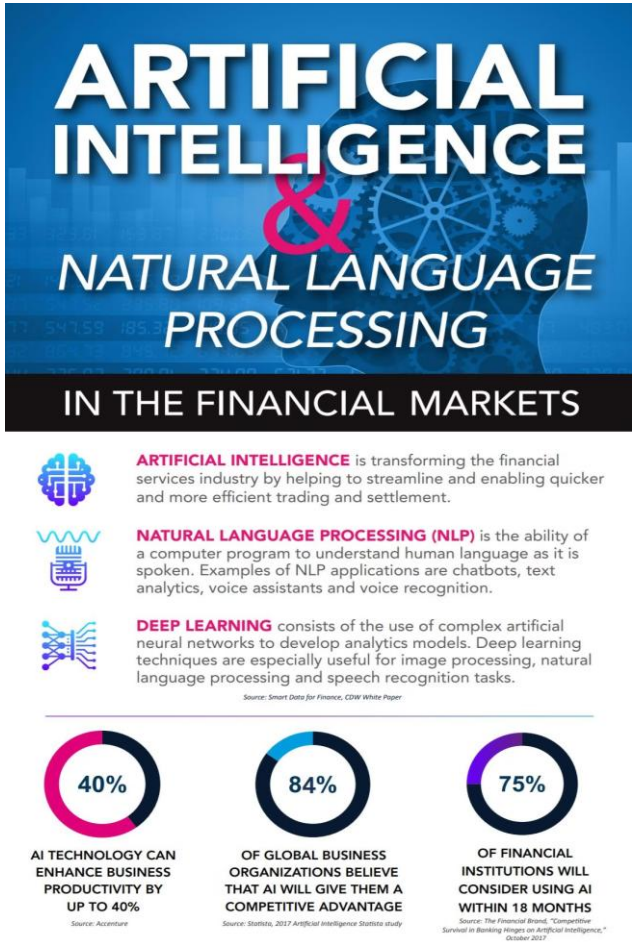
NLP Transforming Operations

Sentiment Analysis
Predict stock movements using news and social media. NLP improves trading accuracy.

Customer Service Chatbots
Automate customer support to improve response times. Reduce operational costs.

Fraud Detection
Identify fraud by analyzing emails and customer reviews. Reduce fraud losses.

Contract Analysis
Extract key data from legal contracts. Streamline due diligence and risk assessment.



Use cases so far JMAN solved

	Use case	*Approach*
1	**Apax**	**Topic Modeling**
2	**Rights Colab**	**Co-Occurrence Analysis**
3	**JMAN Internal Project**	**Sentiment Analysis**

Core NLP Tasks

Key tasks in Natural Language Processing (NLP) include text classification, sentiment analysis, information extraction, summarization, machine translation, and speech recognition.

Text Classification

Assigning predefined categories or labels to text documents.

Purpose & Usage:

- Automate content tagging
- Detect spam or phishing emails
- Categorize support tickets
- Topic identification in news or blogs

Examples:

- Spam vs. Non-spam email
- News classification: Sports, Politics, Tech
- Product reviews: Complaint, Praise, Inquiry

Sentiment Analysis

Identifying the emotional tone or attitude expressed in a text

Purpose & Usage:

- Gauge public opinion about a product, service, or event
- Analyze customer feedback at scale

Examples:

- “I love this phone!” → Positive
- “The service was terrible.” → Negative

Used in:

- Marketing analytics
- Stock market prediction

Information Extraction

Automatically extracting structured data (like names, dates, locations, relationships) from unstructured text.

This involves key subtasks as – Named Entity Recognition(NER), Relation Extraction .

Purpose & Usage:

- Fill databases from documents
- Build knowledge graphs
- Analyze legal, financial, or medical records

Text Summarization

Generating a concise and coherent version of a longer text while retaining key information.

Purpose & Usage:

- Save time reading large texts
- Digest news, reports, or research papers quickly
- Automate summarization in legal, academic, or healthcare documents

Examples:

- Summarizing a 10-page article into 3 bullet points
- AI-generated executive summaries for business reports

Continue…

Machine Translation

Automatic translation of text or speech from one language to another using computational techniques.

Purpose & Usage:

•To break language barriers and enable global communication.

•To provide real-time or offline translation in various domains like education, healthcare, business, etc.

•To improve accessibility to content for non-native speakers.

Applications :

- **Facebook / Instagram** – Auto-translates posts and comments based on user preferences.
- **YouTube** – Auto-translates video captions and descriptions.
- **E-commerce platforms** – Amazon, eBay, and AliExpress use machine translation for product descriptions and reviews.

Speech Recognition

Speech Recognition, also known as Automatic Speech Recognition (ASR) or Speech-to-Text (STT), is a branch of NLP that focuses on converting spoken language into written text using computational models.

Purpose & Usage:

- who prefer voice inputs over typing. To allow hands-free communication with devices and applications.
- To transcribe spoken words into written form for analysis, storage, or further processing.
- To increase accessibility for people with disabilities or those

Applications:

- **Voice Assistants** – Google Assistant, Siri, Alexa, Cortana.
- **Virtual Meetings** – Auto-transcription in Zoom, Google Meet, and Microsoft Teams.
- **Call Centers** – Analyze and transcribe customer conversations.

NLP Techniques

Natural Language Processing (NLP) relies on various techniques to analyze and understand human language. Below are some key NLP techniques:

Text Processing and Preprocessing	Syntax and Parsing	Semantic Analysis	Information Extraction	Text Classification	Language Generation	Speech Processing
Tokenization	PoS Tagging	Named Entity Recognition	Entity Extraction	Sentiment Analysis	Machine Translation	Speech Recognition
Stemming	Dependency Parsing	Word Sense Disambiguation	Relation Extraction	Topic Modelling	Text Summarization	Text-to-Speech (TTS) Synthesis
Lemmatization	Constituency Parsing	Coreference Resolution		Spam Detection	Text Generation	
Stopwords Removal						
Text Normalization						

Text Preprocessing

Text processing is a fundamental step in NLP that prepares raw text data for analysis, modeling, and understanding by machines. It involves cleaning, structuring, and transforming unstructured text into a form suitable for downstream NLP tasks. Transforming raw text into a usable format for NLP analysis. Since human language is messy and unstructured, raw text needs to be converted into a clean, consistent, and structured format before any NLP model or algorithm can work with it.

Text Preprocessing Importance –

- Improves accuracy and efficiency of NLP models
- Helps to reduce noise in the data
- Standardizes diverse language forms (e.g., slangs, misspellings)
- Essential for extracting meaningful patterns and features
- Converts free – flowing natural language into machine- readable format

Text Processing Techniques

1

Tokenization

Breaking Text into Words

- **Word – Based:** Split on spaces and punctuation.
- **Subword:** Split words into smaller units.
- **Character-Based:** Split text into individual characters.
- **Example**: "The quick brown fox." becomes ["The", "quick", "brown", "fox", "."].
- **Tools**: NLTK's `word_tokenize` and spaCy's tokenizers.

2

Lowercasing and Removing Punctuation

Removes punctuation, special characters, and numbers that clutter the analysis

- **Lowercasing:** convert all text to lowercase. Reduces vocabulary size.
- **Punctuation Removal:** Remove punctuation marks. Reduces noise.

3

Stop word removal

Stop words are common words in a language that are often removed from text data during preprocessing because they do not add significant meaning to the analysis.

- Reduce Noise
- Improve efficiency
- Focus on meaningful words

Continue…

4

Stemming

Stemming is the process of reducing a word to its root form by chopping off prefixes or suffixes. It uses heuristic rules rather than understanding the context or grammatical correctness.

- **Advantages:**
  - Simple and fast.
  - Reduces dimensionality of text data.
- **Disadvantages:**
  - May result in non-meaningful stems (e.g., "studies" → "studi").
  - Ignores the meaning or context of words.

5

Lemmatization

Lemmatization is the process of reducing a word to its base or dictionary form (lemma) while ensuring the result is a valid word. It considers the word's context and part of speech (POS).

- **Advantages**:
  - Produces meaningful and grammatically correct words.
  - Better for context-sensitive NLP tasks.
- **Disadvantages**:
  - Slower than stemming.
  - Requires more computational resources.

Text Representation in NLP

The process of converting raw text into a numerical format that machine learning models can understand and work.

Text Representation Importance –

- Transforms unstructured text into a structured numerical form
- Enables algorithms to find patterns, similarities, and relationships
- Affects the accuracy, performance, and scalability of NLP tasks like sentiment analysis, classification, etc.
- Enables efficient processing and computation
- Allows algorithms to measure similarity, identify patterns, or make predictions
- Helps in capturing the semantic, syntactic, and contextual meanings of words or documents

Bag of Words(BOW)

What is Bag of Words?

- BoW represents text as a vector of word counts.
- It ignores grammar and word order, focusing only on word frequency.
- Each document is transformed into a vector where each dimension represents a unique word in the dataset.
- Example Consider two sentences:
  - "Private equity firms invest in startups."
  - "Startups receive investment from private firms.“
- Limitations:
  - Does not consider meaning (e.g., "invest" and "investment" are treated differently).
  - High-dimensional for large vocabularies.
- Applications:
  - Document Classification
  - Information Retrieval / Search Engines
  - Plagiarism Detection

Word	Sentence 1 Count	Sentence 2 Count
private	1	1
equity	1	0
firms	1	1
invest	1	0
startups	1	1
receive	0	1
investment	0	1
from	0	1

TF-IDF (Term Frequency-Inverse Document Frequency)

What is TF-IDF?

- BoW gives equal importance to all words.
- TF-IDF improves this by assigning weights:
  - Term Frequency (TF): How often a word appears in a document.
  - Inverse Document Frequency (IDF): How rare a word is across all documents.
- Advantages
  - Gives more importance to meaningful words.
  - Reduces the impact of common words.
- Limitations
  - Still does not capture word meaning or context.
  - Cannot handle synonyms or polysemy (same word with different meanings).
- Applications
  - Search Engines use TF-IDF to rank the relevance of a document for a query

Word	Sentence 1	Sentence 2
private	0.379	0.379
equity	0.533	0
firms	0.379	0.379
invest	0.533	0
startups	0.379	0.379
receive	0	0.533
investment	0	0.533

Word Embeddings

What are word embeddings?

- Unlike BoW and TF-IDF, which treat words as independent, word embeddings capture relationships between words.
- Word embeddings represent words as dense vectors in a multi-dimensional space.
- Words with similar meanings have vectors that are close together.
- How Does This Help?
  - Captures synonyms and relationships (e.g., "car" and "automobile" are close).
  - Can understand analogies (e.g., "King – Man + Woman = Queen").
  - Works well with deep learning models for NLP.
- Popular Word Embedding Models
  - Word2Vec (CBOW & Skip-Gram)
  - GloVe (Global Vectors for Word Representation)
  - FastText
  - Transformer-based embeddings (BERT, GPT)

Traditional Approach (BoW/TF-IDF) → Sparse & Independent

Word	Vector Representation (BoW)
king	[0, 0, 1, 0, 0, 1, 0, 0, 1]
queen	[1, 0, 0, 1, 0, 0, 1, 0, 0]
prince	[0, 1, 0, 0, 1, 0, 0, 1, 0]

(No relationship between words is captured.)

Word Embeddings → Dense & Meaningful

Word	Vector Representation (Word2Vec)
king	[0.9, 0.1, 0.7, 0.3]
queen	[0.88, 0.15, 0.72, 0.29]
prince	[0.85, 0.12, 0.75, 0.27]

"King" and "Queen" have similar numerical representations!

NLP Feature Extraction

Feature extraction in Natural Language Processing (NLP) is the process of transforming raw text into a format that can be understood and processed by machine learning models.

Since machines don't inherently understand human language, feature extraction helps convert text into **meaningful numerical representations** that capture key characteristics like word frequency, structure, or context.

Feature Extraction Importance –

- Transforms unstructured data (text) into structured features
- Highlights relevant patterns in the text for the model to learn from
- Improves model performance in tasks like classification, clustering, or regression
- Reduces dimensionality while preserving important linguistic information

N- grams

Capturing sequences of N words to preserve some context and word order
N-grams are continuous sequences of n items (words or characters) from a given text.
Captures **word order** and **context** better than individual words

Applications :

Text classification

spam filtering

sentiment analysis

Types of N-grams

- Unigrams: Single words (e.g., "the", "quick", "brown")
- Bigrams: Two-word sequences (e.g., "the quick", "quick brown")
- Trigrams: Three-word sequences (e.g., "the quick brown", "quick brown fox")
- Higher-order N-grams: Four-word sequences and beyond (less common due to data sparsity)

Examples of N-gram Usage

- Language Modeling: Predicting the next word in a sequence using N-gram probabilities.
  - Example: Given "the quick brown," predict "fox" with a probability based on the frequency of "the quick brown fox" in the training data.
- Spelling Correction: Identifying and correcting misspelled words by comparing N-gram frequencies with known correct sequences.
  - Example: Correcting "teh" to "the" based on the higher frequency of "the" in the corpus.

Sentiment Analysis

Sentiment analysis automates the process of identifying opinions in text. The goal is to determine the attitude or emotions of a speaker or writer. It helps understand customer perceptions and improve decision-making. Sentiment analysis enables real-time monitoring of brand reputation and customer feedback.

1. **Financial Sentiment Analysis (stock market prediction)**
2. **Customer Reviews**
3. **Social Media Monitoring**
4. **Employee Feedback Analysis**

Sentiment Analysis Tools and Techniques

Rule Based System

These rely on predefined rules and sentiment lexicons, such as assigning positive or negative scores to words (e.g., "good" = +1, "bad" = -1). They are simple and interpretable but struggle with sarcasm, negation, and complex context. Often used in **basic sentiment analysis tasks** or as a **baseline approach**.

NLTK and VADER

NLTK is a powerful open-source Python library for NLP, providing tools for **text processing, tokenization, stemming, and sentiment analysis**.

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a specialized lexicon-based tool optimized for short, informal text like tweets and product reviews. VADER works well with **emoji-based and slang-heavy social media text**.

Machine Learning

Supervised learning models like **Naive Bayes, SVM, or Logistic Regression** are trained on labeled text data to classify sentiment. Features like **TF-IDF or word embeddings** help improve accuracy. ML-based approaches perform better than rule-based ones but require **large, high-quality datasets** for training.

Cloud APIs

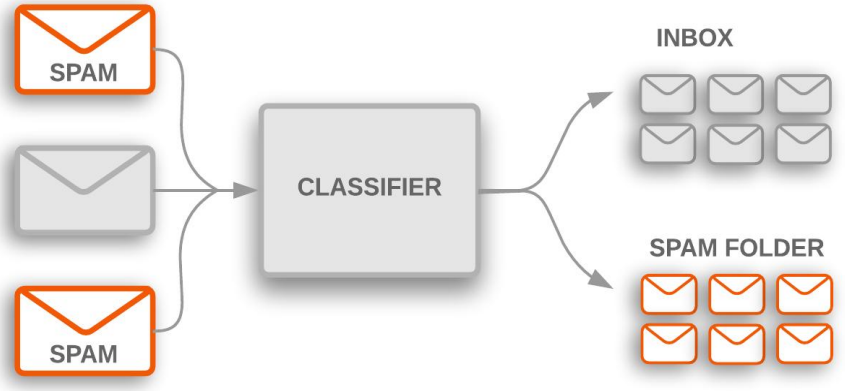
Services like **Google Cloud Natural Language, Amazon Comprehend, and Microsoft Azure Text Analytics** use deep learning models to analyse sentiment in real-time. They support **multilingual sentiment analysis, contextual understanding, and large-scale processing**. These APIs are easy to integrate but may have **higher costs and limited customization**.

NLP Text Classification

Categorizing text into predefined classes (e.g., spam detection, sentiment analysis).

Naïve Bayes, Logistic Regression, SVM, CNNs, RNNs, Transformers algorithms are used with NLTK / spaCy, scikit-learn , HuggingFace Transformers libraries

To evaluate metrics like
Accuracy, F1-score, Confusion Matrix, **ROC-AUC Score** (for binary classification)



Importance -
To automatically organize and sort text data
Enables machines to understand the context or intent behind written words
Helps in automating decision-making processes in systems that involve large volumes of text

Applications -
Email spam filters, Customer support ticket routing, Social media sentiment tracking, Fake news detection, Product and movie review analysis

Pipeline Overview –
Text → Preprocessing → Vectorization → Model Training → Evaluation → Prediction

Information Extraction

Information Extraction (IE) is a subfield of NLP that extracts structured data from unstructured text like documents, reports, and emails. It identifies key details such as names, dates, and relationships to make text more meaningful. IE is widely used in finance, healthcare, and legal domains to automate data processing and improve decision-making.

1. **Healthcare Applications**
2. **Finance Sector Utilization**
3. **Marketing Insights**
4. **Investment Research**

Co-Occurrence

Co-occurrence refers to words appearing together in each context, helping understand relationships between words.

Types of Co-occurrence

- Local (Window-based) – Words near each other (e.g., "New York")
- Document-level – Words appearing in the same document
- Collocations – Frequent word pairs (e.g., "strong tea")

Example

- "bank" co-occurs with "river" → related to water
- "bank" co-occurs with "money" → related to finance

Real Time Applications

- **Topic Modeling**
  - ➢ Finds word distributions across documents using co-occurrence patterns
- **Name Entity Recognition**
  - ➢ Identifies entities (names, locations, organizations) based on contextual words
- **Information Retrieval & Search Engines**
  - ➢ Helps rank results by analysing keyword co-occurrence patterns in documents
- **Sentiment Analysis**
  - ➢ Determines sentiment by analysing co-occurrence of words like "happy" & "excited" (positive) vs. "sad" & "angry" (negative)

Name Entity Recognition

Identifies and classifies named entities into categories like Person, Organization, Location, Date, Time, etc.

Why is NER Important?	How Does NER Work?	NER Approaches	Applications
Extracts Structured Data *– Converts unstructured text into meaningful categories (Person, Organization, Location, etc.).*	***Tokenization*** *– Break text into words*	***Rule-based Systems*** *– Uses predefined dictionaries and pattern-matching techniques to identify entities.*	***Search Engines*** *– Improves Google results*
Enhances Search Engines *– Improves search accuracy by recognizing key entities (e.g., "Apple" as a company vs. a fruit).*	***POS Tagging*** *– Identify word roles*	***Statistical Machine Learning*** *(ML) – Uses Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) to classify entities based on features.*	***Chatbots*** *– Extracts user intent (e.g., "Book a flight to Paris")*
Powers Chatbots & Virtual Assistants *– Helps chatbots understand and respond to user queries (e.g., "Book a flight to Paris").*	***Entity Recognition*** *– Detect named entities*	***Deep Learning-Based NER*** *– Leverages LSTMs, BiLSTMs, and Transformers (BERT) for higher accuracy and contextual understanding.*	***Finance*** *– Identifies company names & stock trends*
Supports Finance & Healthcare *– Identifies company names, stock trends, diseases, and drug names for analysis.*	***Classification*** *– Asisgn categorise (Person, Location, etc.)*	***Popular NER Tools*** *– Frameworks like SpaCy, Stanford NER, and Hugging Face Transformers offer pre-trained models for entity recognition.*	***Healthcare*** *– Extracts drug names, diseases from medical records*

NLP Relationship Extraction

Relationship Extraction involves identifying and classifying semantic relationships between entities in a piece of text.

Example: In the sentence "Apple was founded by Steve Jobs and Steve Wozniak," the relationship extraction model identifies "Apple" as the *organization*, "Steve Jobs" and "Steve Wozniak" as *people*, and "founded by" as the *relationship* between them.

Key Applications & Use Cases:

- **Knowledge Graph Construction:** Populating knowledge bases with structured data extracted from text. Example: Building a medical knowledge graph by extracting relationships between diseases, symptoms, and treatments from medical literature.
- **Customer Service Chatbots:** Understanding customer needs and connecting them with relevant solutions. Example: Identifying the relationship between a customer's problem ("broken screen") and a product ("iPhone 13").
- **Financial Analysis:** Discovering connections between companies, executives, and financial events. Example: Identifying relationships between mergers, acquisitions, and stock price changes from news articles.
- **Scientific Research:** Automating the extraction of relationships between genes, proteins, and diseases from scientific publications. Example: Identifying gene-disease associations to accelerate drug discovery.
- **Cybersecurity Threat Intelligence:** Correlating threat actors, malware, and vulnerabilities from security reports and blogs. Example: Linking specific malware strains to known attacker groups.

Benefits of relationship extraction: Automation and efficiency improvements, better insights from unstructured data, decision making augmentation.

Thank You!...