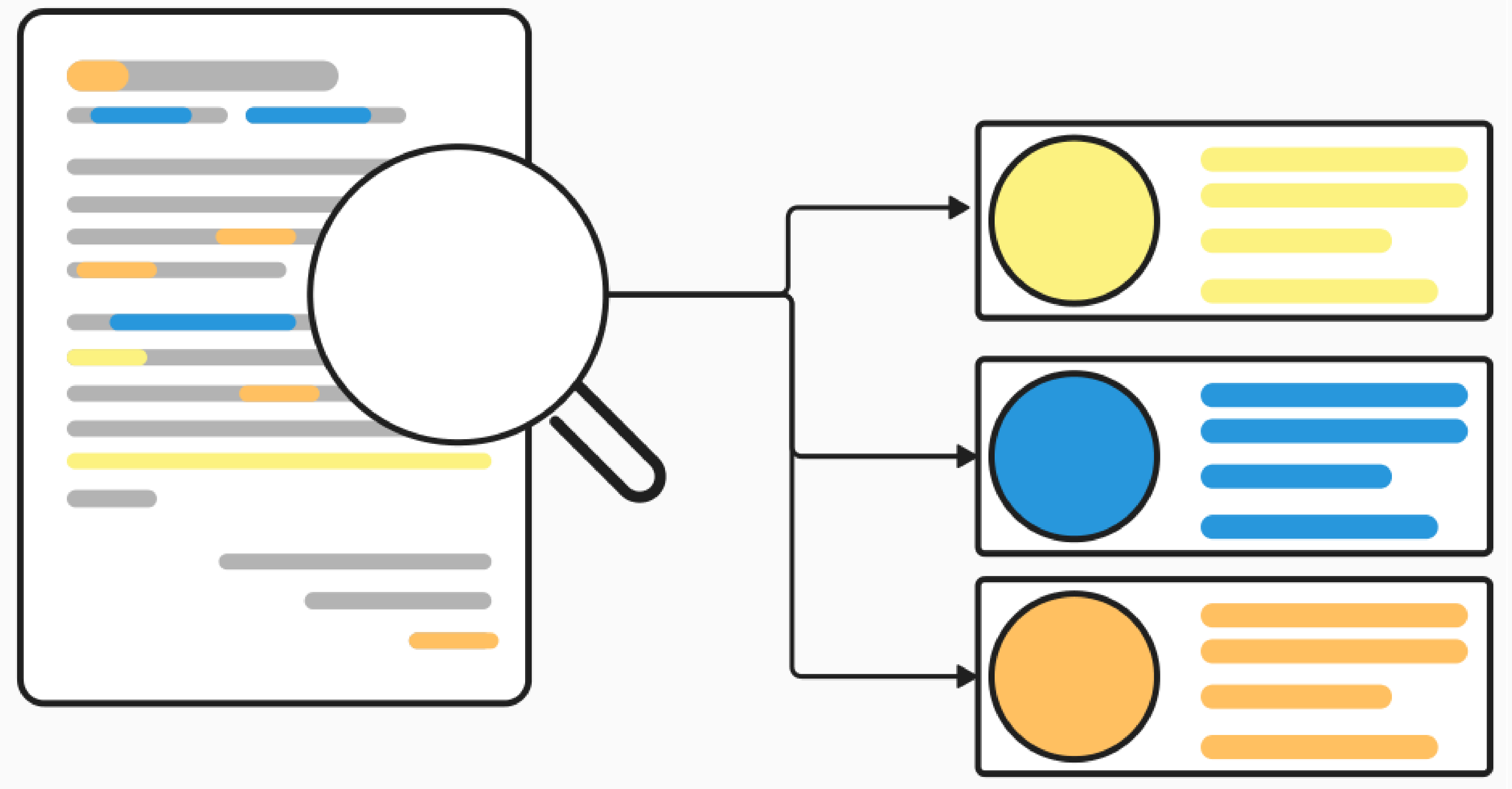
Text Classification and Natural Language Processing



Course overview

Material → https://github.com/lu-ser/NLPSeminar

A comprehensive seminar covering modern text classification techniques, from fundamentals to advanced LLM applications. The course combines theoretical foundations with hands-on practice, emphasizing real-world applications.

Learning objectives

- Understand fundamental concepts of text classification
- Master modern NLP techniques and architectures
- Gain practical experience with LLMs
- Develop hands-on skills in RAG implementation
- Explore current trends and future directions

Agenda

03 Feb 2025	Fundamentals of Text Classification	10:00 - 12:00
04 Feb 2025	Modern Text Classification	10:00 - 12:00
05 Feb 2025	Large Language Models	10:00 - 12:00
06 Feb 2025	Introduction to RAG	10:00 - 12:00
07 Feb 2025	Future Trends and Project Work	10:00 - 12:00

Lesson Overview

Key Topics Covered:

1.Introduction to NLP

a) Definition and importance of Natural Language Processing.
b) Applications: Sentiment Analysis, Machine Translation, Speech Recognition.
c) Societal, economic, and educational impacts.

2.Text Classification

a) Definition and practical use cases (e.g., spam detection, fake news classification).
b) Real-world applications: Brand monitoring, content categorization, user behavior analysis.

3.NLP Pipeline

a) Data Acquisition: Sources, quality considerations, and privacy.
b) Text Preprocessing: Cleaning, normalization (lemmatization, stemming).
c) Feature Engineering: BoW, TF-IDF, embeddings, and challenges.
d) Modeling & Evaluation: Algorithms (e.g., SVM, BERT), metrics (accuracy, precision, recall).

4.Hands-on Workshop

a) Build a sentiment classifier for movie reviews.
b) Tools: Hugging Face datasets, TF-IDF, Multinomial Naive Bayes.

Definition according to IBM

Natural language processing (NLP) is a subfield of computer science and artificial intelligence that uses machine learning to enable computers to understand and communicate with human language.

Why it matters:

Fundamental to Human-Computer Interaction:

- Enables natural communication between humans and machines
- Breaks down language barriers in technology access
- Makes technology more inclusive and accessible to non-technical users

Impact on Everyday Life:

- **Voice Assistants**: Siri, Alexa, Google Assistant rely on NLP for speech recognition and response.
- **Social Media Insights**: Analyzing trends, sentiment, and user behavior in real-time.
- **Translation**: Real-time language translation facilitates global communication.

Social Impact:

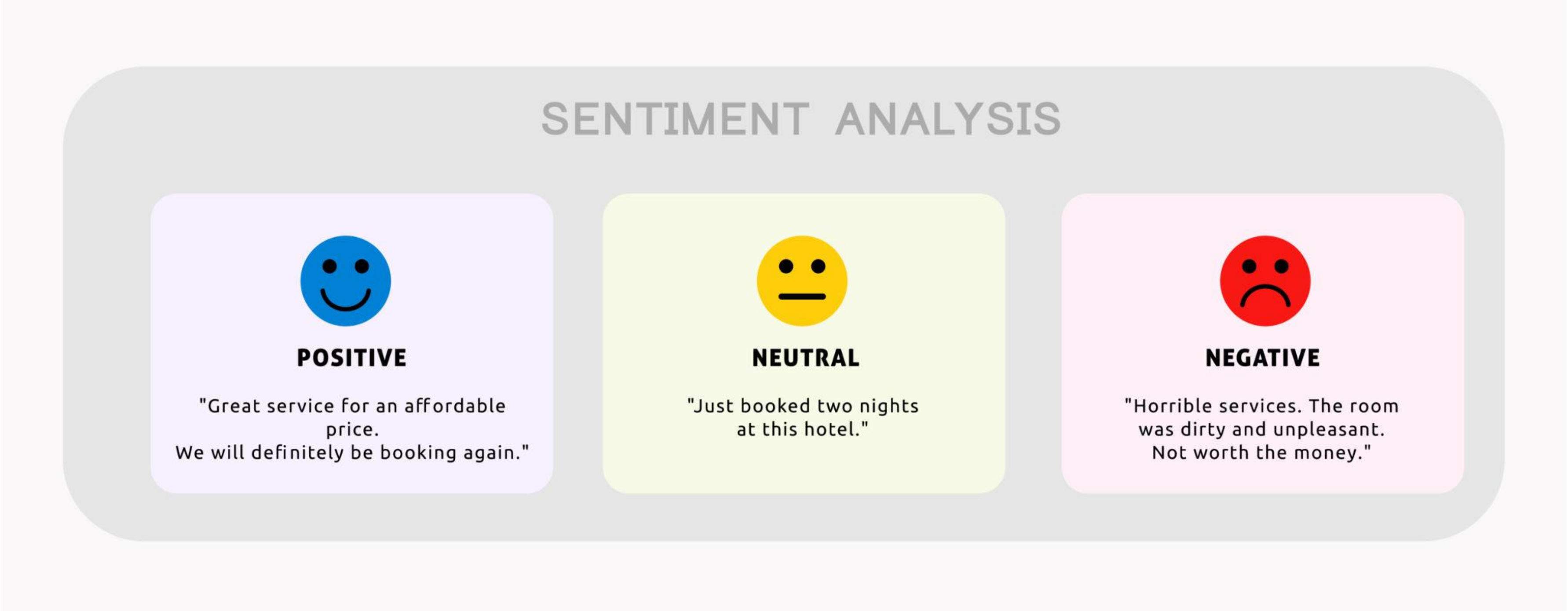
- Democratizes access to information across languages
- Improves healthcare through better documentation and analysis
- Assists people with disabilities through text-to-speech and speech-to-text

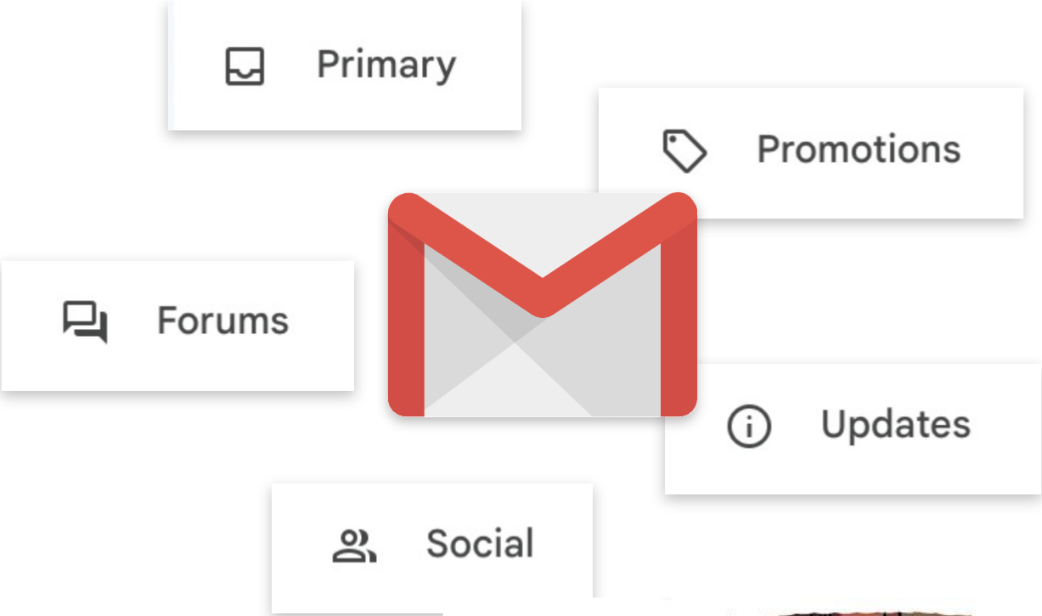
Economic and Education Impact

- **Business Applications:** Automates customer support, improves search engines, and optimizes workflows.
- **Education**: Enables personalized learning experiences through adaptive learning platforms.

Applications:

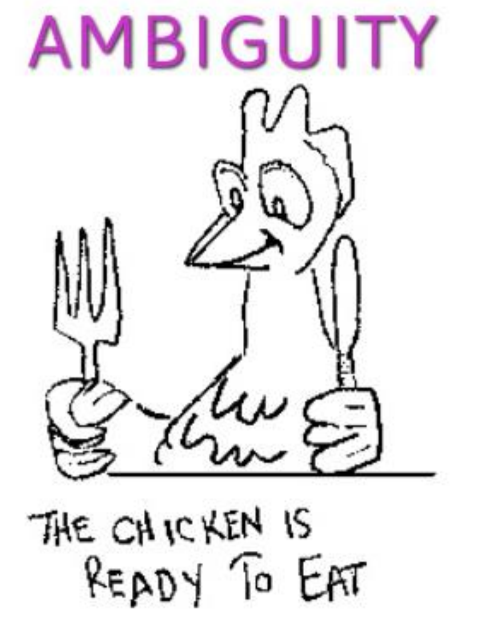
- Sentiment analysis
- Machine translation
- Information extraction and summarisation, categorisation
- Speech recognition and text-to-speech
- Question-answering systems
- Language Generation

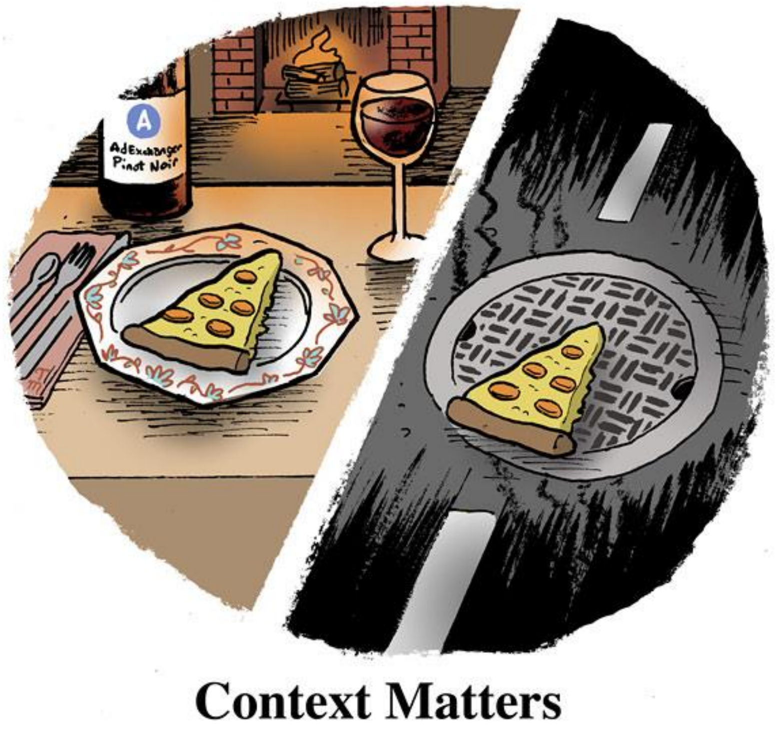




Key Challenges:

- Ambiguity in language
- Context understanding
- Handling different languages and dialects
- Processing informal language and slang
- Maintaining semantic meaning





1.2 Key Challenges

Riddle 1: how many eyes does Mississippi have?

Okay, let's try to figure out how many eyes Mississippi has. Hmm, the question is a bit tricky because it's a play on words, right? Mississippi is a state in the US, but obviously, a state doesn't have literal eyes. So maybe the question is asking about something else.

Wait, "eyes" could be referring to the letter 'i' in the word "Mississippi." Sometimes in riddles, they use homophones or letters. Let me check the spelling of Mississippi. Let's break it down: M-I-S-S-I-S-S-I-P-P-I. Let's count the number of 'i's.

The question is a playful riddle. Mississippi, when spelled out, contains the letter **'i'** four times. If each **'i'** is considered an "eye," then Mississippi has **4 eyes**.

Answer: 4.

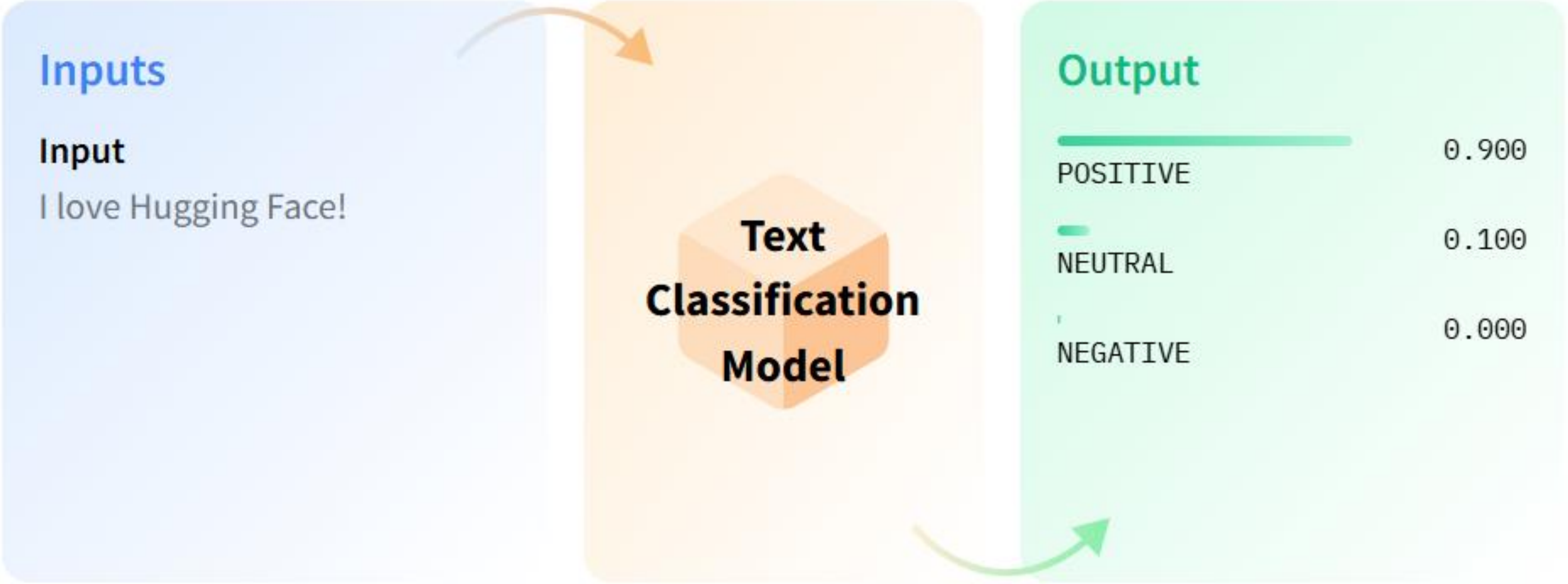
1.2 Key Challenges

L I remember a movie title that was like: The Quick and the Angry. Is there a movie with a similar title?

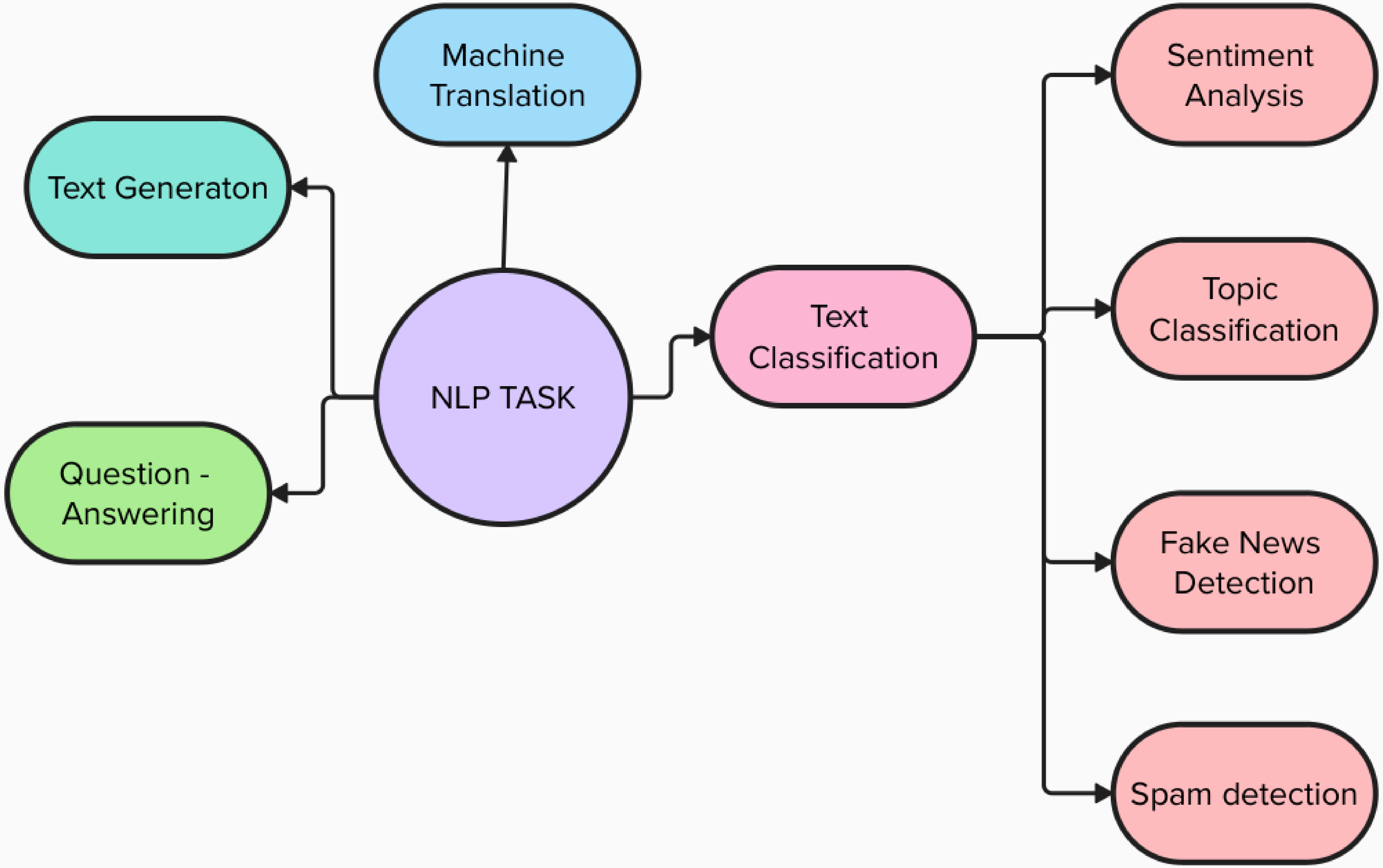
You might be thinking of "The Fast and the Furious," which is a popular action film franchise that began in 2001. The title structure (The [Adjective] and the [Adjective]) is similar to what you remembered. The franchise has become one of the most successful movie series, focusing on street racing, heists, and action sequences.

According to Hugging Face

“Text classification is a common NLP task that assigns a label or class to text. Some of the largest companies run text classification in production for a wide range of practical applications. One of the most popular forms of text classification is sentiment analysis, which assigns a label like 🙂 positive, 🙁 negative, or 😐 neutral to a sequence of text.”



3. Text Classification Tasks



4. Real-world text classification applications

Content & News Classification

- Automated categorization of news articles and digital content into relevant topics
- Organization and routing of internal documents within companies based on their content
- Fake News Detection

Sentiment Analysis, Brand Monitoring & Hate Speech Detection

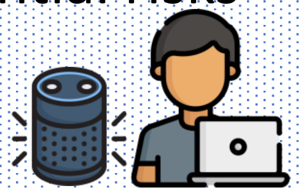
- Real-time analysis of customer reviews and social media posts to assess product sentiment
- Tracking brand reputation through continuous monitoring of online comments and feedback

Email & Spam Detection

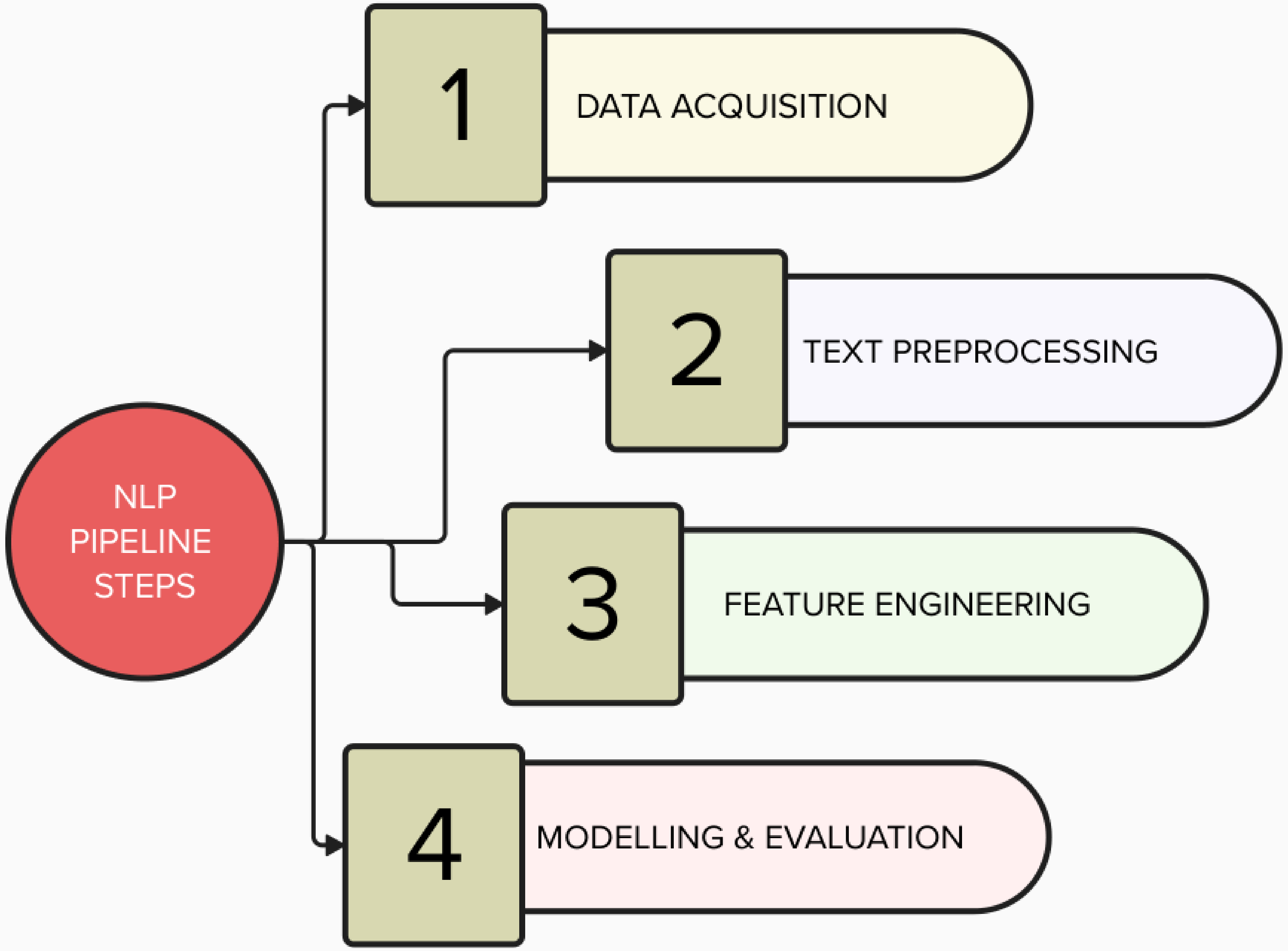
- Automatic identification and filtering of unwanted or potentially harmful emails
- Classification of incoming messages by priority and department routing

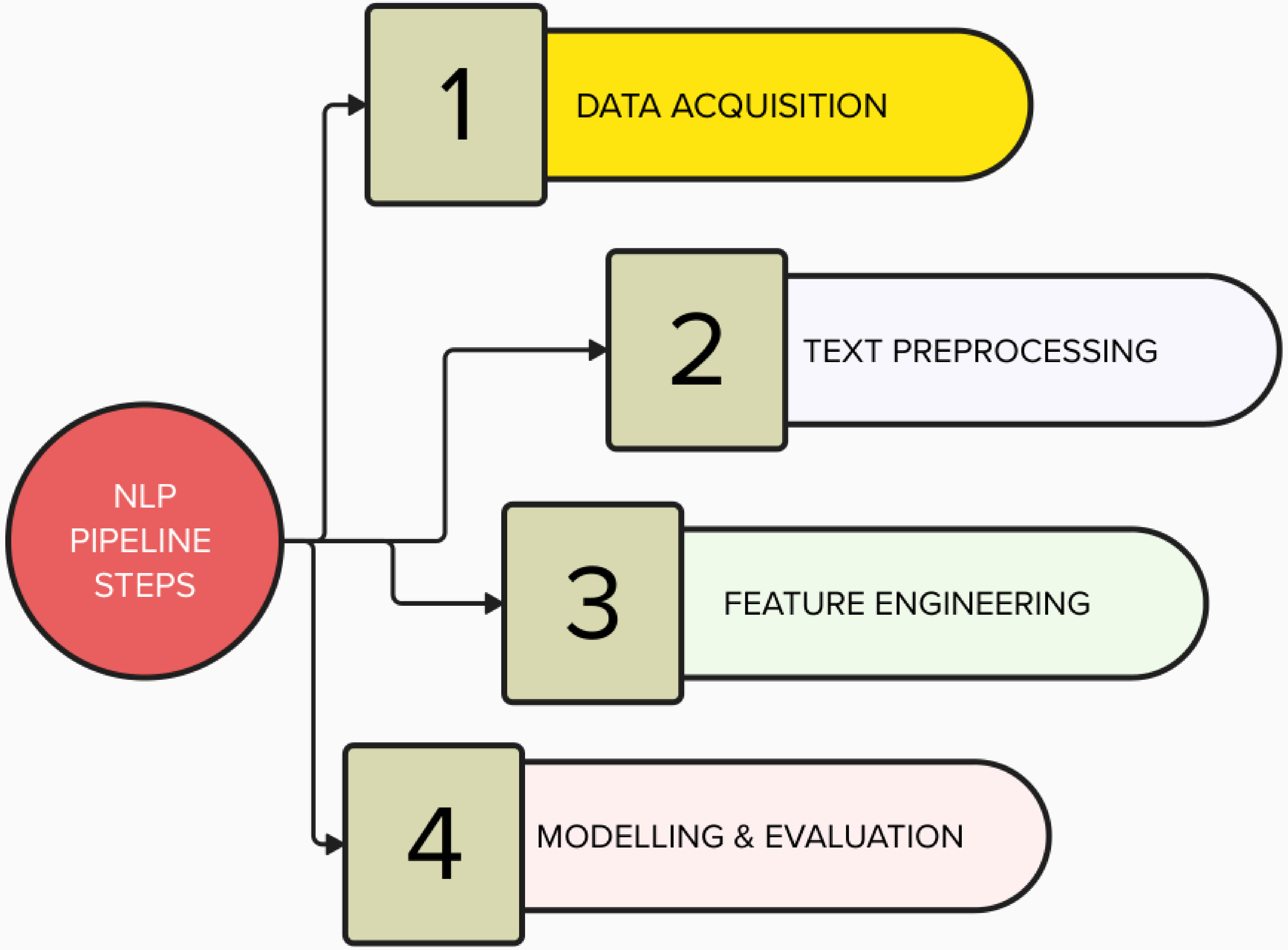
User Behavior Analysis

- Pattern recognition in user interactions and communications to predict future actions
- Analysis of user language and intent to identify potential risks or opportunities



5. NLP Pipeline



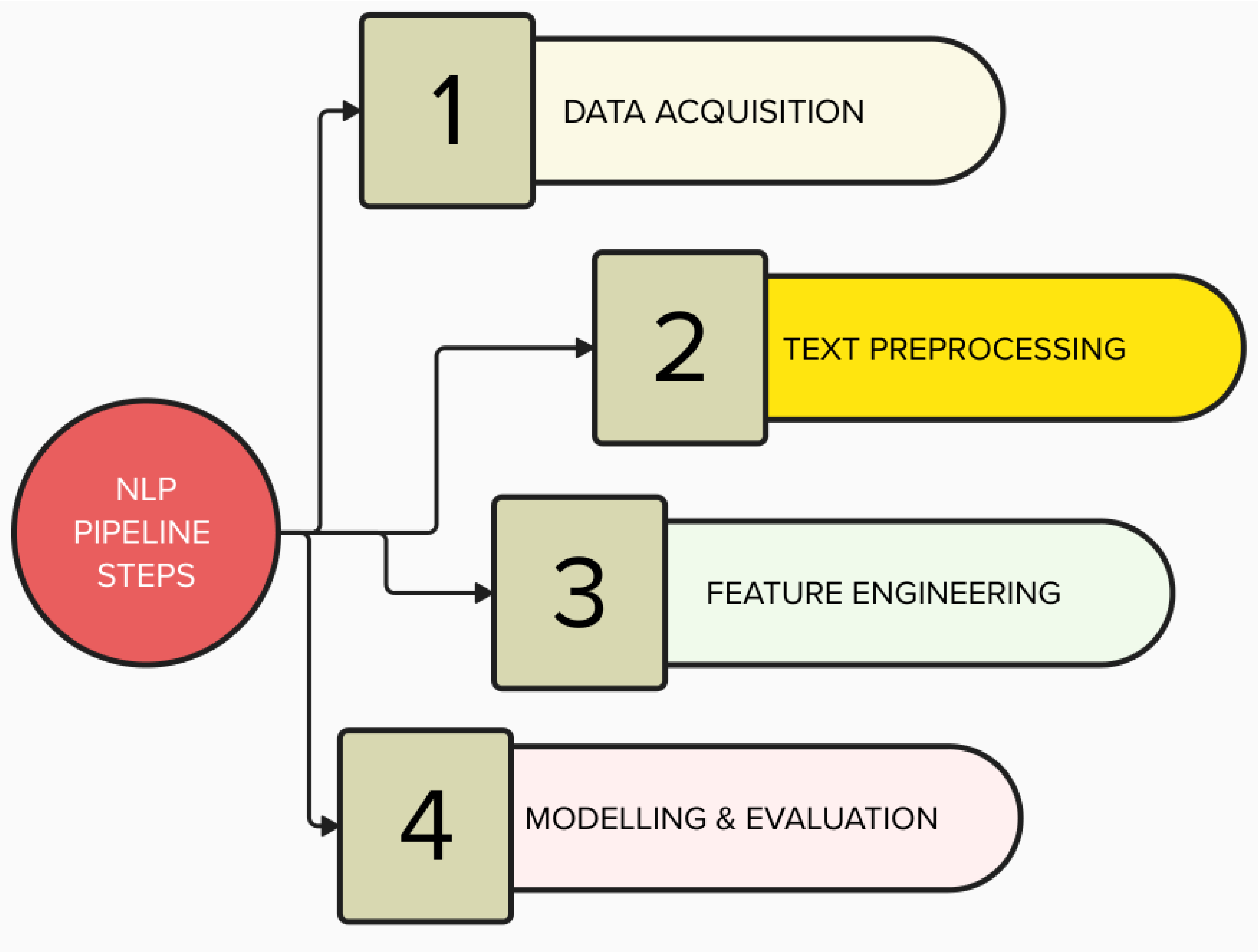


Data Acquisition:

- Raw Data Collection Methods
  - Web Scraping of online content
  - Datasets or Databases from repositories
  - APIs and Third-party data providers
  - Ad hoc solutions, e.g., mobile applications

- Data Quality Considerations
  - Handling missing or incomplete data
  - Managing data privacy and compliance
  - Establishing data storage infrastructure

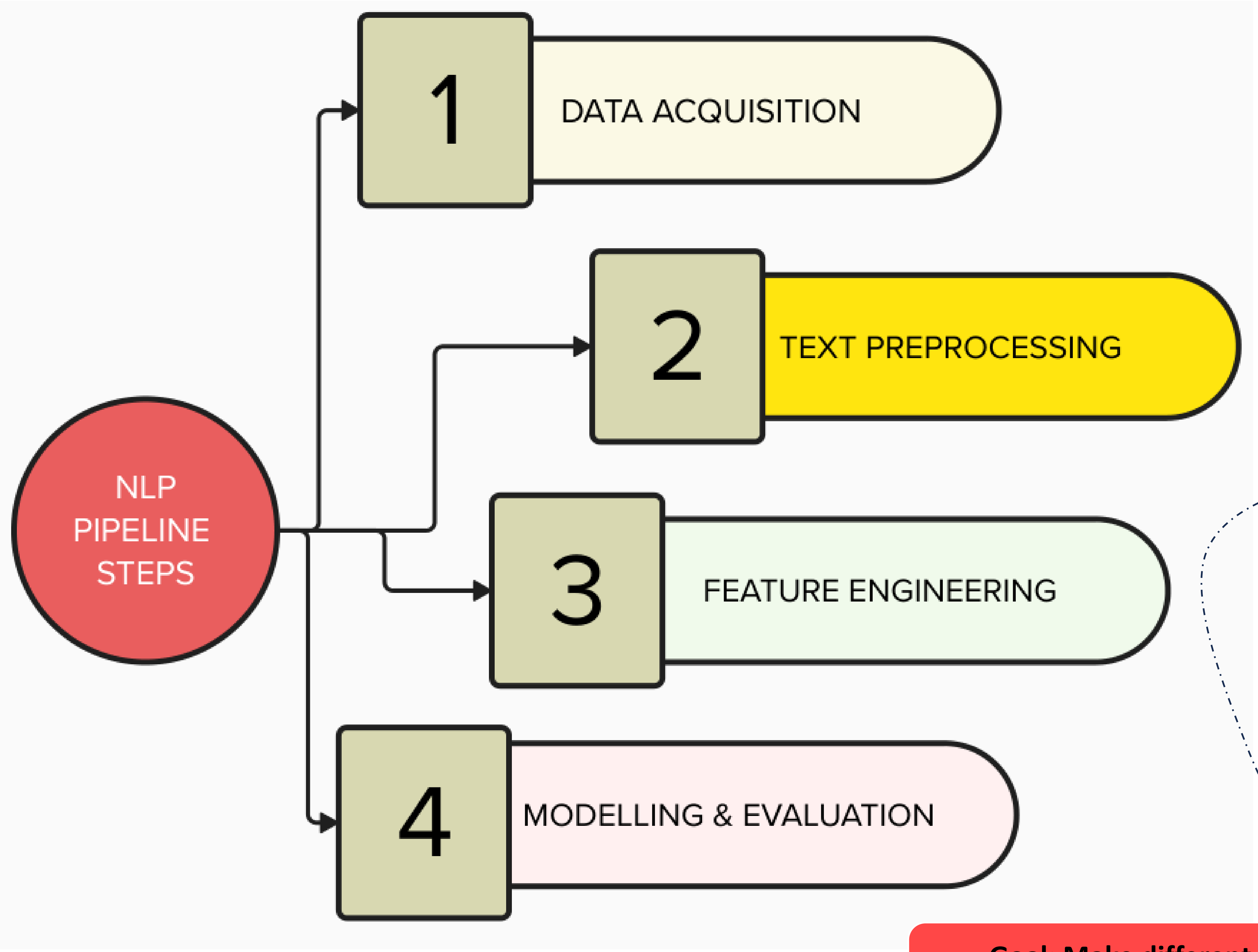
5.1. Data Acquisition – Microsoft Tay twitter bot



Text Preprocessing

- Text Cleaning
  - Removal of HTML tags and special characters
  - Deleting extra whitespaces
  - Handling noise and irrelevant characters
- Text Normalization
  - Converting text to lowercase/uppercase
  - Lemmatization (converting words to their base form: running → run)
  - Stemming (cutting words to their root: fishing → fish)
  - Standardizing spellings (color/colour → color)

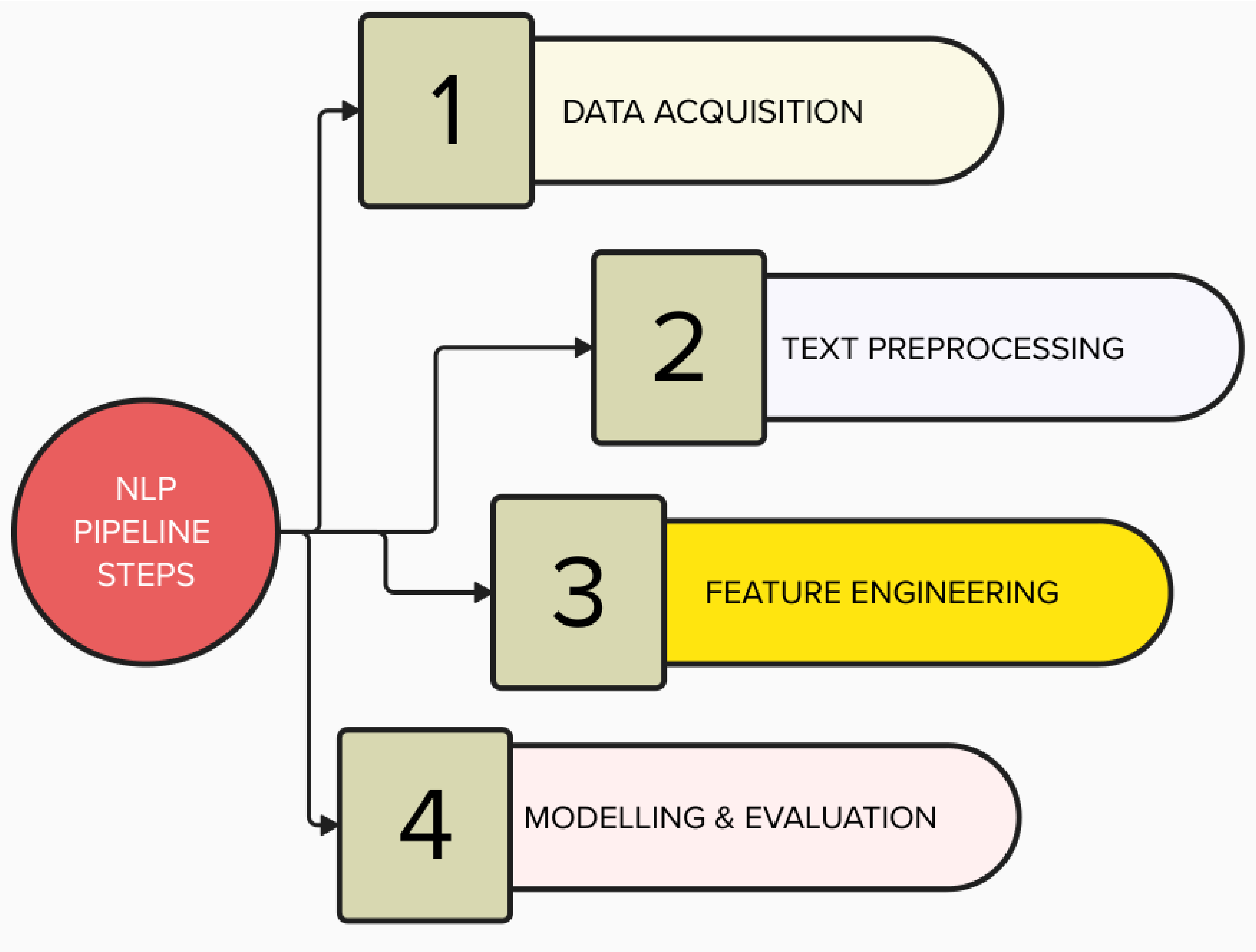
Goal: "Clean" the text from elements that don't contribute to meaning



Text Preprocessing

- Text Cleaning
  - Removal of HTML tags and special characters
  - Deleting extra whitespaces
  - Handling noise and irrelevant characters
- Text Normalisation
  - Converting text to lowercase/uppercase
  - Lemmatization (converting words to their base form: running → run)
  - Stemming (cutting words to their root: fishing → fish)
  - Standardizing spellings (color/colour → color)

Goal: Make different forms of the same word/concept identical for processing



Feature Engineering

- Text Representation Methods
  - Bag of Words (BoW) and N-grams implementation
  - TF-IDF
  - Embeddings
- Feature Optimization
  - Dimensionality reduction (PCA, t-SNE)
  - Handling of sparse matrices

TRY GUESSING!

Document D1	*The child makes the dog happy*

↓

	child	dog	happy	makes	the	BoW Vector representations
D1						

BoW

- <u>It captures how often each word appears in a document</u>
- Text represented as a collection of word occurrences
- Loses word order but maintains frequency information
- Simple and computationally efficient

5.3.1 Feature Engineering - BOW

Document D1	*The child makes the dog happy* the: 2, dog: 1, makes: 1, child: 1, happy: 1

↓

	child	dog	happy	makes	the	BoW Vector representations
D1	1	1	1	1	2	**[1,1,1,1,2]**

BoW

- <u>It captures how often each word appears in a document</u>
- Text represented as a collection of word occurrences
- Loses word order but maintains frequency information
- Simple and computationally efficient

5.3.1 Feature Engineering - BOW

Document D1	*The child makes the dog happy* the: 2, dog: 1, makes: 1, child: 1, happy: 1
Document D2	*The dog makes the child happy*

	child	dog	happy	makes	the	BoW Vector representations
D1	1	1	1	1	2	**[1,1,1,1,2]**

From AIML.com

BoW

- <u>It captures how often each word appears in a document</u>
- Text represented as a collection of word occurrences
- Loses word order but maintains frequency information
- Simple and computationally efficient

5.3.1 Feature Engineering - BOW

Document D1	*The child makes the dog happy* the: 2, dog: 1, makes: 1, child: 1, happy: 1
Document D2	*The dog makes the child happy* the: 2, child: 1, makes: 1, dog: 1, happy: 1

↓

	child	dog	happy	makes	the	BoW Vector representations
D1	1	1	1	1	2	**[1,1,1,1,2]**
D2	1	1	1	1	2	**[1,1,1,1,2]**

From AIML.com

BoW

- It captures how often each word appears in a document
- Text represented as a collection of word occurrences
- Loses word order but maintains frequency information
- Simple and computationally efficient

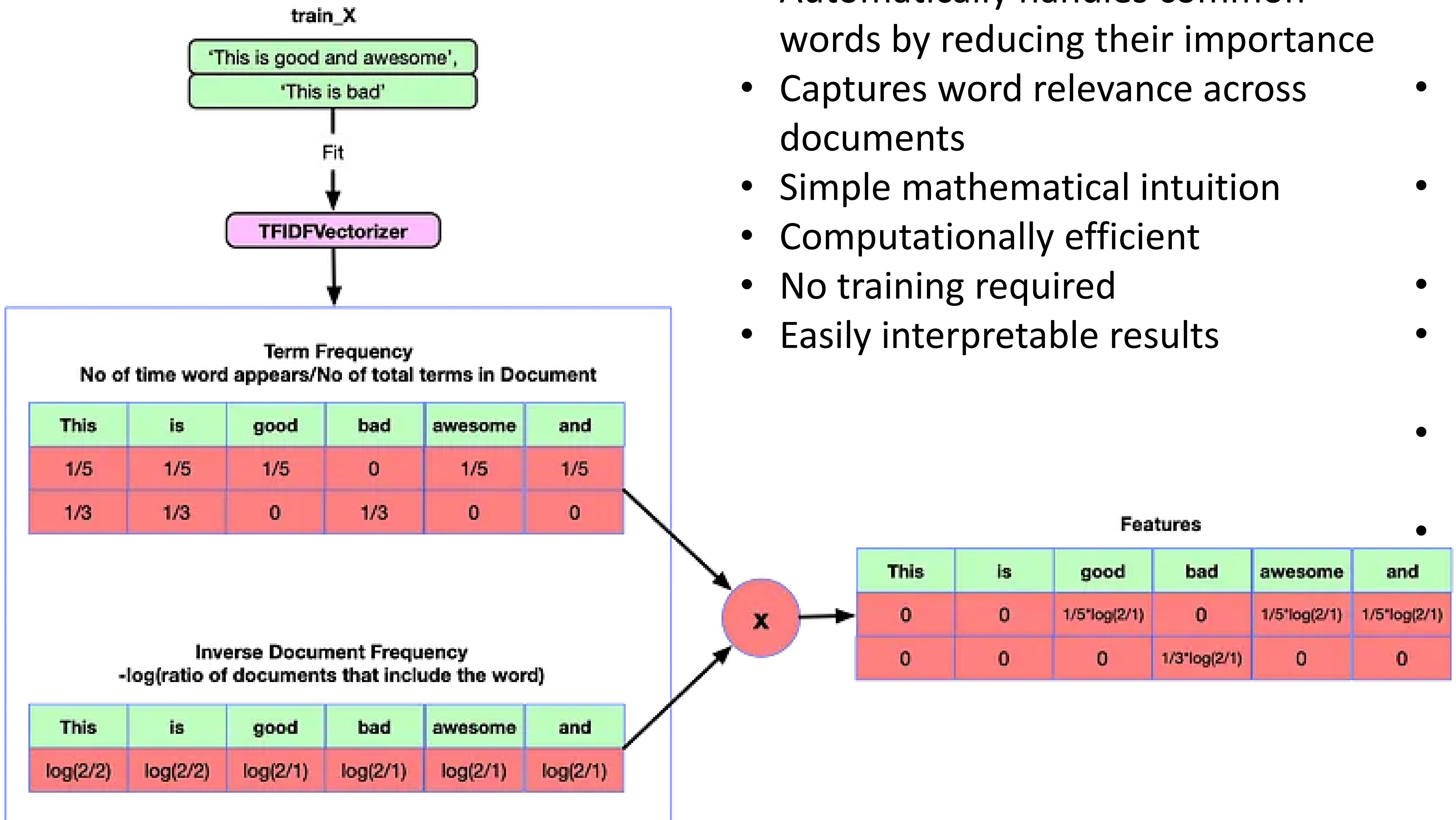
Term Frequency-Inverse Document Frequency

Pros:

- Automatically handles common words by reducing their importance
- Captures word relevance across documents
- Simple mathematical intuition
- Computationally efficient
- No training required
- Easily interpretable results

Cons:

- Still a bag-of-words approach: loses word order
- Cannot capture semantic relationships between words
- "Dog" and "puppy" are treated as completely different terms
- No understanding of synonyms
- High dimensionality with large vocabularies
- Sparse representation: most values are zero
- Cannot handle out-of-vocabulary words



From Medium.com

How to represent words with vector?

One-hot Encoding

dog → 1 0 0

cat → 0 1 0

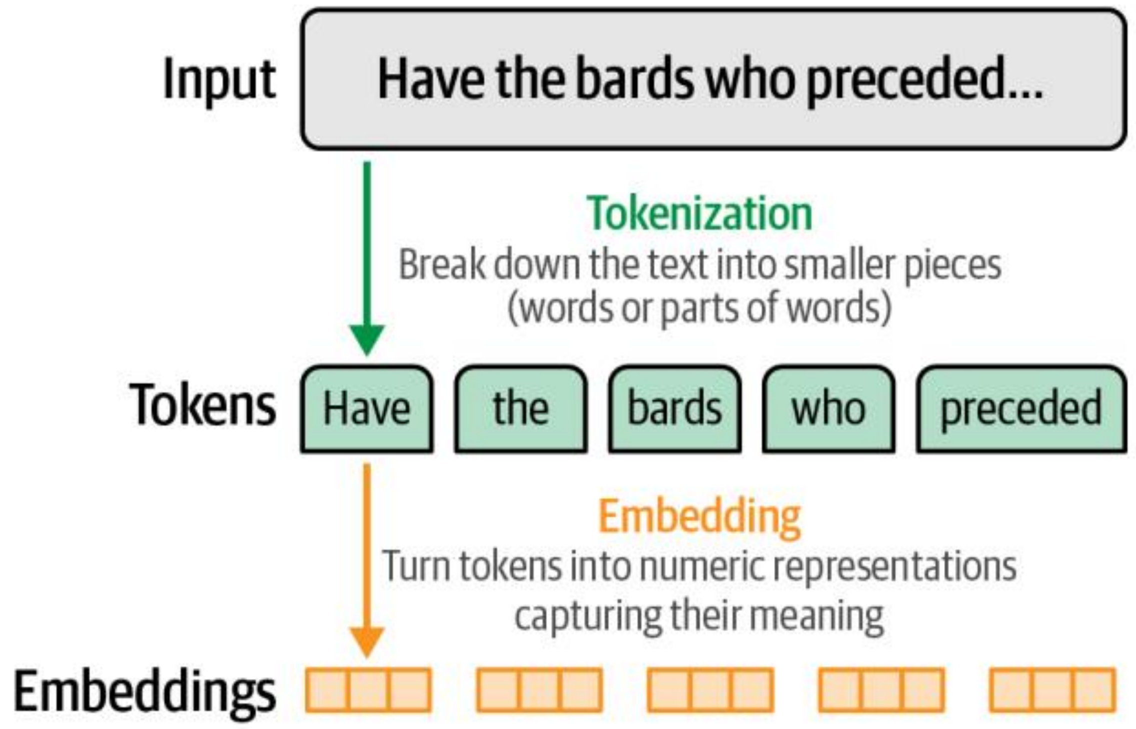
mouse → 0 0 1

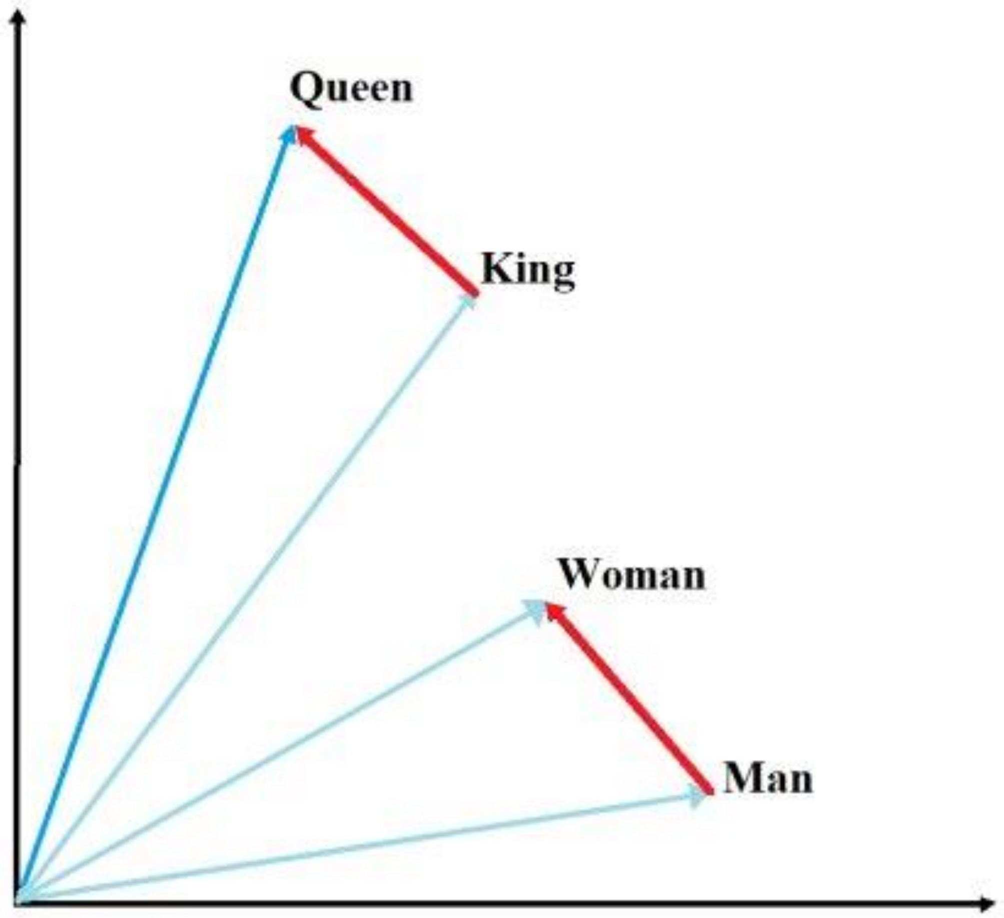
Problems…

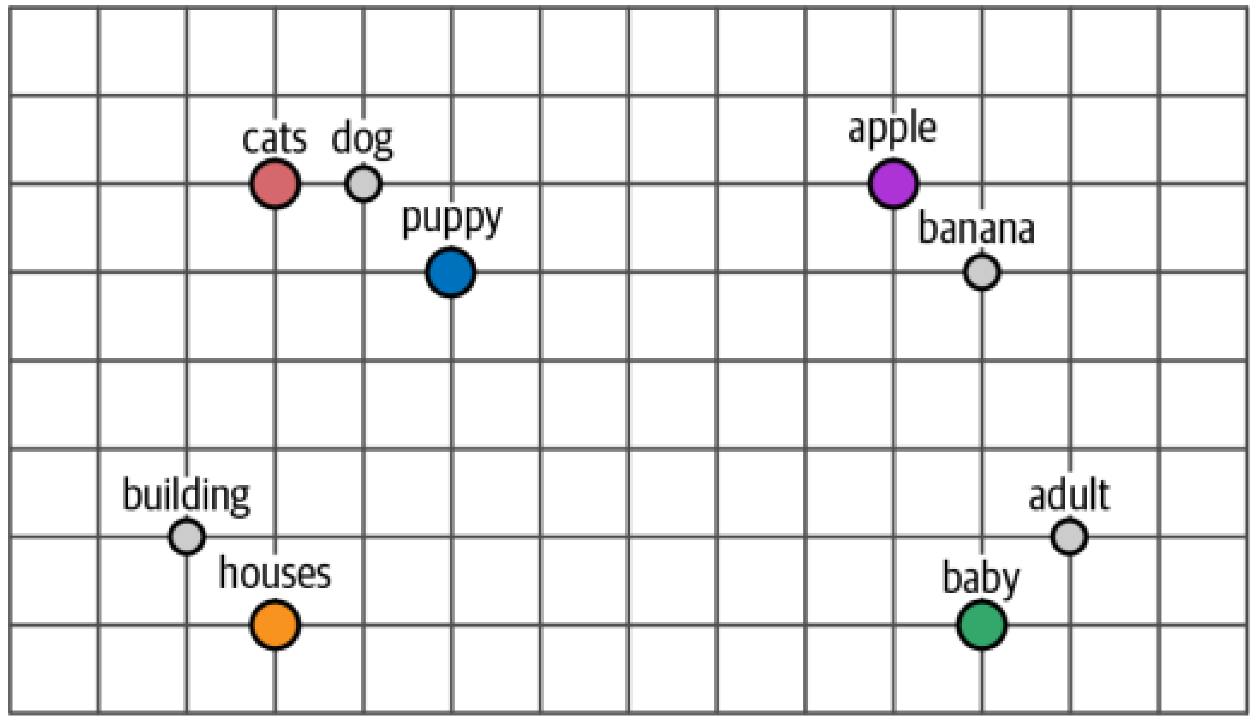
- Vector Size Problem
  - Each word becomes a vector with length equal to the total vocabulary size
  - For example, using just the Oxford English Dictionary (171,476 words): The word "cat" becomes a vector of 171,476 dimensions
  - Only one position contains 1, all others are 0
  - Every single word requires the same massive vector length
- Inefficiency Issues
  - Computationally inefficient
  - Again, no semantic relationship between words is captured

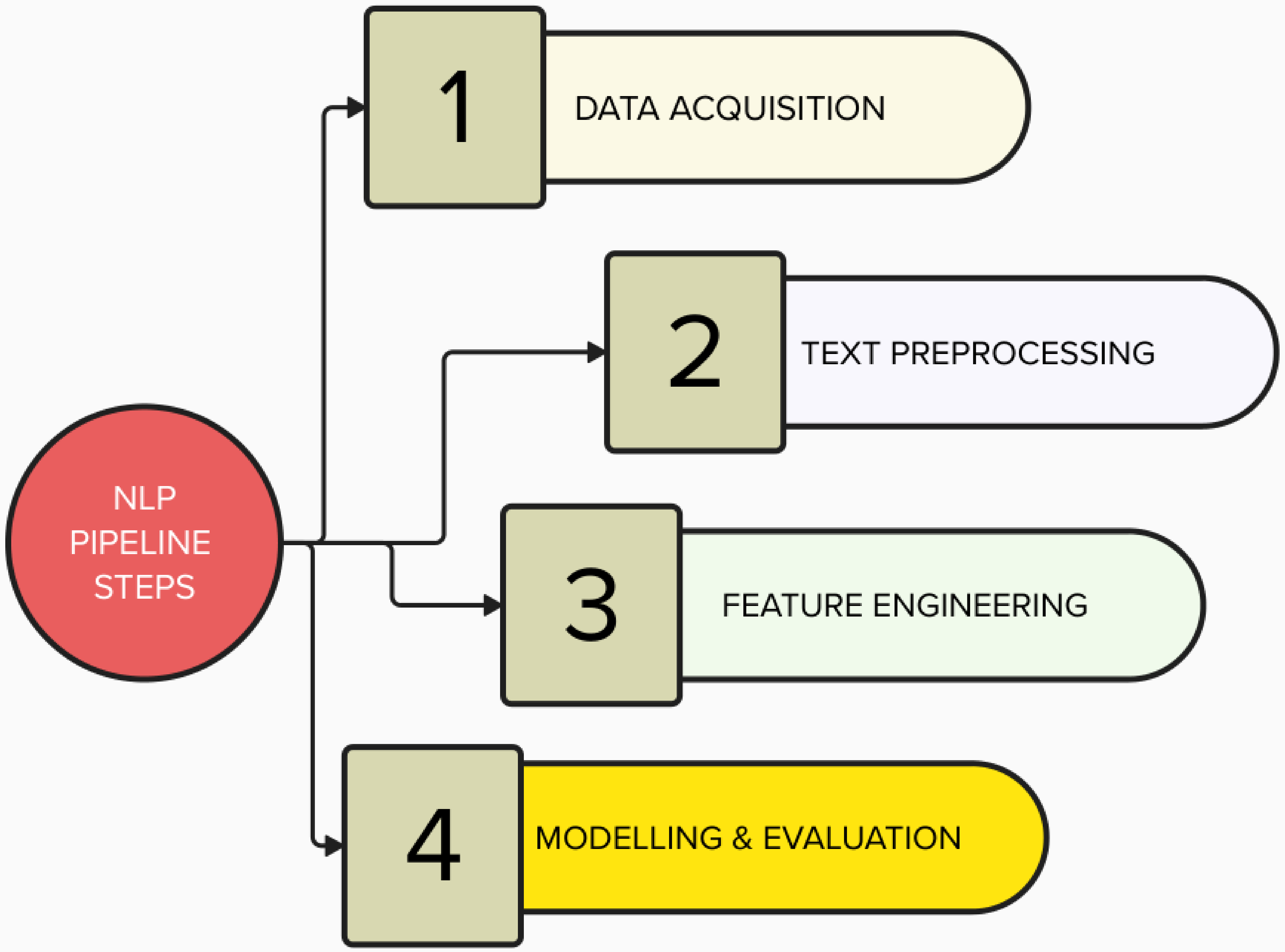
How to represent words with vector?

Embeddings





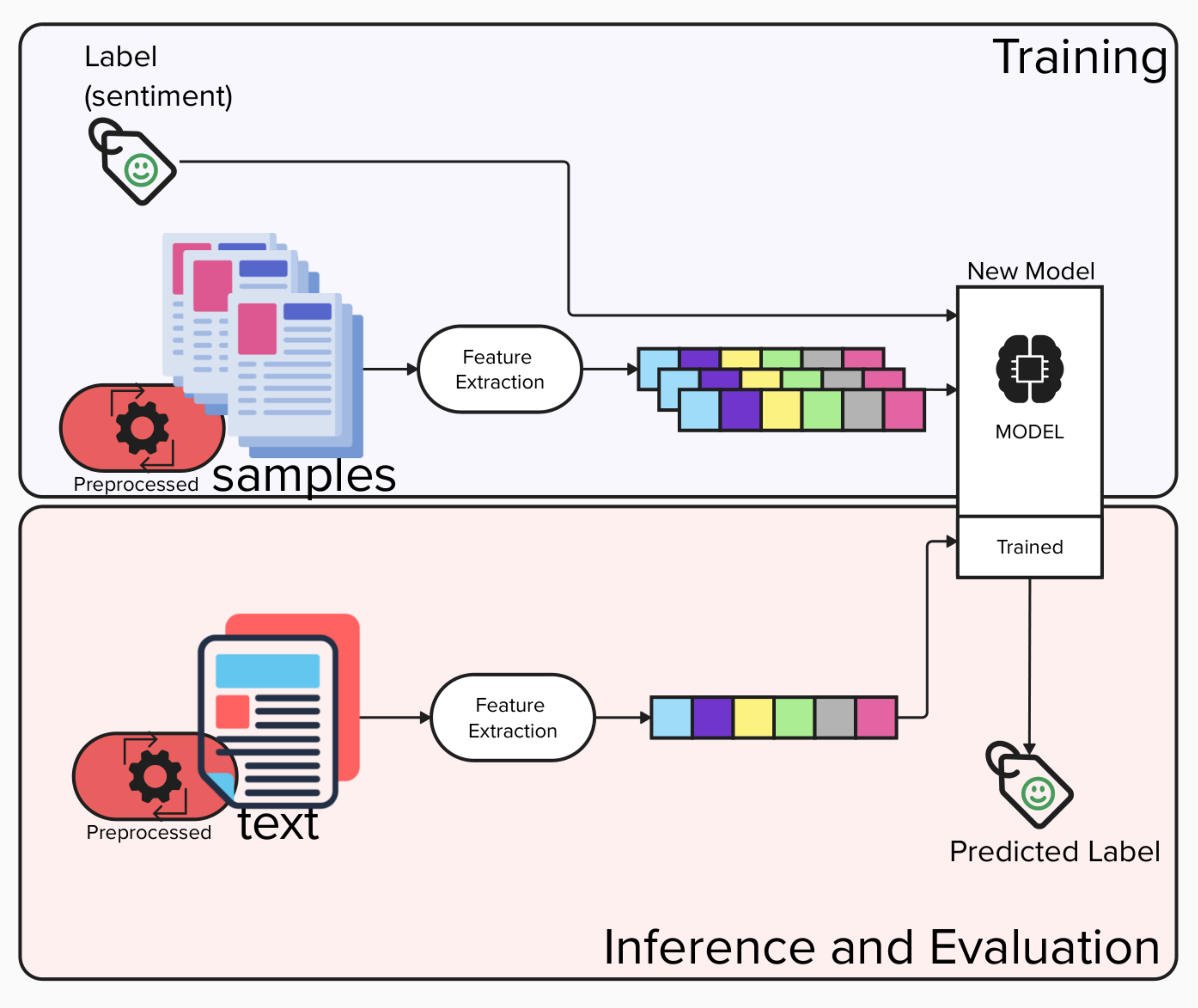




Modelling and Evaluation

- Model Development
  - Selection of appropriate algorithms (SVM, BERT, etc.)
  - Hyperparameter optimization strategies
  - Cross-validation techniques
- Model Development
  - Metrics selection and implementation
  - Error analysis and model debugging

5.4.1 Modelling and Evaluation



	Predicted Positive 😊	Predicted Negative 😟
Actually Positive 😊	120 True Positive *"Great product, exactly what I needed!" (Predicted: Positive ✓)*	20 False Negative *"This works perfectly!" (Predicted: Negative ✗)*
Actually Negative 😟	15 False Positive *"The quality is terrible" (Predicted: Positive ✗)*	145 True Negative *"Waste of money" (Predicted: Negative ✓)*

Accuracy	Precision	Recall
88.3%	88.9%	85.7%
Overall correctness	Accuracy of positive predictions	Found positive reviews

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

Hugging Face

Objective

Build a basic sentiment classifier for movie reviews using classical NLP techniques

- **Dataset**:
  - Rotten Tomatoes movie reviews
  - Binary classification: positive (1) vs negative (0) reviews
  - Training set + Test set for evaluation
- **Pipeline Steps**:
  - **Data Acquisition**
    Load the dataset using Hugging Face's datasets library
  - **Text Preprocessing**
    Convert to lowercase , Remove special characters and numbers, Remove English stopwords, Apply stemmin
  - **Feature Engineering**
    Convert text to Bag-of-Words representation (CountVectorizer)
    Transform to TF-IDF features
  - **Model Training**
    Algorithm: Multinomial Naive Bayes
    Train on TF-IDF features
  - **Evaluation**
    - Generate predictions on test set
    - Calculate accuracy, precision, recall