

# Cloud Computing



# Semantic Analysis

Tina Tavakkoli  
Parsa HedayatNia  
Hadi Amini



# Contents



- 01 Semantic Analysis
- 02 AI Model
- 03 Applications
- 04 steps
- 05 Docker Containers
- 06 Code Explanation
- 07 Work Flow
- 08 Result

”

# Semantic Analysis

Semantic analysis is the process of understanding the meaning of text by analyzing the relationships between words, phrases, and sentences. It is a crucial part of NLP that enables computers to extract deeper meaning from text data



# AI Model



## introduction

- available in the Hugging Face library.
- The model is based on the BERTweet model, which is a RoBERTa model fine-tuned on a massive dataset of English tweets.

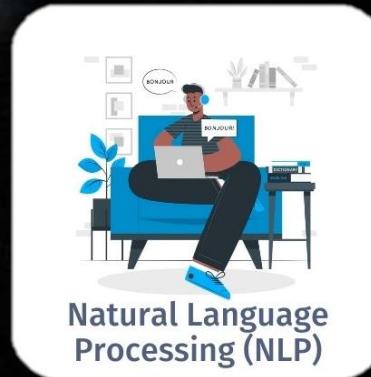
## Key capabilities

- ❖ **Sentiment classification:** The model can classify tweets as positive, negative, or neutral.
- ❖ **Sentiment intensity analysis:** The model can also determine the intensity of sentiment in tweets.
- ❖ **Generality:** The model can be used to analyze different types of tweets, including tweets about news, events, and personal opinions.

## Advantages

- ✓ **High accuracy:** The model is highly accurate in sentiment analysis of tweets.
- ✓ **Fast speed:** The model can analyze tweets quickly.
- ✓ **Ease of use:** The model is easy to use with the Hugging Face library.

# Applications of Semantic Analysis



Natural Language Processing (NLP)



Search Engines



Information Retrieval



Chatbots and Virtual Assistants

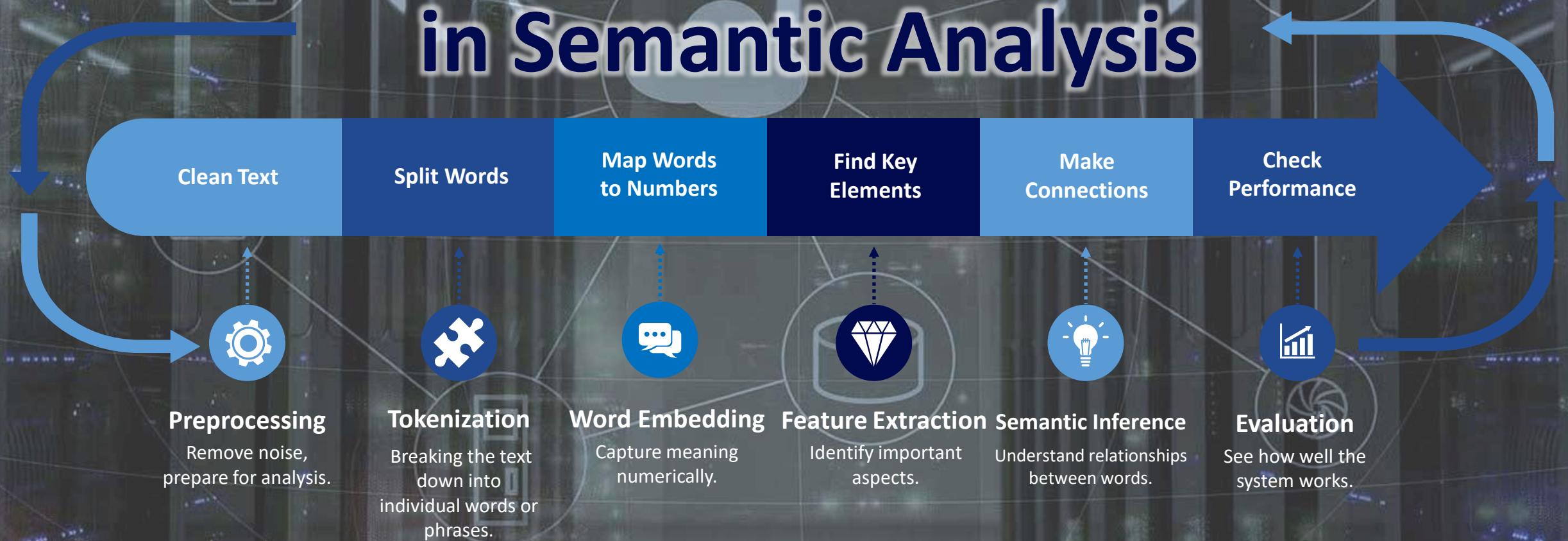


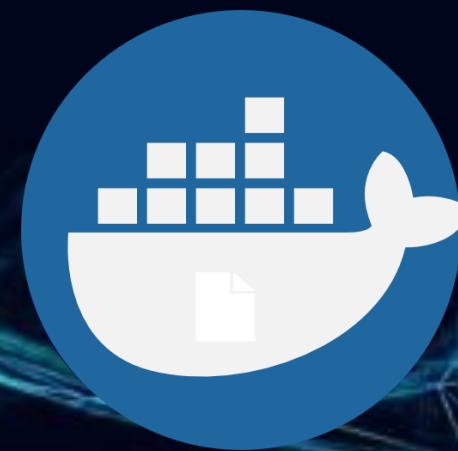
Machine Learning and AI



Customer Service and Support

# Steps in Semantic Analysis





# Docker Containers

## Name Node

Runs the HDFS Namenode service, exposes web UI on port 9870.

## Resource Manager

Runs the YARN ResourceManager service, exposes web UI on port 8088, allows mounting a custom script.



## Node manager

Runs the YARN NodeManager service, likely communicates on internal ports.

## Data Node

Runs the HDFS Datanode service, likely communicates on internal ports.

# Code Explanation



# Overall Functionality

This Java code implements a sentiment analysis program using the MapReduce framework. It takes text files as input, analyzes the sentiment of each line using a sentiment analysis API, and then determines the overall sentiment (positive, negative, or neutral) for each file based on the sentiment scores of its lines.



**Mapper  
Class**

**Run  
Method**

**Analyze  
Sentiment  
Method**

**Combiner  
Class**

**Main  
Method**

**Reducer  
Class**

## analyzeSentiment Method

- This method takes a text string as input.
- It uses an HTTP connection to send a POST request to a sentiment analysis API with the text content.
- It parses the JSON response from the API to determine the sentiment label.
- It returns 1 for positive, -1 for negative, and 0 for neutral sentiment.

## Mapper Class

- This class implements the map function, which is called for each line of the input text file.
- It extracts the text from the line and calls the analyzeSentiment method to get the sentiment score (positive, negative, or neutral)
- Based on the sentiment score, it emits a key-value pair. The key is the filename (obtained from the input split) combined with the sentiment label (e.g., "positive" , neutral or "negative"), and the value is always 1 (indicating one occurrence).

## Combiner Class

- The combiner performs a partial aggregation on the mapper outputs before sending them to the reducer.
- In this case, the combiner simply sums the counts for each key (filename-sentiment combination) locally on each mapper. This can help reduce network traffic by sending smaller intermediate data to the reducer

## Reducer Class

- This class implements the reduce function, which is called for each unique key generated by the mapper.
- It iterates through the values associated with the key (sum of counts for each sentiment category) and stores them in separate maps for positive, negative, and neutral counts.
- In the cleanup method, it determines the dominant sentiment for the file based on the count comparisons and emits a final key-value pair. The key is the filename, and the value is the dominant sentiment label ("positive", "negative", or "neutral").

## Run Method

- takes input and output paths as arguments.
- It checks for valid arguments, sets up the MapReduce job with configuration, JAR file, mapper, combiner ,reducer classes, output key/value types, and input/output paths.
- Finally, it submits the job and returns 0 for success or 1 for failure.

## Main Method

- Creates an instance of SentimentAnalysis.
- Calls the run method to execute the MapReduce job with the provided arguments.
- Exits the program with the result (0 for success, 1 for failure) returned by the run method.

## Input



File1  
File 2  
File 3

## Mapper

("file1  
("file1  
neutral", 1)  
negative ", 1)

("file2  
("file2  
neutral", 1)  
neutral", 1)  
neutral ", 1)  
negative", 1)

("file3  
("file3  
neutral", 1)  
positive", 1)  
positive", 1)

## Combiner

("file1  
("file1  
neutral", 1)  
negative ", 1)

("file2  
("file2  
neutral", 3)  
negative", 1)

("file3  
("file3  
neutral", 1)  
positive", 2)

## Reducer

("file1  
("file1  
neutral", 1)  
negative ", 1)  
("file2  
("file2  
neutral", 3)  
negative", 1)  
("file3  
("file3  
neutral", 1)  
positive", 2)

## Clean Up

(file1 , negative)  
(file2 , neutral)  
(file3 , positive)

# Work Flow

Run job

ResourceManager UI

Final Result

```
2024-06-14 13:12:04 INFO  YarnRMClientImpl:558 - Submitted application application_1718348039586_0026
2024-06-14 13:12:04 INFO  Job:1682 - The url to track the job: http://resourcemanager:8088/proxy/application_1718348039586_0026/
2024-06-14 13:12:04 INFO  Job:1727 - Running job: job_1718348039586_0026
2024-06-14 13:12:12 INFO  Job:1748 - Job job_1718348039586_0026 running in uber mode : false
2024-06-14 13:12:12 INFO  Job:1755 - map 0% reduce 0%
2024-06-14 13:12:38 INFO  Job:1755 - map 33% reduce 0%
2024-06-14 13:12:42 INFO  Job:1755 - map 50% reduce 0%
2024-06-14 13:12:43 INFO  Job:1755 - map 67% reduce 0%
2024-06-14 13:12:44 INFO  Job:1755 - map 100% reduce 0%
2024-06-14 13:12:46 INFO  Job:1755 - map 100% reduce 100%
2024-06-14 13:12:46 INFO  Job:1766 - Job job_1718348039586_0026 completed successfully
2024-06-14 13:12:47 INFO  Job:1773 - Counters: 55
```

```
File System Counters
  FILE: Number of bytes read=648
  FILE: Number of bytes written=1946445
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=7135
  HDFS: Number of bytes written=179
  HDFS: Number of read operations=23
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0

Job Counters
  Killed map tasks=1
  Launched map tasks=6
  Launched reduce tasks=1
  Rack-local map tasks=6
  Total time spent by all maps in occupied slots (ms)=143975
  Total time spent by all reduces in occupied slots (ms)=5386
  Total time spent by all map tasks (ms)=143975
  Total time spent by all reduce tasks (ms)=5386
  Total vcore-milliseconds taken by all map tasks=143975
  Total vcore-milliseconds taken by all reduce tasks=5386
  Total megabyte-milliseconds taken by all map tasks=147430400
  Total megabyte-milliseconds taken by all reduce tasks=5515264
```

```
Map-Reduce Framework
```

```
  Map input records=87
  Map output records=87
  Map output bytes=2939
```

# Work Flow

Run job

ResourceManager UI

localhost:8088

Final Result

### All Applications

Used Resources	Total Resources	Reserved Resources								
<memory:8 GB, vCores:7>	<memory:8 GB, vCores:8>	<memory:0 B, vCores:0>								
Decommissioned Nodes		Rebooted Nodes								
0		0								
Lost Nodes		Unhealthy Nodes								
0		0								
Minimum Allocation		Maximum Cluster Application Priority								
<memory:8192, vCores:4>		0								
LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB	Allocated GPUs	Reserved CPU Vcores	Reserved Memory MB	Reserved GPUs
Jun 14 42:04 +0350 ?4	N/A	RUNNING	UNDEFINED	7	7	8192	-1	0	0	-1
Jun 14 40:17 +0350 ?4	Fri Jun 14 16:41:04 +0350 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A

# Work Flow

Run job

ResourceManager UI

Final Result

```
bash-4.2$ hdfs dfs -cat output23/part-r-00000  
combined_text_14.txt      neutral  
combined_text_12.txt      negative  
combined_text_13.txt      positive  
combined_text_10.txt      negative  
combined_text_15.txt      negative  
combined_text_16.txt      positive  
bash-4.2$
```



THANK YOU