IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Student Name :- Kalusaykkarage Sugeeshwara	
Student Reference Number : **10899493**	

Module Code: PUSL 3189	**Module Name: Natural Language Processing (NLP)**
Coursework Title: Final Individual assignment (Report)	
Deadline Date: 02 | 01 | 2025	**Member of staff responsible for coursework: Mrs. Nethmi Weerasinghe**
Program: BSc.(Hons) in Data Science	

Signed: ***S***

I *have ~~used~~ / not used translation software.

If used, please state name of software………………………………………………………………

NLP Individual Coursework

PUSL 3189

WordCount – 5200

(For main tasks analysis, insights)

Additional tasks : Model performing – 1700 Words

Contents

1. Introduction (Task 1)

Brief Introduction to NLP and Its Applications

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) focused on enabling machines to understand, interpret, and generate human language. NLP has numerous applications, including:

- **Sentiment Analysis**: Determining the emotional tone behind words.
- **Machine Translation**: Automatically converting text from one language to another.
- **Chatbots**: Simulating conversations with human users.
- **Text Summarization**: Condensing large volumes of text into shorter, more digestible formats.

Objectives of the Project

The project aims to develop two key components:

1. **Human-AI Text Classifier:**
    - **Purpose**: To identify difference between text generated by humans and text generated by AI.
    - **Importance**: Make sure the authenticity and integrity of content.
2. **AI-to-Humanized Text Converter:**
    - **Purpose**: To transform AI-generated text into a more natural, human-like format.
    - **Importance**: Enhances the readability and acceptability of AI-generated content, making it suitable for publication and broader use.

Why I have Chosen This Application

- **Knowledge Sharing:**

    - **Problem**: Many experts lack the writing ability to publish their knowledge effectively.
    - **Solution**: AI can assist in writing, but the formal and patterned nature of AI-generated text often hinders publication due to AI checkers.

- **AI Writing Challenges:**
    - **Formal Language**: AI tends to write in a formal manner with identifiable patterns.
    - **Publication Barriers**: AI-generated text may not pass AI checkers, preventing publication. For the experts that lack of writing skills
- **Program Benefits:**
    - **Classification**: The Human-AI Text Classifier can determine the probability of text is AI-generated or not.
    - **Conversion**: The AI-to-Humanized Text Converter can transform AI-generated text into a more natural, human-like format, making it suitable for publication.

In conclusion, the development of a Human-AI Text Classifier and an AI-to-Humanized Text Converter addresses needs in content creation and publication. These tools enhance the authenticity , readability of AI-generated content, making them invaluable in an era where AI-generated content is becoming increasingly prevalent.

2. Dataset Collection and Description (Task 1)

2.1. OVERVIEW OF THE DATASET USED:

Talking about the data set is mainly showing the sentence written by humans, and the exact same sentence when it written by AI . So, the data set is having basically 3 main components

doc_id : This is to identify where the text comes from.

sentence: This is a sentence that can be confirmed (the logic will be explained in the next content) written by a human

ai_sentence: The exact same sentence represents, How it would be if it written by AI

2.2. Data collection process:

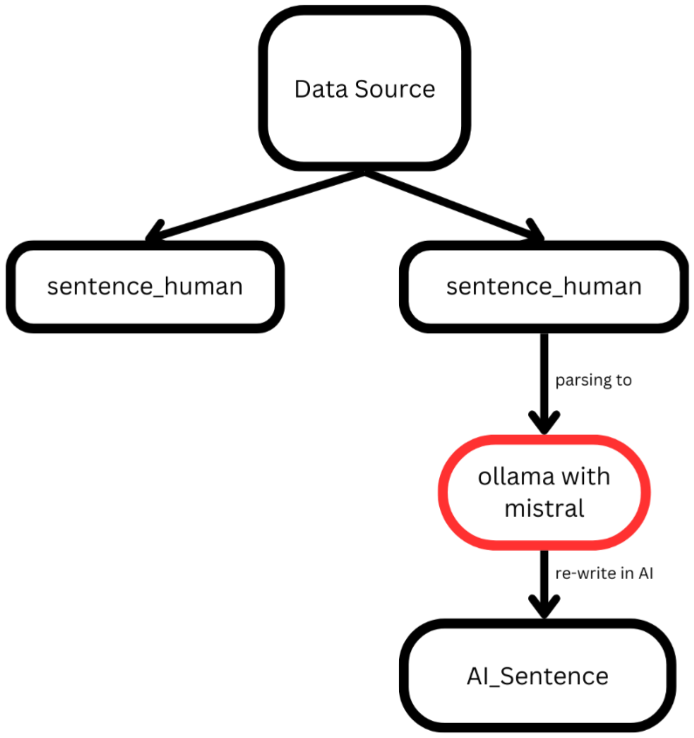
My goal was to gather data from articles as those articles are written by humans. And make an analysis for that. Tried this approach which, scraping from websites like medium, simple learn, dev.to But faced restrictions for scraping like html structure like. nested div class which is hard to scrape and mine

And also then tried to make a python script to google search passing a query using selenium and googlesearch libraries. Then extract the most relevant links. And tried to extract the contents from them. but unnecessary content extracted and failed.

And then found a site from “tqwed” which stores articles . inside the website . First I have extracted those and filter all unnecessary links . and then extracted the Contents from the article .

Next problem was making sure all the articles was written before the AI for content writing was spreading . To achieve this ,have extracted all the time tags in the article and remove all the links that are above from 2020 . And successfully gained like 83 articles , which is one containing like 4000+ words. Which was good

2.2.1. How was the ai_sentences column came to the DataSet



Aspect of the dataset, To perform an analysis, for the classification model .needed a data set to spot the difference between Ai sentences and Human sentences for that .made an activity flow as the above diagram shows.

The diagram showed a workflow in which human-written sentences processed by an AI model (Ollama with Mistral) to generate AI-rewritten sentences. When considering about the articles. Those articles needed to rewrite in AI . but it was really heavy to pass it to a LLM.. So Article were spitted into sentences and it was then pass it to the model and rewrote in AI. Which gave 2500 sentences which was good.

---

3. Preprocessing and Tokenization (Task 2)

3.1. preprocessing steps:

Preprocessing is a crucial step in NLP which prepares raw text information for analyses. In my program, who focused on distinguishing between AI-created and human-authored text, effective preprocessing guarantees precise and meaningful responses. In my application, As application follows there was separated sections that used the preprocessing part. So the comparison table would be help for the readability and evaluate

	tokenization	Stop_words	Punctuation	Stemming / Lemmatizing	n-gram	normalization	Reason for steps
Classification_model	✓	✓	✓	✗	✓	✓	**Tokenization**: Converts text into processable units. **Stop Words**: Removes noise, focusing on relevant words. **Punctuation**: Reduces noise unless punctuation is a feature. **Stemming/Lemmatizing**: Avoided to retain word nuances. **N-grams**: Captures contextual patterns between words. **Normalization**: Ensures consistency in text.
Document_clustering	✓	attempted	attempted	attempted	✗	✓	**Tokenization**: Breaks text into meaningful units for clustering. **Stop Words**: Reduces dimensionality, improving clustering quality. **Punctuation**: Tried but clustering handled it inherently. **Stemming/Lemmatizing**: Tried but variations helped cluster distinctions. **N-grams**: Captures phrases to improve clustering coherence. **Normalization**: Standardizes text for better cluster consistency.
Sentiment_analysis	✓	✗	✗	✗	✗	✗	**Tokenization**: Splits text into words for sentiment detection. **Stop Words**: Retained as stop words can carry sentiment. **Punctuation**: Retained because punctuation conveys sentiment. **Stemming/Lemmatizing**: Avoided to retain sentiment word forms. **N-grams**: Avoided as unigram models suffice. **Normalization**: Ensures uniform input for analysis.
Stylometric_analysis	✓	✓	✓	✗	✗	✓	**Tokenization**: Enables word-level stylistic analysis. **Stop Words**: Retained as they reflect stylistic features. **Punctuation**: Retained because it's integral to style. **Stemming/Lemmatizing**: Avoided to preserve stylistic nuances. **N-grams**: Avoided as unigrams capture style effectively. **Normalization**: Standardizes text for stylistic comparison.
Topic_modeling	✓	✓	✓	✗	✗	✓	**Tokenization**: Prepares text for identifying topics. **Stop Words**: Removes noise to focus on topic-specific words. **Punctuation**: Reduces irrelevant symbols for modeling. **Stemming/Lemmatizing**: Avoided to keep word variations for topics. **N-grams**: Captures multi-word phrases relevant to topics. **Normalization**: Ensures consistent text for topic coherence.

AI_to_human_converter	✓	✗	✗	✗	✓	✓	**Tokenization**: Splits text into units for targeted replacements. **Stop Words**: Retained because stop words are essential for conversational tone. **Punctuation**: Retained to preserve natural sentence structure. **Stemming/Lemmatizing**: Avoided to maintain word variety and fluency. **N-grams**: Used to identify and replace formal phrases with casual alternatives. **Normalization**: Ensures text consistency for smooth replacements.

4. POS Tagging and Named Entity Recognition (Task 3)

POS tagging was mainly used for the classification model to perform well.

4.1. POS tagging for the classification model

A. POS-Tag Preprocessing

1. Count POS Tags:

- Convert the frequency of **POS** tags to a count dictionary.
- **Example** : "**NN VB DT NN**" -- > {NN: 2, VB: 1, DT: 1}.

2. Normalize POS Frequencies:

- Divide the count for each POS tag by the total number of tags to get the frequency.
- Example : {**NN: 2, VB: 1, DT: 1**} -- > {**NN: 0.5, VB: 0.25, DT: 0.25**}.
- This normalization method really helps in standardizing the data, that would affect models' learning patterns regardless of the text length.

B. POS Tag Features

1) To Create Features those POS tags were taken:

- Used the normalized POS frequencies as features.

- So the model would understand based on POS tags to capture syntactic patterns that may differ between human and AI-generated text. For the classification.

Discussion

The text classification model detects lexical and grammatical differences between AI and human writing styles using TF-IDF vectorization and POS tag importance. By using grammatical and semantic data, this method would be increase overall accuracy in classification by identifying

4.2. Name Entity Recognition Analysis

This method would be used for identifying the NER of human written and AI written sentences to analyze and identify if the AI written sentences would harm to the NER of the human writing sentences

Data Preprocessing

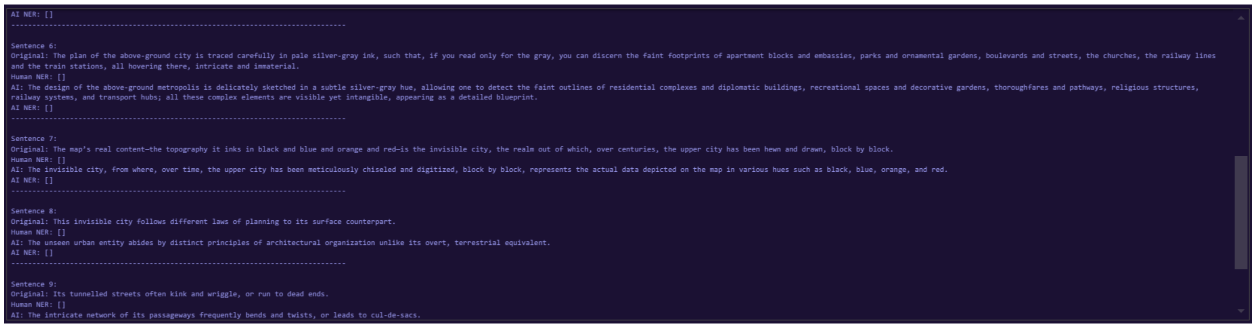
- Loaded the English language model with NER support using SpaCy.

NER Tag Extraction

- Defined a function **get_ner_tags** to extract named entities from text.
- Applied this function to both human and AI-generated sentences to get NER tags.

Comparison DataFrame

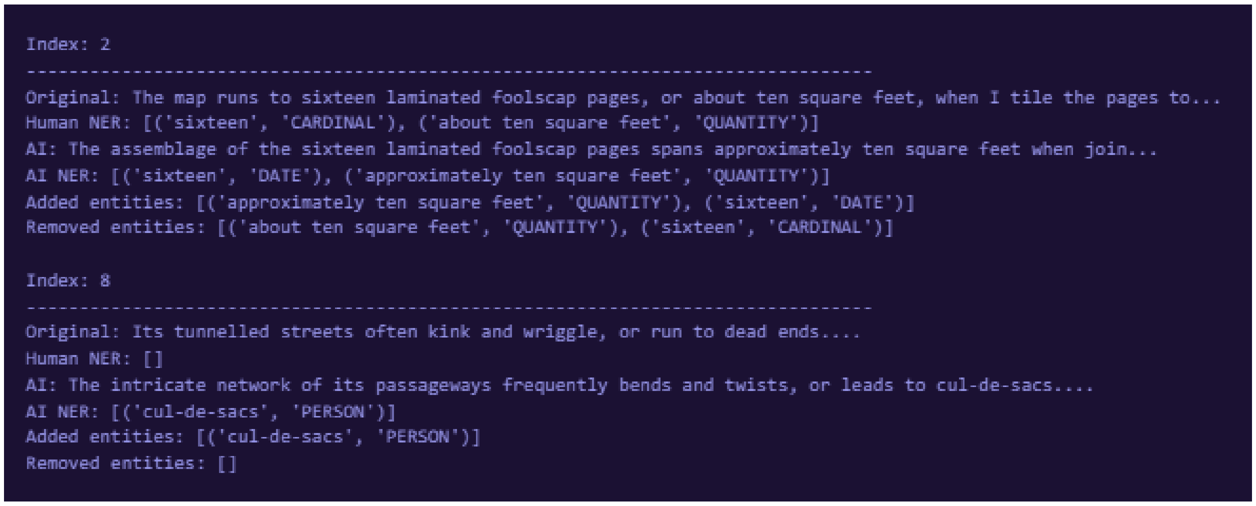
- Created a DataFrame to compare NER tags between human and AI texts.
- Displayed the comparison for the first 10 sentences.



```
AI NER: []
--------------------------------------------------------------------------------

Sentence 6:
Original: The plan of the above-ground city is traced carefully in pale silver-gray ink, such that, if you read only for the gray, you can discern the faint footprints of apartment blocks and embassies, parks and ornamental gardens, boulevards and streets, the churches, the railway lines
and the train stations, all hovering there, intricate and immaterial.
Human NER: []
AI: The design of the above-ground metropolis is delicately sketched in a subtle silver-gray hue, allowing one to detect the faint outlines of residential complexes and diplomatic buildings, recreational spaces and decorative gardens, thoroughfares and pathways, religious structures,
railway systems, and transport hubs; all these complex elements are visible yet intangible, appearing as a detailed blueprint.
AI NER: []
--------------------------------------------------------------------------------

Sentence 7:
Original: The map's real content—the topography it inks in black and blue and orange and red—is the invisible city, the realm out of which, over centuries, the upper city has been hewn and drawn, block by block.
Human NER: []
AI: The invisible city, from where, over time, the upper city has been meticulously chiseled and digitized, block by block, represents the actual data depicted on the map in various hues such as black, blue, orange, and red.
AI NER: []
--------------------------------------------------------------------------------

Sentence 8:
Original: This invisible city follows different laws of planning to its surface counterpart.
Human NER: []
AI: The unseen urban entity abides by distinct principles of architectural organization unlike its overt, terrestrial equivalent.
AI NER: []
--------------------------------------------------------------------------------

Sentence 9:
Original: Its tunnelled streets often kink and wriggle, or run to dead ends.
Human NER: []
AI: The intricate network of its passageways frequently bends and twists, or leads to cul-de-sacs.
```

Identifying NER Differences

- Compared NER tags between human and AI sentences for the first 10 entries.
- Identified and printed differences in named entities, including added and removed entities.



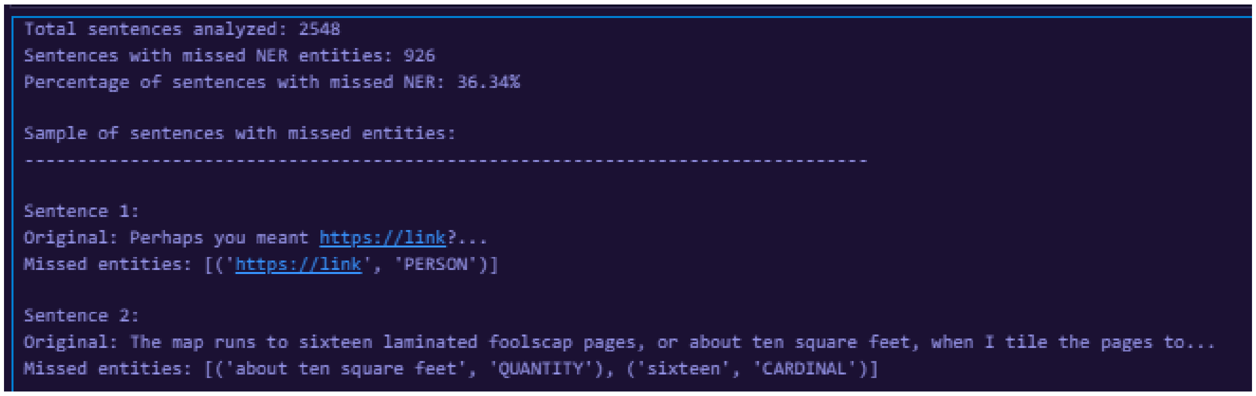
```
Index: 2
--------------------------------------------------------------------------------
Original: The map runs to sixteen laminated foolscap pages, or about ten square feet, when I tile the pages to...
Human NER: [('sixteen', 'CARDINAL'), ('about ten square feet', 'QUANTITY')]
AI: The assemblage of the sixteen laminated foolscap pages spans approximately ten square feet when join...
AI NER: [('sixteen', 'DATE'), ('approximately ten square feet', 'QUANTITY')]
Added entities: [('approximately ten square feet', 'QUANTITY'), ('sixteen', 'DATE')]
Removed entities: [('about ten square feet', 'QUANTITY'), ('sixteen', 'CARDINAL')]

Index: 8
--------------------------------------------------------------------------------
Original: Its tunnelled streets often kink and wriggle, or run to dead ends....
Human NER: []
AI: The intricate network of its passageways frequently bends and twists, or leads to cul-de-sacs....
AI NER: [('cul-de-sacs', 'PERSON')]
Added entities: [('cul-de-sacs', 'PERSON')]
Removed entities: []
```

Full Dataset Analysis

- Used whole the NER comparison.
- Identified sentences, AI model missed named entities compared to the human text.
- Calculated summary statistics:

Results

- Printed summary statistics and examples of sentences with missed entities.



```
Total sentences analyzed: 2548
Sentences with missed NER entities: 926
Percentage of sentences with missed NER: 36.34%

Sample of sentences with missed entities:
--------------------------------------------------------------------------------

Sentence 1:
Original: Perhaps you meant https://link?...
Missed entities: [('https://link', 'PERSON')]

Sentence 2:
Original: The map runs to sixteen laminated foolscap pages, or about ten square feet, when I tile the pages to...
Missed entities: [('about ten square feet', 'QUANTITY'), ('sixteen', 'CARDINAL')]
```

The statistics provided in the image give a quantitative overview of the performance of the Named Entity Recognition (NER) system:

1. Total Sentences Analyzed: 2544 sentences.
2. Sentences with Missed NER Entities: 926 sentences' NER system failed to recognize.
3. Percentage of Sentences with Missed NER: This amounts to approximately 36.34% of the total sentences analyzed, indicating that more than one-third of the sentences had at least one missed entity.

These statistics highlight that there is a huge area for improvement in the AI writing systems. This suggests that the AI sentences really harmed the Entities of the original text when writing int the AI such as URLs, person names, quantities, and cardinal numbers.

---

5. Analysis for the Model Design

5.1. Analysis for the implementation (task 4,5,6,7 ,8)

5.1.1. Using Sentiment Analysis to Highlight the Model's Importance (Task 4)

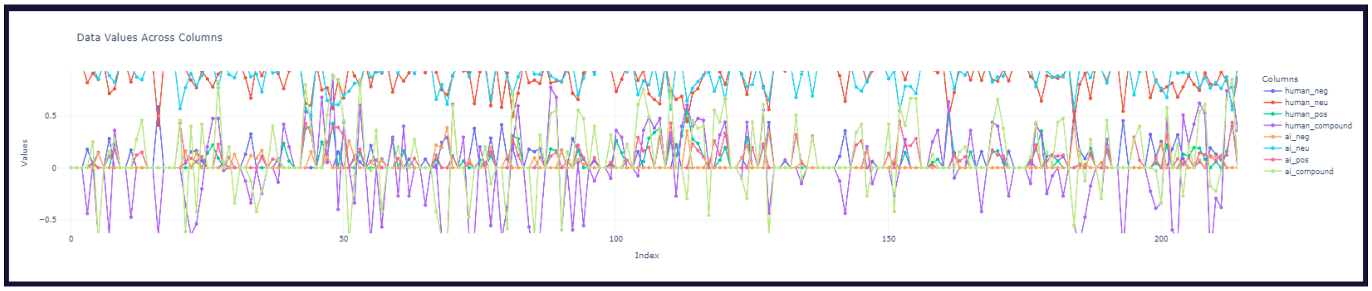
To understand how AI-generated would harm texts in emotional tone,conducted a sentiment analysis using the **VADER** tool. This tool helps measure the sentiment of each sentence in the texts. My goal was to identify these differences and find ways to minimize them.

Methodology

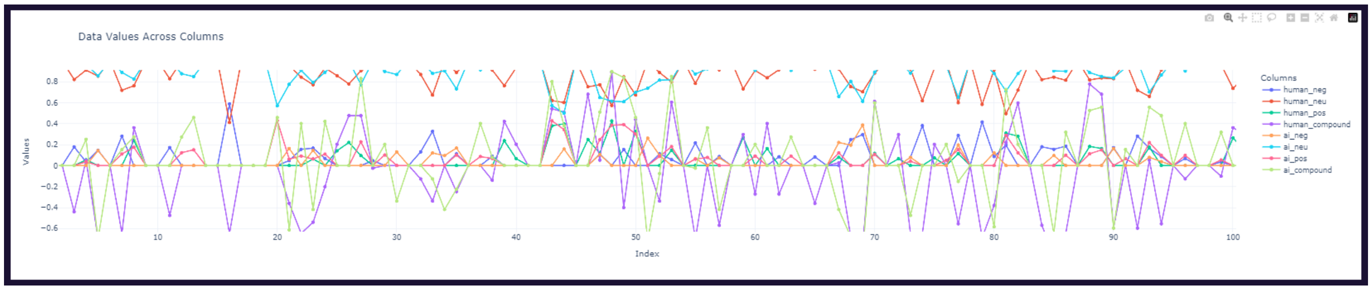
1. Sentiment Scoring:

    - I used VADER , **SentimentIntensityAnalyzer** to analyze each sentence in both datasets.

      This gave me scores for positive, negative, neutral, and compound sentiments.

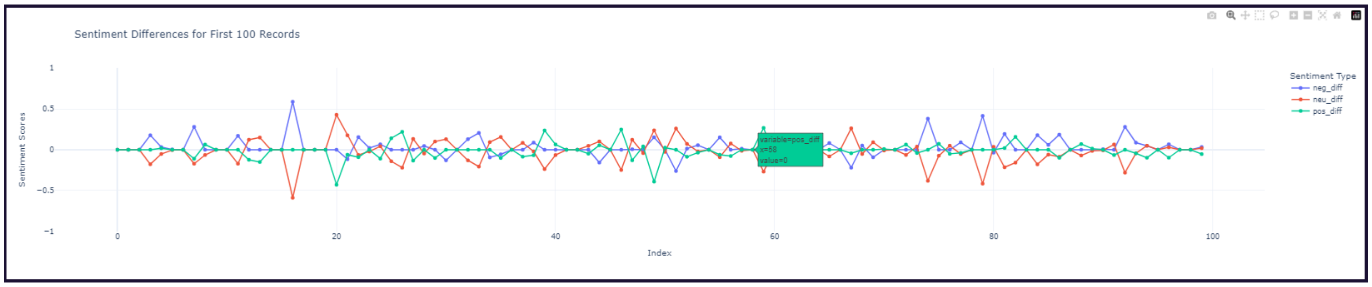


When inspect the visualization It clearly shows at most of data points there is significant difference occurred that should not happen



2. Difference Calculation:

- To measure how different the emotions have calculated the difference in sentiment scores between AI-generated and human-written texts for each category (positive, negative, neutral, and compound).



This visualization really shows how difference each sentence would get for the emotion

Discussion on Differences

These significant differences need to be addressed somehow. Target would be to minimize those differences by application of mine

5.1.2. Topic modeling to analyze content affection (Task 5)

The topic modeling approach have done using two methods. Which were:

1) LatentDistinctAllocation (LDA) with TF-IDF
2) Gensim LdaModel with CoherenceModel

LatentDistinctAllocation (LDA) with TF-IDF

Data Preprocessing

First,have preprocess the text data to make remove any noise. This includes:

- Converting Text to Lowercase
- Removing Special Characters
- Filtering Out Stopwords

TF-IDF Vectorization

Next, Convert the preprocessed text into numbers using the **TF-IDF vectorizer**. This helps to understand the importance of each word in a document while reducing the impact of common words.

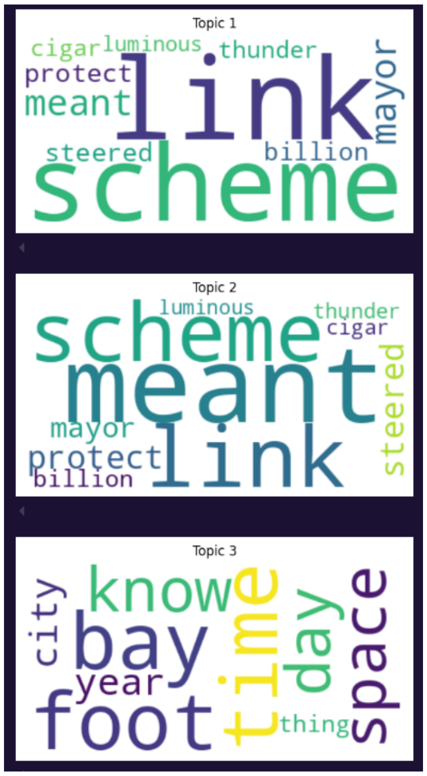
Latent Dirichlet Allocation (LDA)

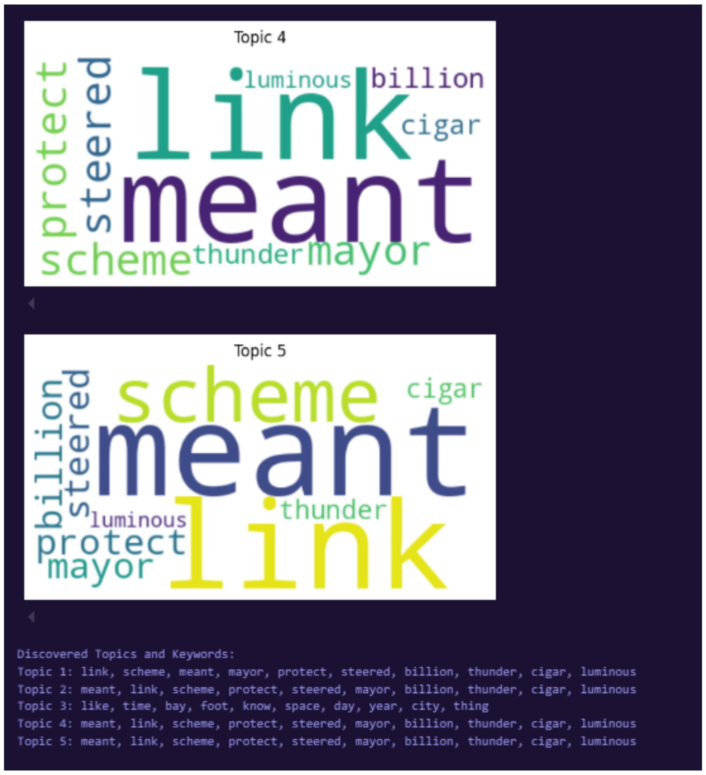
Then applied **LDA**, a probabilistic topic modeling technique, to the **TF-IDF** matrix. LDA helps us find hidden topics by looking at how words are used and how they appear together across different documents. Set the number of topics to 6 for this analysis.

Keyword Extraction

For each topic identified by **LDA**, extract the most important words (keywords) that define the topic. These keywords help understand what each topic is about. With a respective word cloud. Create word clouds to visually show the most important words in each topic. This gives us a quick and easy way to see the main themes in the text.

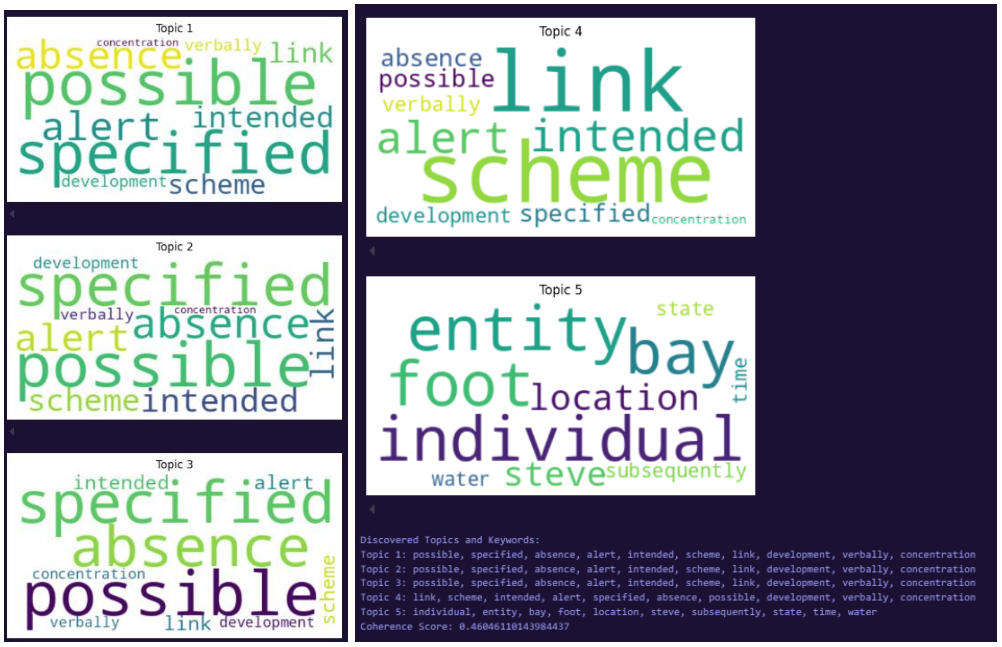
For the Human Sentences


Topic 1
cigar luminous thunder protect link meant mayor steered billion scheme
Topic 2
scheme luminous thunder cigar meant mayor protect link steered billion
Topic 3
city know bay time day space year foot thing


Topic 4
protect steered link luminous billion cigar meant scheme thunder mayor
Topic 5
billion steered scheme cigar meant luminous thunder protect mayor link

```
Discovered Topics and Keywords:
Topic 1: link, scheme, meant, mayor, protect, steered, billion, thunder, cigar, luminous
Topic 2: meant, link, scheme, protect, steered, mayor, billion, thunder, cigar, luminous
Topic 3: like, time, bay, foot, know, space, day, year, city, thing
Topic 4: meant, link, scheme, protect, steered, mayor, billion, thunder, cigar, luminous
Topic 5: meant, link, scheme, protect, steered, mayor, billion, thunder, cigar, luminous
```

For the AI_Sentences



```
Discovered Topics and Keywords:
Topic 1: possible, specified, absence, alert, intended, scheme, link, development, verbally, concentration
Topic 2: possible, specified, absence, alert, intended, scheme, link, development, verbally, concentration
Topic 3: possible, specified, absence, alert, intended, scheme, link, development, verbally, concentration
Topic 4: link, scheme, intended, alert, specified, absence, possible, development, verbally, concentration
Topic 5: individual, entity, bay, foot, location, steve, subsequently, state, time, water
Coherence Score: 0.46046110143984437
```

But as the above word cloud and key word extraction’s visualize not giving accurate topics and keywords. So that was the reason that had to try with the **Gensim LdaModel**. The implementation process will be shown below for that

<u>Gensim LdaModel with CoherenceModel</u>

This approach uses the Gensim library for Latent Dirichlet Allocation (LDA) modeling. Here are the stepsfollowed:

Data Preprocessing

1. Convert to Lowercase

2. Remove Special Characters

3. Tokenization

4. Remove Stopwords

5. Lemmatizatio

Corpus and Dictionary Creation

1. Create Dictionary:

    - Built a dictionary that maps each unique word to a unique integer ID.

2. Create Corpus:

- The **doc2bow** function transforms each document (a list of tokens) into a **Bag-of-Words (BoW)** representation, which is a list of tuples where each tuple contains a word's ID and its frequency in the document.

LDA Model Construction

1. Build LDA Model:

    - Use the **Gensim** library to build the **LDA** model, setting the number of topics to **6**.

2. Train the Model:

    - **The passes=10** parameter controls the number of times the algorithm iterates over the entire corpus for improved convergence.

Topic Interpretation and Labeling

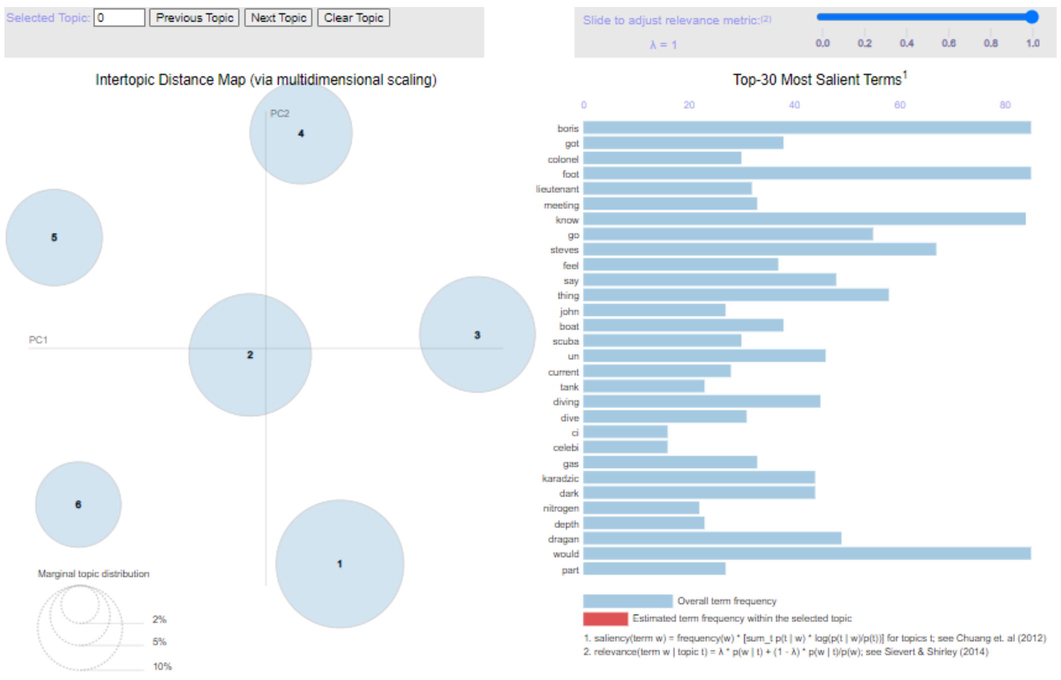
1. Extract Top Words:

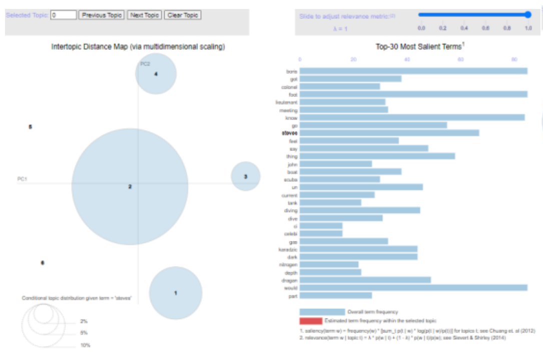
    - Identify the top words for each topic.:

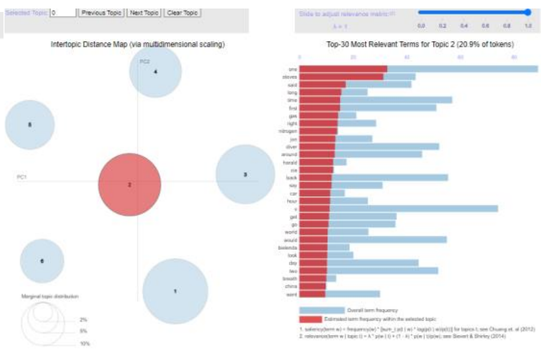
Visualization with pyLDAvis

1. Interactive Visualization:

    - I use the **pyLDAvis** tool to create an interactive visualization of the topics. Which was really impressive and showed clickable word frequencies. This visualization shows the relationships between topics and the most relevant words within each topic.


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC1
PC2
1
2
3
4
5
6
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms1
0 20 40 60 80
boris
got
colonel
foot
lieutenant
meeting
know
go
steves
feel
say
thing
john
boat
scuba
un
current
tank
diving
dive
ci
celebi
gas
karadzic
dark
nitrogen
depth
dragan
would
part
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Salient Terms1
Conditional topic distribution given term = 'steves'

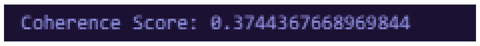

Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Relevant Terms for Topic 2 (20.9% of tokens)

2. Save Visualization:

- The visualization is saved as an HTML file for further needs.

Topic Coherence Evaluation

1. Compute Coherence Score:

- Calculate the **coherence score** .This score checks the semantic similarity of words within a topic, to make sure the topics are meaningful.
- 

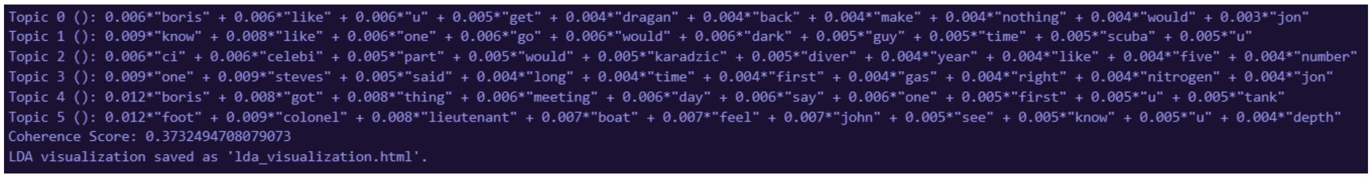
```
Coherence Score: 0.3744367668969844
```

Key Words extracted

Finally,list the topics and their keywords for further analysis. This helps us understand the themes present in the text

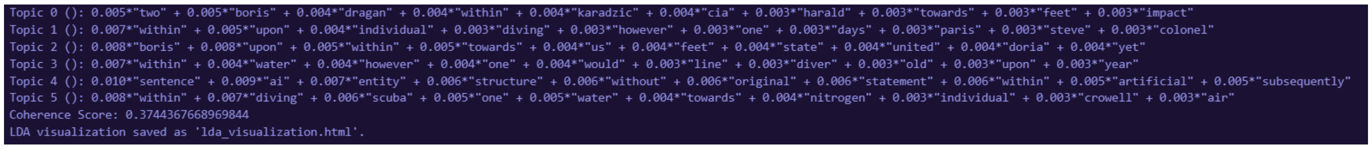
For the Human sentences

.



```
Topic 0 (): 0.006*"boris" + 0.006*"like" + 0.006*"u" + 0.005*"get" + 0.004*"dragan" + 0.004*"back" + 0.004*"make" + 0.004*"nothing" + 0.004*"would" + 0.003*"jon"
Topic 1 (): 0.009*"know" + 0.008*"like" + 0.006*"one" + 0.006*"go" + 0.006*"would" + 0.006*"dark" + 0.005*"guy" + 0.005*"time" + 0.005*"scuba" + 0.005*"u"
Topic 2 (): 0.006*"ci" + 0.006*"celebi" + 0.005*"part" + 0.005*"would" + 0.005*"karadzic" + 0.005*"diver" + 0.004*"year" + 0.004*"like" + 0.004*"five" + 0.004*"number"
Topic 3 (): 0.009*"one" + 0.009*"steves" + 0.005*"said" + 0.004*"long" + 0.004*"time" + 0.004*"first" + 0.004*"gas" + 0.004*"right" + 0.004*"nitrogen" + 0.004*"jon"
Topic 4 (): 0.012*"boris" + 0.008*"got" + 0.008*"thing" + 0.006*"meeting" + 0.006*"day" + 0.006*"say" + 0.006*"one" + 0.005*"first" + 0.005*"u" + 0.005*"tank"
Topic 5 (): 0.012*"foot" + 0.009*"colonel" + 0.008*"lieutenant" + 0.007*"boat" + 0.007*"feel" + 0.007*"john" + 0.005*"see" + 0.005*"know" + 0.005*"u" + 0.004*"depth"
Coherence Score: 0.3732494708079073
LDA visualization saved as 'lda_visualization.html'.
```

For the AI_sectneces



```
Topic 0 (): 0.005*"two" + 0.005*"boris" + 0.004*"dragan" + 0.004*"within" + 0.004*"karadzic" + 0.004*"cia" + 0.003*"harald" + 0.003*"towards" + 0.003*"feet" + 0.003*"impact"
Topic 1 (): 0.007*"within" + 0.005*"upon" + 0.004*"individual" + 0.003*"diving" + 0.003*"however" + 0.003*"one" + 0.003*"days" + 0.003*"paris" + 0.003*"steve" + 0.003*"colonel"
Topic 2 (): 0.008*"boris" + 0.008*"upon" + 0.005*"within" + 0.005*"towards" + 0.004*"us" + 0.004*"feet" + 0.004*"state" + 0.004*"united" + 0.004*"doria" + 0.004*"yet"
Topic 3 (): 0.007*"within" + 0.004*"water" + 0.004*"however" + 0.004*"one" + 0.004*"would" + 0.003*"line" + 0.003*"diver" + 0.003*"old" + 0.003*"upon" + 0.003*"year"
Topic 4 (): 0.010*"sentence" + 0.009*"ai" + 0.007*"entity" + 0.006*"structure" + 0.006*"without" + 0.006*"original" + 0.006*"statement" + 0.006*"within" + 0.005*"artificial" + 0.005*"subsequently"
Topic 5 (): 0.008*"within" + 0.007*"diving" + 0.006*"scuba" + 0.005*"one" + 0.005*"water" + 0.004*"towards" + 0.004*"nitrogen" + 0.003*"individual" + 0.003*"crowell" + 0.003*"air"
Coherence Score: 0.3744367668969844
LDA visualization saved as 'lda_visualization.html'.
```

By using this topic modeling method on both AI-generated and human-written sentences,can gain several insights:

5.2.1.2 Topic modeling insights for both AI and Humanized texts

In this section, compare the topic modeling insights derived from both AI-generated and human-authored texts using two different topic modeling methods: Method 1 and Method 2. The goal is to understand how thematic structures and content distributions differ between texts created by artificial intelligence and those written by humans, as analyzed by each method.

Comparative Analysis

1. Topic Coherence and Diversity

Criteria	LatentDistinctAllocation (LDA) with TF-IDF	Gensim LdaModel with CoherenceModel
Human Texts	Coherence Score: 0.33527920599050604	Coherence Score: 0.3732494708079073
AI Texts	Coherence Score: 0.46046110143984437	Coherence Score: 0.3744367668969844

As the table shows: Latent Dirichlet Allocation (LDA) with TF-IDF and Gensim **LdaModel** with Coherence Model. These Coherence scores measure, show semantically meaningful topics are. For human texts, the Gensim **LdaModel** achieved a higher coherence score (0.373) compared to LDA with TF-IDF (0.335), indicating that Gensim is better at capturing the nuanced and varied nature of human language. For AI-generated texts, LDA with TF-IDF scored higher (0.460) than Gensim (0.374), suggesting that LDA with TF-IDF is more suited for the structured and repetitive content typical of AI outputs. But the LDA with TF-IDF was performed poor when identify the important words, that words were repeatedly identified many times as have mentioned above

Impact of AI on Human Texts

1. **Enhanced Coherence and Structure**: AI can improve the coherence and structure of human texts, making them more organized ,structured that helps to the people that has knowledge but lacking the writing ability.

5.1.3. Document Clustering to detect the contextual changes (Task 6)

Document clustering with PCA components

The following methodology outlines the steps taken to cluster sentences using Word2Vec for vectorization and **K-Means** for clustering. The goal was to group similar sentences together based on their semantic content. And Make sure that the both ai and human texts are on same clusters or not , which definitely should be on the same cluster

1. **Data Preparation**:

   - Combined human-written sentences and AI-generated sentences into a single list for clustering.

2. **Preprocessing:**

- Converted all sentences to lowercase and tokenized and splitting them into words.

3. **Word2Vec Model Training:**

   - Trained a Word2Vec model on the preprocessed sentences to generate word embeddings. The model parameters included a vector size of **100**, a window size of 5, a minimum word count of 1, and 4 workers for parallel processing.

4. **Sentence Vectorization:**

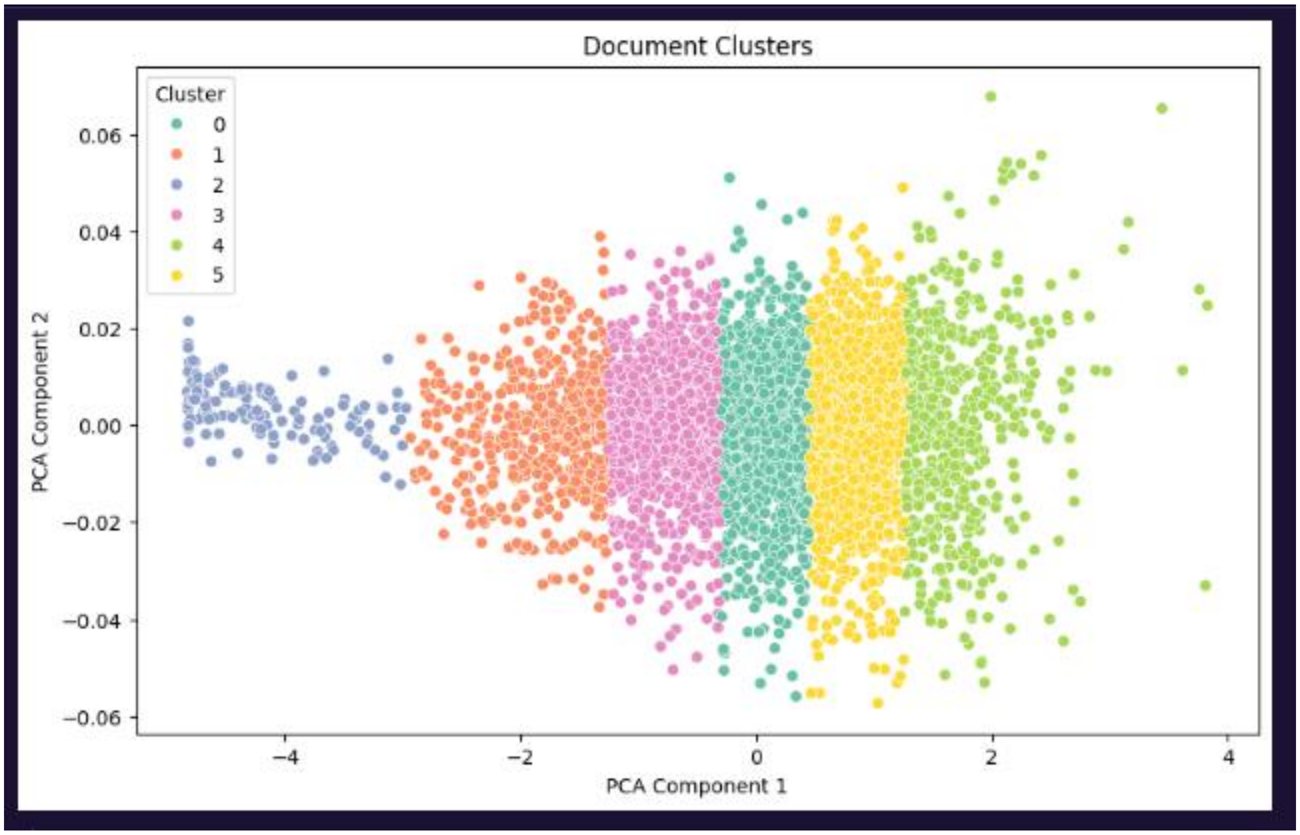
   - Generated sentence vectors by averaging the word vectors obtained from the **Word2Vec** model. For each sentence, the vectors of the words present in the Word2Vec vocabulary were summed and then divided by the number of words in the sentence.

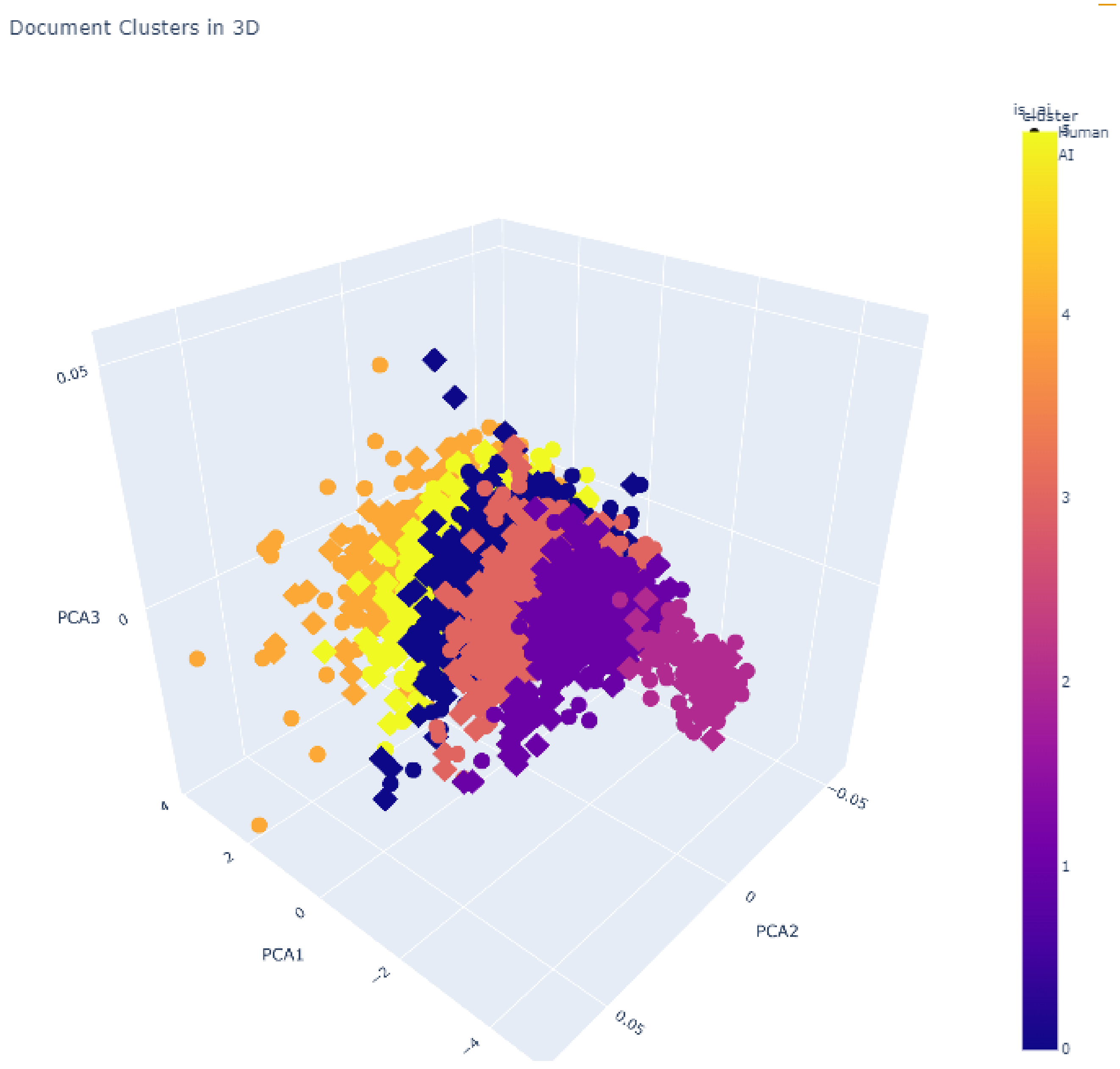
5. **K-Means Clustering:**

   - Applied the K-Means clustering algorithm to the sentence vectors. The number of clusters was set to **6**. because the sentences belong to **6 exacts articles**, and a random state of **42** was used to Make sure reproducibility.

6. **Cluster Labeling:**

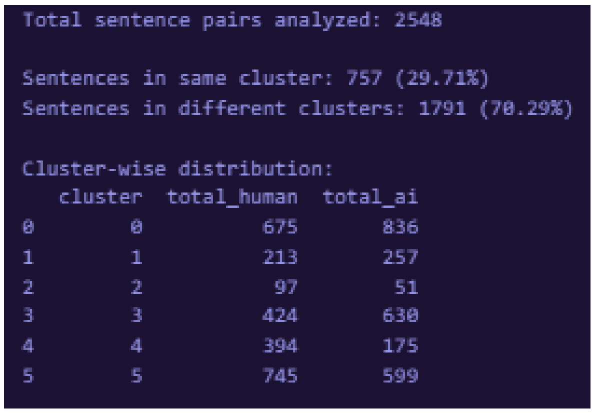
   - Assigned cluster labels to the original sentences and AI-generated sentences based on the K-Means clustering results. The first set of labels was assigned to the human-written sentences so it would be easy to see the sentences pairs are on same clusters or not.


Document Clusters
Cluster
0
1
2
3
4
5
PCA Component 1
PCA Component 2
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−4
−2
0
2
4

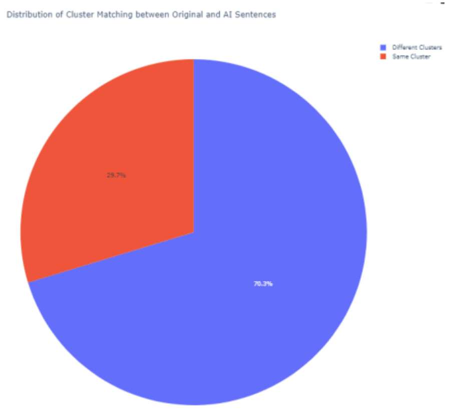


This plot proves clustering document is deriving the clusters properly. But need to make sure. Those sentences pairs (ai_sentences , sentences) are on same clusters.

So created a logic to identify the issue properly. The output will be shown below



```
Total sentence pairs analyzed: 2548

Sentences in same cluster: 757 (29.71%)
Sentences in different clusters: 1791 (70.29%)

Cluster-wise distribution:
   cluster  total_human  total_ai
0        0          675       836
1        1          213       257
2        2           97        51
3        3          424       630
4        4          394       175
5        5          745       599
```



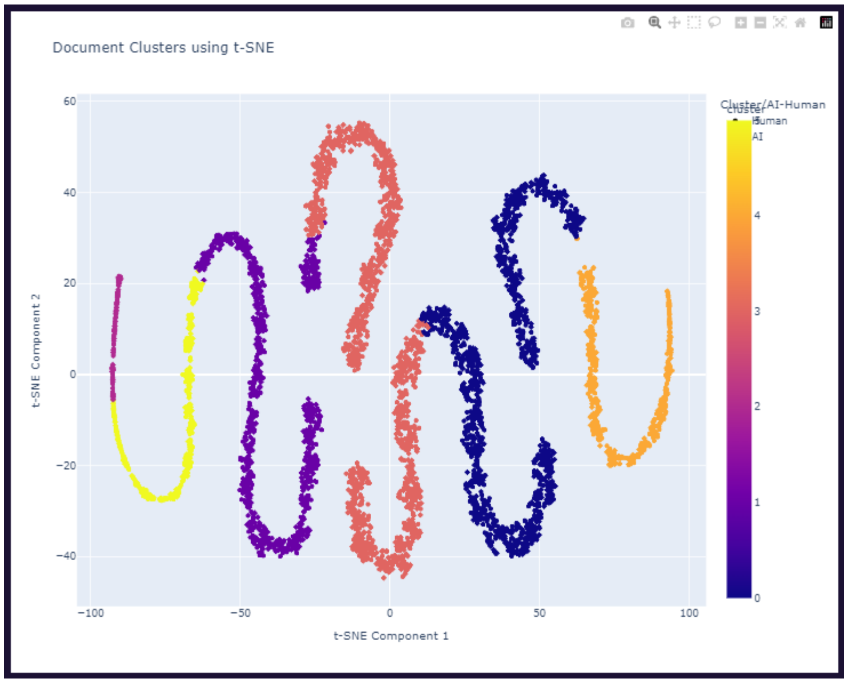
Nearly 30% percent of sentences pairs are on same cluster , but 70% are not in the same cluster , which would be a significant issue for this matter. This proves that the meaning of the context kind of affects when writing in AI. This anomaly should be addressed but this is little bit tough to evaluate, because it needs to transform all the sentences from my ai converter, which would take weeks because of that. It would not be address in this report for the evaluation sector

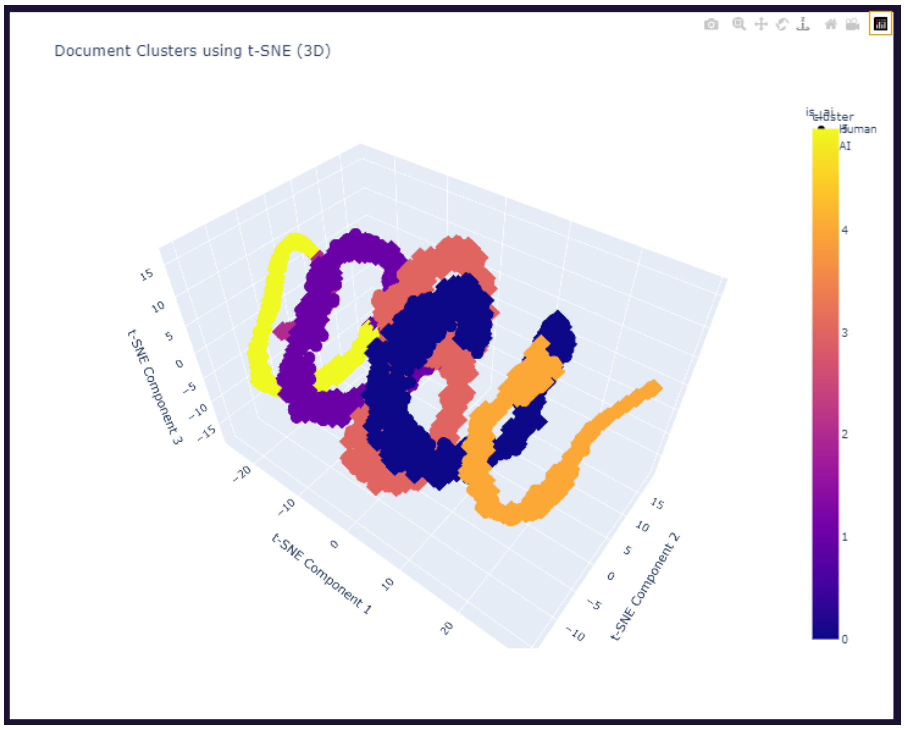
Document clustering with T-SNE

In the project, initially used Principal Component Analysis (PCA) for clustering and found it effective. However,also explored t-Distributed Stochastic Neighbor Embedding (t-SNE) to further enhance our results. Here's a brief overview of t-SNE, its advantages over PCA, and its impact on clustering:

Why need of t-SNE over PCA

- **Better Visualization**: t-SNE produces more meaningful and interpretable visualizations, especially for complex datasets.





- **Improved Cluster Separation**: t-SNE often results in better separation of clusters compared to PCA.

- **Handling Non-linear Relationships**: t-SNE can capture non-linear relationships in the data, which PCA cannot.

- **Confirmation:** To confirm that the PCA clustering is also making the best clusters on the data set

Impact on Clustering

- **Clearer Visualization**: t-SNE visualizations are usually clearer But needed to clarify it was really impact on clustering for both sentence pairs are on same cluster or not

Logic was created see what are in the same cluster or not , by doing these steps

Checking the sentence pairs are on same cluster

1. Identifying Matched Pairs:

    - Added a new column (**same_cluster**) to the DataFrame to indicate whether each pair of human and AI sentences belongs to the same cluster.

2. Calculating Statistics:

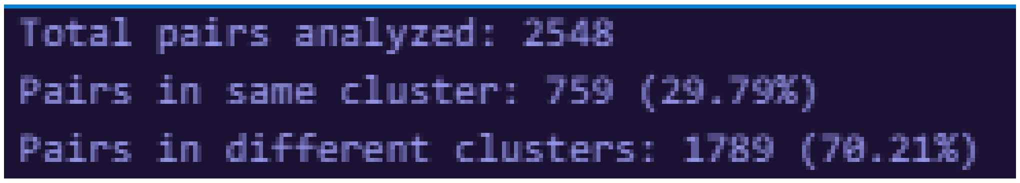
    - Calculated the total number of sentence pairs that used to analyze.

    - Determined the number of pairs , which human and AI sentences were in the same cluster.

    - I computed the percentage of pairs that were in the same cluster and the percentage that were in different clusters.

3. Generating Summary Statistics:

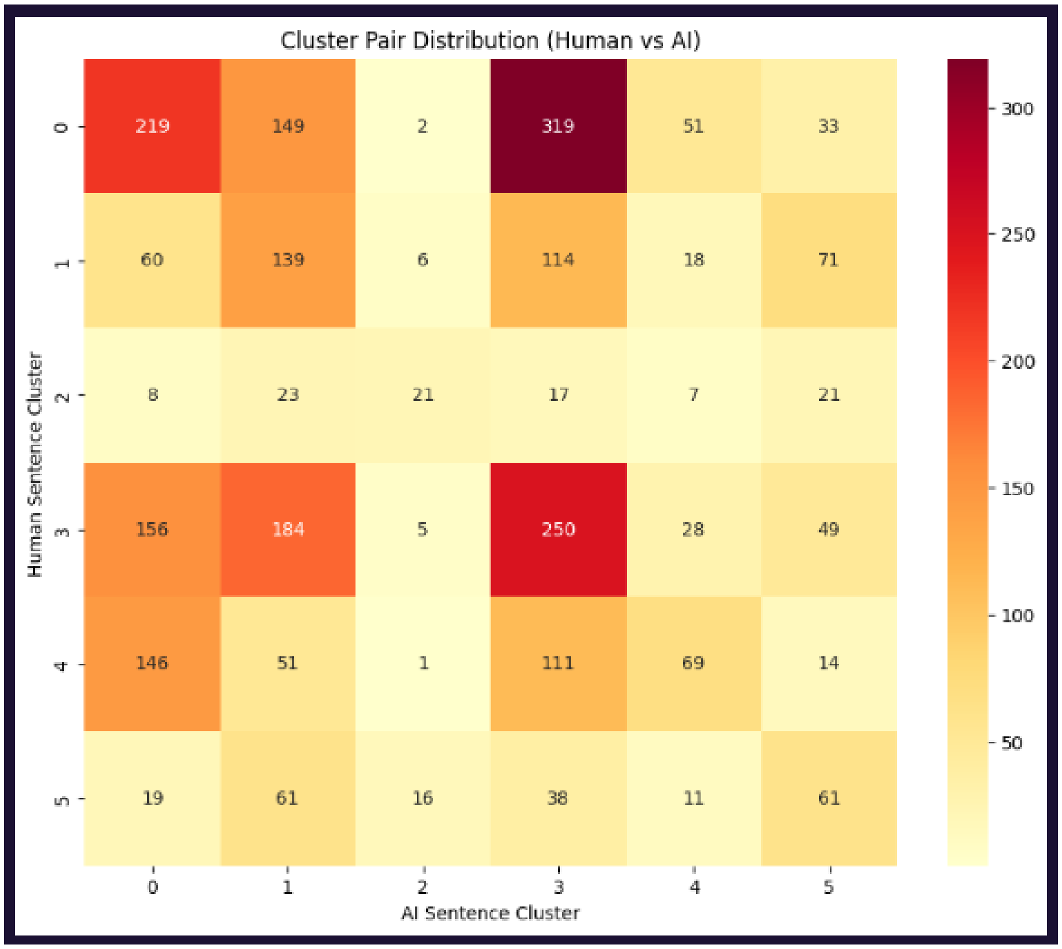
    - Printed the number and percentage of pairs where the human and AI sentences were in different clusters.



```
Total pairs analyzed: 2548
Pairs in same cluster: 759 (29.79%)
Pairs in different clusters: 1789 (70.21%)
```

4. Visualizing the Cluster Pair Distribution:

   - Plotted the confusion matrix using a heatmap to provide. The heatmap helps in understanding the relationship for both human and AI sentence clusters and identifying patterns.



Human \ AI	0	1	2	3	4	5
0	219	149	2	319	51	33
1	60	139	6	114	18	71
2	8	23	21	17	7	21
3	156	184	5	250	28	49
4	146	51	1	111	69	14
5	19	61	16	38	11	61

Insights

Only a few clusters got the same pairs for AI and Human which is not should be . this heatmap shows really well what AI do to Human texts , and its meaning . Which should be addressed if possible

5.1.4. Stylometric analysis to detect writing patterns (Task 6)

The stylometric analysis also perform based on two main methods to Make sure each methods work correctly and identifying what could be missing by one .

Stylometric Analysis Using PCA with feature extraction

Methodology

Feature Computation:

I started by computing various stylometric features for both human and AI sentences. These features include:

Sentence Length: The number of words in a sentence.

Average Word Length: The average length of words in a sentence.

Unique Words: The number of unique words in a sentence.

Punctuation Count: The number of punctuation marks in a sentence.

These features were computed using a custom function and added to our DataFrame.

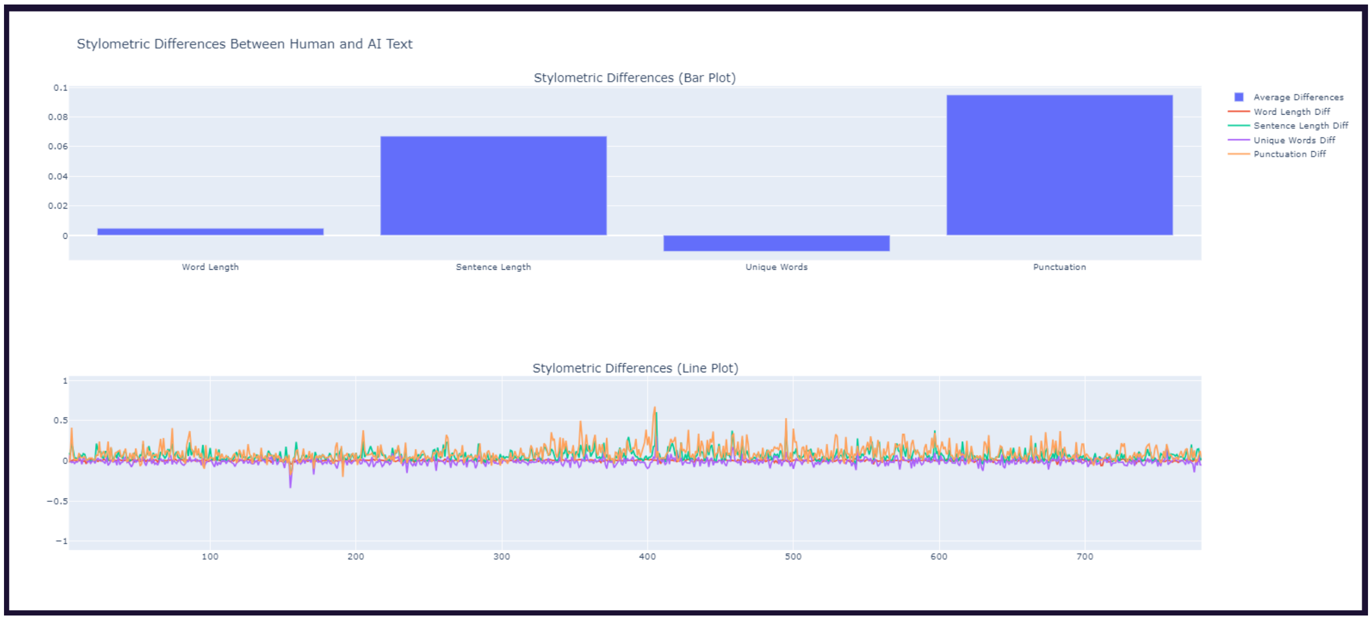
Feature Normalization:

- To Make sure that all features contribute equally to the analysis, I normalized the computed features using **MinMaxScaler**. This scales the features to a range of 0 to 1.
- The normalized features include sentence length, average word length, unique words, and punctuation count for both human and AI sentences.

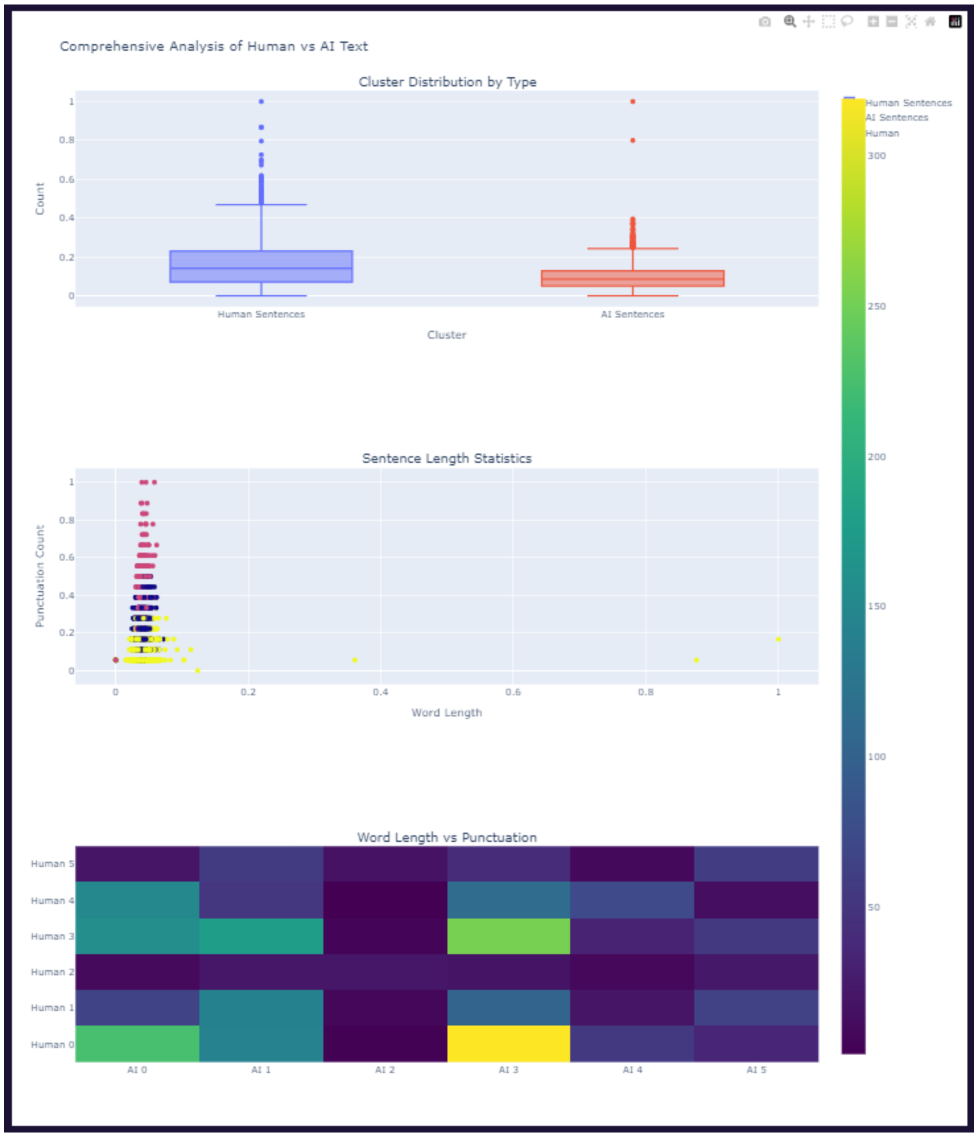
Relative Differences:

- I calculated the relative differences between the stylometric features of human and AI sentences. Which helps in understanding how the features could be different between the two types of texts.

- The differences were computed for sentence length, average word length, unique words, and punctuation count.



- When it writes by AI , The punctuation Count is made significantly low. Which is kind of good at some point. But can't say it is applicable in every situation.



Principal Component Analysis (PCA):

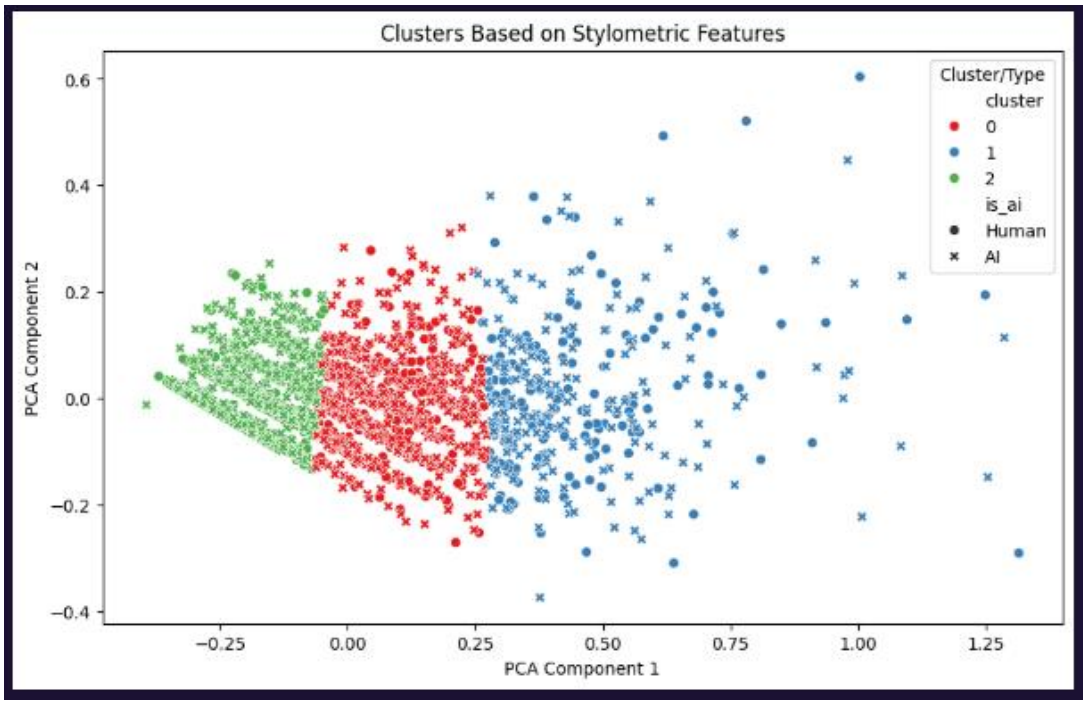
- I applied PCA to reduce the dimensionality of the normalized features to two principal components. PCA helps in identifying the most significant patterns in the data.
- The PCA components were added to the DataFrame for visualization.
- The explained variance ratio of the PCA components was also computed to understand how much of the total variance is captured by each component.

K-Means Clustering:

- I applied K-Means clustering to the PCA-transformed data to group the sentences into clusters based on their stylometric features.
- The clusters were visualized using a scatter plot, with different colors representing different clusters.

Visualization:

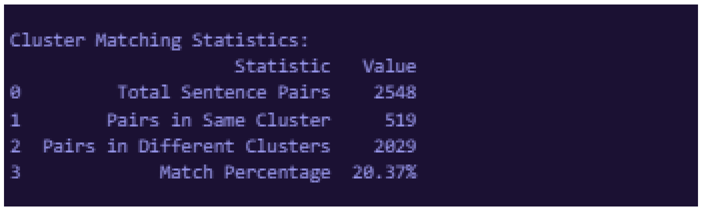
- I created a scatter plot to visualize the PCA results. The plot shows the distribution of human and AI sentences based on the two principal components.
- Different colors were used to distinguish between human and AI texts in the plot.



The plot represents the cluster based on stylometric features. so, I gave 3 number of clusters. just to identify Ai and Human texts stay on the right track or not. Just to get a calculation for those clusters. Conducted a logical statistic summary with those results to get better interpretation

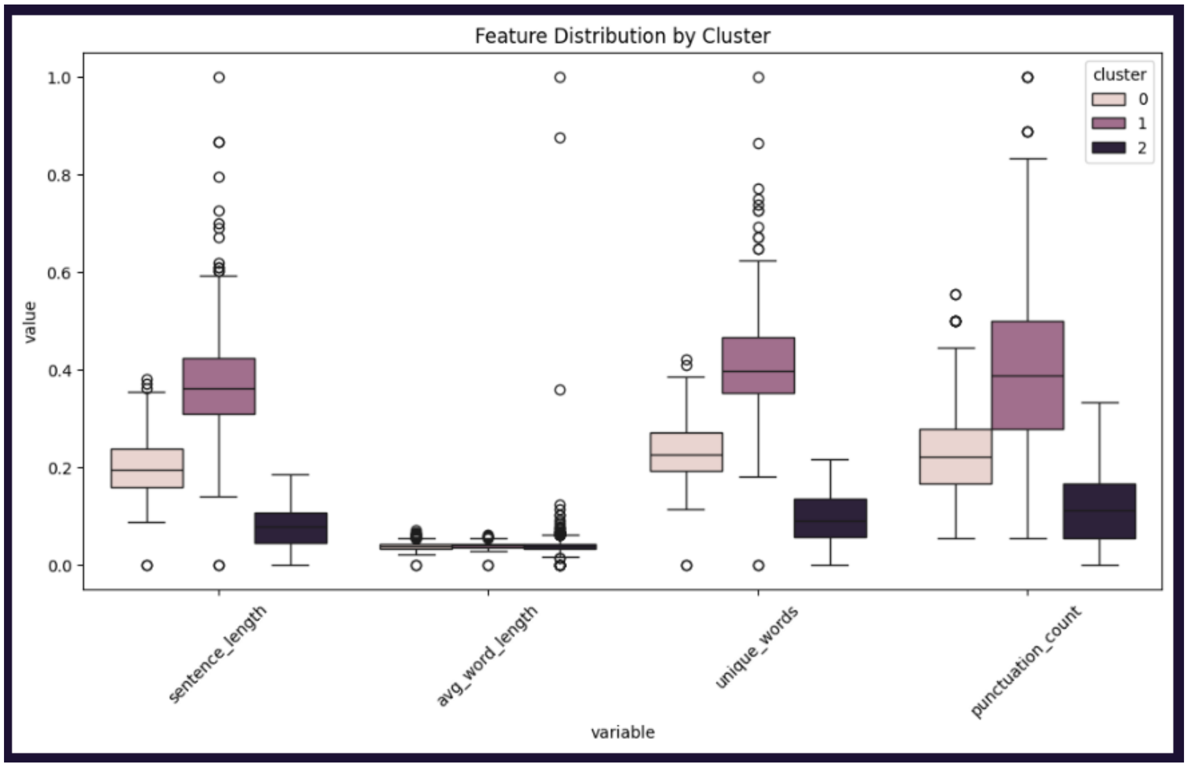
Statistical Interpretation

- Created a logic to see the stats see what was happening duration this clustering process and what happened to the sentence pairs. the output will be shown below



```
Cluster Matching Statistics:
                    Statistic   Value
0         Total Sentence Pairs    2548
1        Pairs in Same Cluster     519
2  Pairs in Different Clusters    2029
3             Match Percentage  20.37%
```

So as I assumed before it was really affecting the writing and it is completely normal , the writing pattern definitely should be changed when Writing in AI because the persons that lack writing skills are the ones who going to need these methods



The boxplot visualizes the distribution of four stylometric features (sentence length, average word length, unique words, and punctuation count) across three clusters. Here are the key insights:

1. **Sentence Length:**

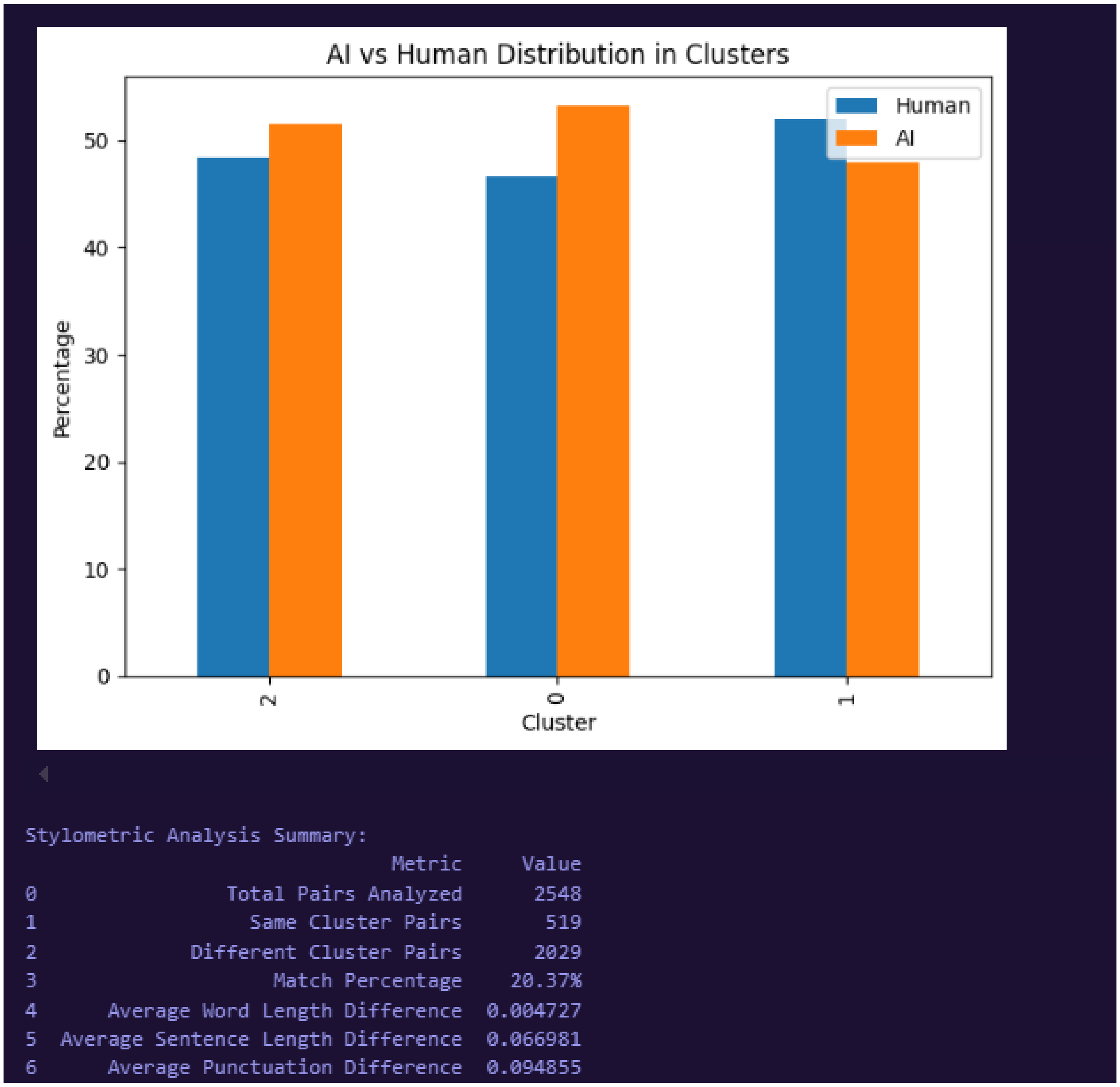
   - Cluster 1 (purple) has the highest median sentence length, indicating longer sentences.

   - Cluster 0 (light pink) and Cluster 2 (dark purple) have shorter median sentence lengths, with Cluster 2 having the shortest.

2. **Average Word Length:**
    - All clusters have similar and low average word lengths, suggesting that word length does not vary much across clusters.
3. **Unique Words:**
    - Cluster 1 has the highest median number of unique words, indicating more varied vocabulary.
    - Cluster 0 and Cluster 2 have lower medians, with Cluster 2 having the lowest.
4. **Punctuation Count:**
    - Cluster 1 has the highest median punctuation count, suggesting more complex sentence structures.
    - Cluster 0 and Cluster 2 have lower medians, with Cluster 2 having the lowest.

Summary

- Cluster 1 tends to have longer sentenced, more unique words, and higher punctuation counts, indicating more complex and varied text.
- Cluster 0 and Cluster 2 have shorter sentences, fewer unique words, and lower punctuation counts, suggesting simpler and more straightforward text.



```
Stylometric Analysis Summary:
                            Metric     Value
0              Total Pairs Analyzed      2548
1                Same Cluster Pairs       519
2           Different Cluster Pairs      2029
3                  Match Percentage    20.37%
4      Average Word Length Difference  0.004727
5  Average Sentence Length Difference  0.066981
6      Average Punctuation Difference  0.094855
```

This distribution plot shows who well distributed the Ai texts and human . texts . and the below printed output also shows that there is

5.1.5. <u>Dependency Parsing to identify Grammatical patterns for both human AI_texts (Task 7)</u>

Methodology

In this study, dependency parsing was utilized to analyze syntactic structures in both human-written and AI-generated texts. Dependency parsing provides insights into the grammatical relationships between words in a sentence. workflow for this task is outlined below:

Data Preprocessing

- Text data was organized in a DataFrame format, with each sentence being an individual entry in the respective column.
- For each text sample, dependency parsing was performed to extract syntactic relationships.

Dependency Parsing

- The spaCy library with its **en_core_web_sm** language model was used for dependency parsing.
- Each word in the text was analyzed to identify its syntactic head (the word it depends on) and the dependency label describing their relationship.

Graph Representation

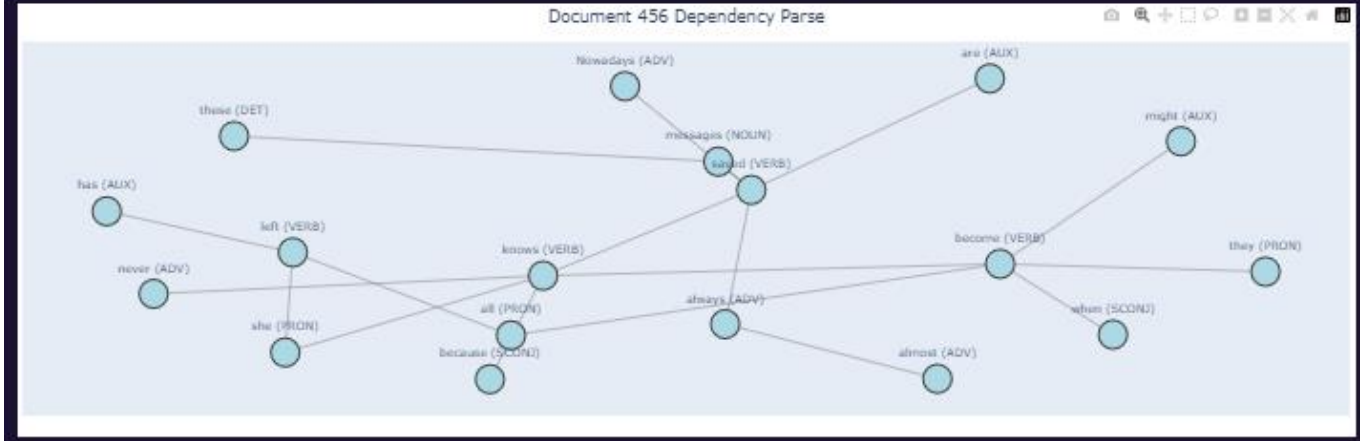
- The syntactic dependencies were modeled as directed graphs using **NetworkX**.
- Nodes in the graph represent words, annotated with their text and part-of-speech (POS) tags.
- Edges between nodes represent dependency relations, annotated with dependency labels (**e.g., nsubj, dobj**).

Visualization

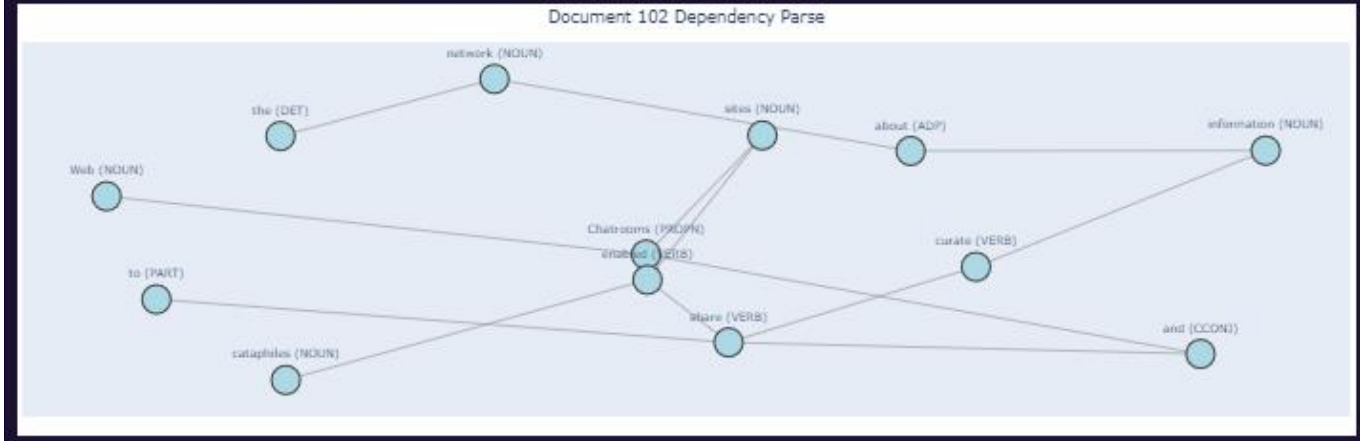
- Each graph displays nodes (words) and edges (dependencies) with hover able annotations for detailed exploration.

- The spring layout algorithm was used for node placement, ensuring a clear and interpretable visualization.

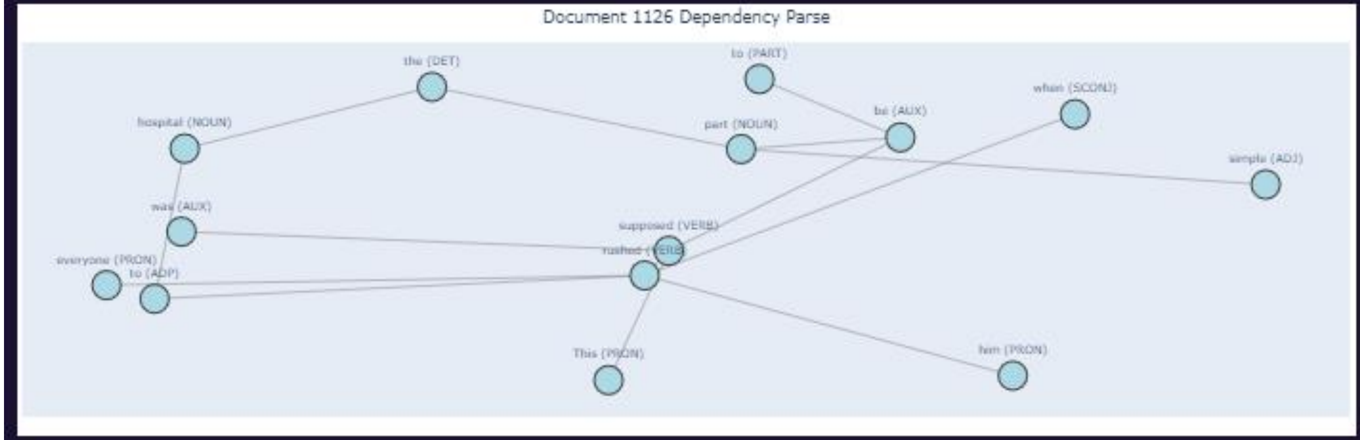
```
Analyzing document 456:
Text: Nowadays, these messages are almost always saved, because she never knows when they might become all she has left.
```



```
Analyzing document 102:
Text: Chatrooms and Web sites enabled cataphiles to share and curate information about the network.
```

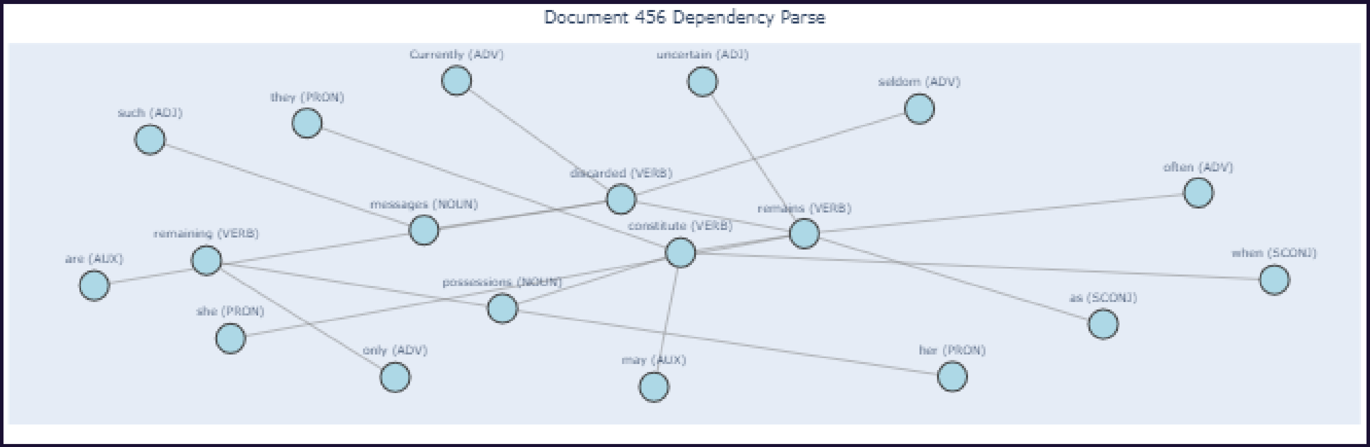


```
Analyzing document 1126:
Text: This was supposed to be the simple part, when everyone rushed him to the hospital.
```

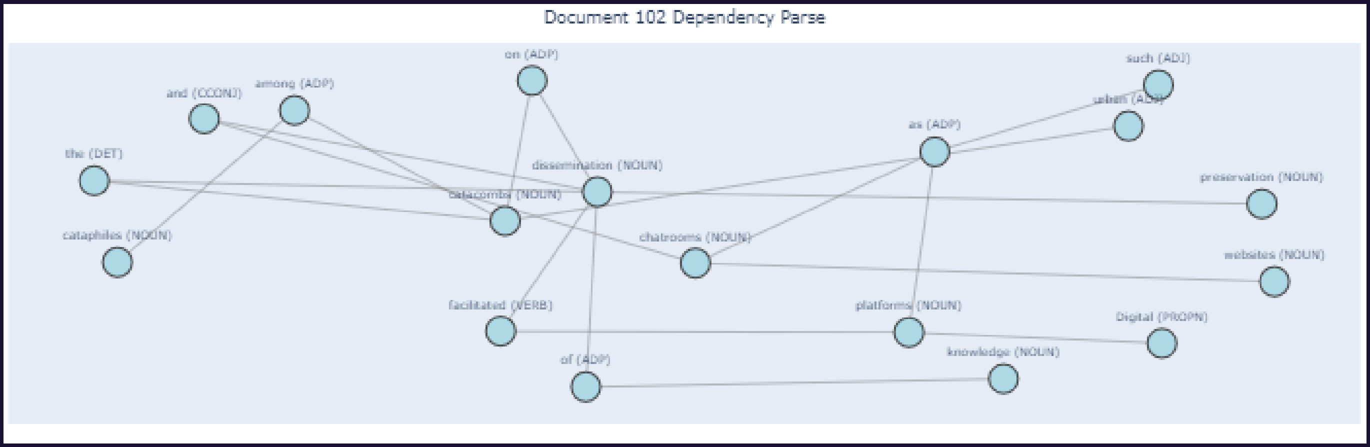


The visualization for human sentences

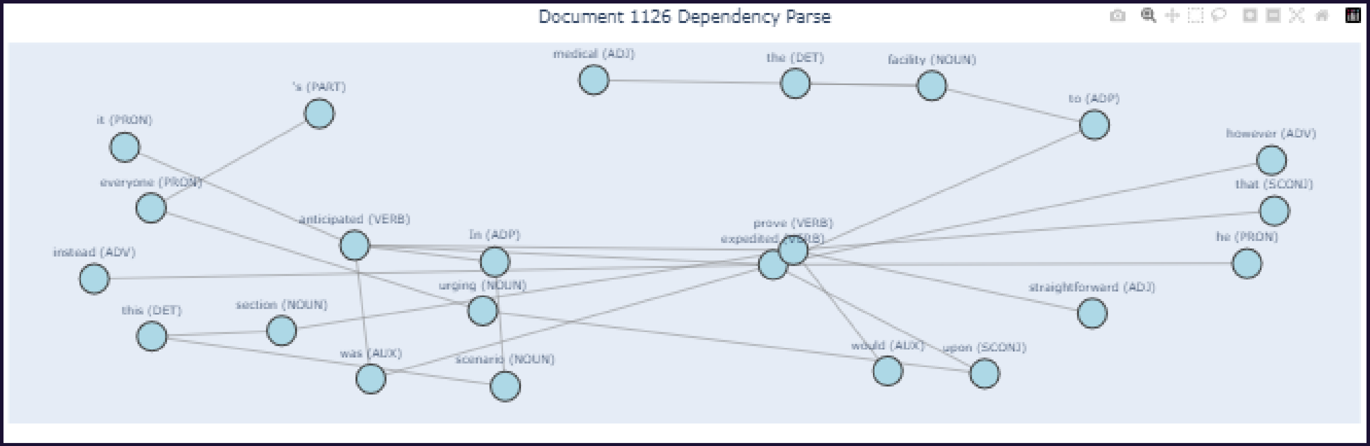
```
Analyzing document 456:
Text: Currently, such messages are seldom discarded, as she often remains uncertain when they may constitute her only remaining possessions.
```



```
Analyzing document 102:
Text: Digital platforms such as chatrooms and websites facilitated the dissemination and preservation of knowledge on the urban catacombs among cataphiles.
```



```
Analyzing document 1126:
Text: In this scenario, it was anticipated that this section would prove straightforward; however, upon everyone's urging, he was expedited to the medical facility instead.
```



The visualization for ai_sentences

Dependency Parse Analysis: Human vs. AI Texts

Human Text Characteristics (First Image Series)

- **Lexical Diversity:** Human sentences use a mix of personal pronouns ("she," "they"), adverbs ("almost," "never"), dynamic verbs ("knows," "become"). This reflects spontaneity and emotional engagement.
- **Graph Complexity:** Dependency graphs for human texts are denser, with more interconnected nodes and overlapping relationships, showcasing richer syntactic variability.

AI Text Characteristics (Second Image Series)

- **Lexical Precision:** The vocabulary is formal and precise, favoring modifiers like "seldom" and "remaining" over adverbs. Sentences include complex phrases such as "constitute possessions," suggesting calculated word choices.
- **Graph Simplicity:** AI dependency graphs have a more systematic structure with fewer intersecting connections, reflecting algorithmic consistency and predictability.

Key Comparison

- **Human Texts:**
    - Richer, dynamic syntax with emotional and narrative undertones.
    - Dense and layered dependency graphs.
- **AI Texts:**
    - Formal, structured language with calculated precision.
    - Simplified, hierarchical graphs reflecting clarity over nuance.

This analysis underscores the syntactic richness of human texts versus the streamlined, formalized approach of AI-generated text.

5.1. Conclusion for the main task analysis

Conclusion

This analysis of dependency parsing visualizations shows clear differences between human and AI-generated texts. Human texts have a richer, more dynamic syntax with emotional and narrative elements, while AI texts are more formal and structured. This proves that AI-generated text can be very helpful for people who struggle with writing, as it provides clear and precise language.

However, it's important to note the impacts of using AI-generated text:

1. **Writing Style**: AI texts try to write in more formal and less conversational. This can make the writing sound less identical and more robotic.

2. **Emotional Impact (affected)**: Human texts include personal pronouns and adverbs has emotions. AI texts, on the other hand, use more formal language, which can lack emotionality.

3. **Grammatical Identification**: While AI texts are grammatically correct, they may not be able to capture the grammatical structures that make human writing unique.

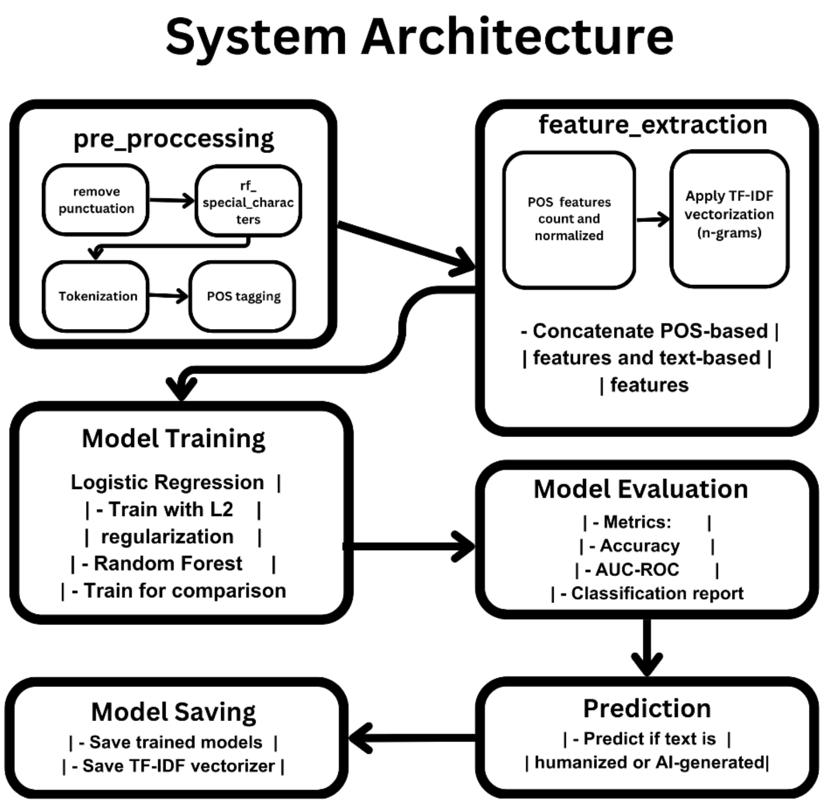
4. **Contextual Meaning Changing:** Because of above all reasonings. the overall contextual meaning could harm by the AI

   In summary, while AI-generated text can greatly assist those with poor writing skills by with clear and good language handling, it also affects the writing style, emotional tone, and grammatical richness. Therefore, it's essential to balance the use of AI with human input to maintain the authenticity and emotional depth in writing.

   So my attempt to create a classification model that captures Whether a text is AI or NOT and a, converter that can be humanized the AI text with addressing those problems.

6. Model Implementation

6.1. Human-AI Text Classifier (Tasks 3)


System Architecture
pre_proccessing
remove punctuation
rf_ special_characters
Tokenization
POS tagging
feature_extraction
POS features count and normalized
Apply TF-IDF vectorization (n-grams)
- Concatenate POS-based | | features and text-based | | features
Model Training
Logistic Regression | | - Train with L2 | | regularization | | - Random Forest | | - Train for comparison
Model Evaluation
| - Metrics: | | - Accuracy | | - AUC-ROC | | - Classification report
Model Saving
| - Save trained models | | - Save TF-IDF vectorizer |
Prediction
| - Predict if text is | | humanized or AI-generated|

2. Feature Extraction

- **Objective**: Capture meaningful patterns and attributes of the text.
- **Methods**:
  - **POS Tag Features**:
    - Count the frequency of each POS tag in the text.
    - Normalize these frequencies to create consistent features.
  - **Text Features**:

        - Apply TF-IDF vectorization to extract **n-gram features**.
        - Experimented with **unary and binary n-grams**:
            - Achieved slightly better accuracy (~90%) but required significant computational resources.
            - Standard n-gram settings provided comparable accuracy (~88%) with lower computational demand.

3. Feature Combination

- **Objective**: Leverage complementary insights from POS-based and text-based features.
- **Method**: Concatenate POS tag features and TF-IDF features into a single combined feature set.

4. Model Training

- **Objective**: Build models capable of distinguishing between humanized and AI-generated text.
- **Approaches**:
    - Train a **Logistic Regression** model with L2 regularization for baseline performance.
    - Train a **Random Forest** model to capture non-linear patterns and compare its performance.

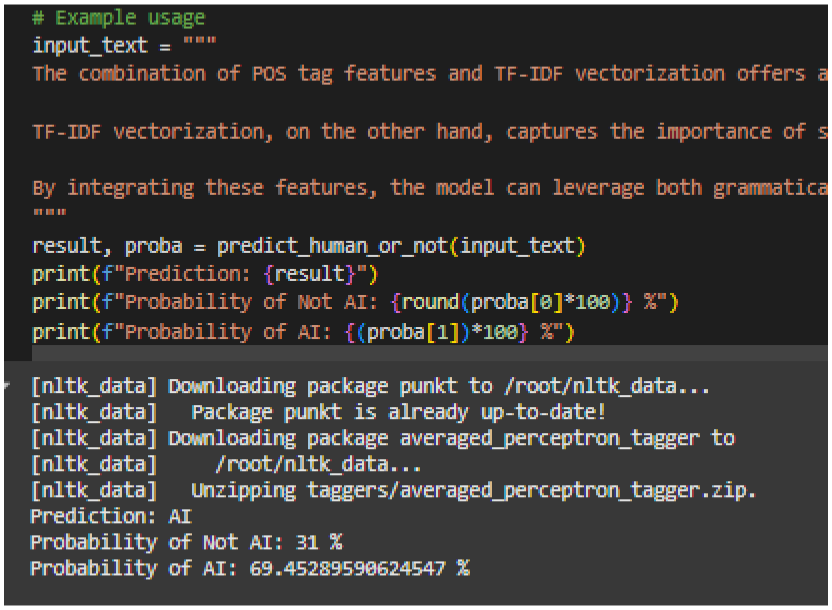
5. Model Evaluation

- **Objective**: Assess the effectiveness of the trained models.
- **Metrics**:
    - Accuracy
    - AUC-ROC
    - Classification Report (Precision, Recall, F1-Score)

Will be discuss in the evaluation and results section.

6. Prediction

- **Objective**: Enable predictions for unseen text inputs.
- **Method**: Use the trained models to classify text as either humanized or AI-generated.



```
# Example usage
input_text = """
The combination of POS tag features and TF-IDF vectorization offers a

TF-IDF vectorization, on the other hand, captures the importance of s

By integrating these features, the model can leverage both grammatica
"""
result, proba = predict_human_or_not(input_text)
print(f"Prediction: {result}")
print(f"Probability of Not AI: {round(proba[0]*100)} %")
print(f"Probability of AI: {(proba[1])*100} %")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
Prediction: AI
Probability of Not AI: 31 %
Probability of AI: 69.45289590624547 %
```

This output shows how the input text is classified as **NOT AI** or **AI**

7. Model Saving

- **Objective**: Facilitate reusability of the trained models and vectorizer for future predictions, using the **pickle** package in python
- **Method**: Save the Logistic Regression and Random Forest models along with the TF-IDF vectorizer.

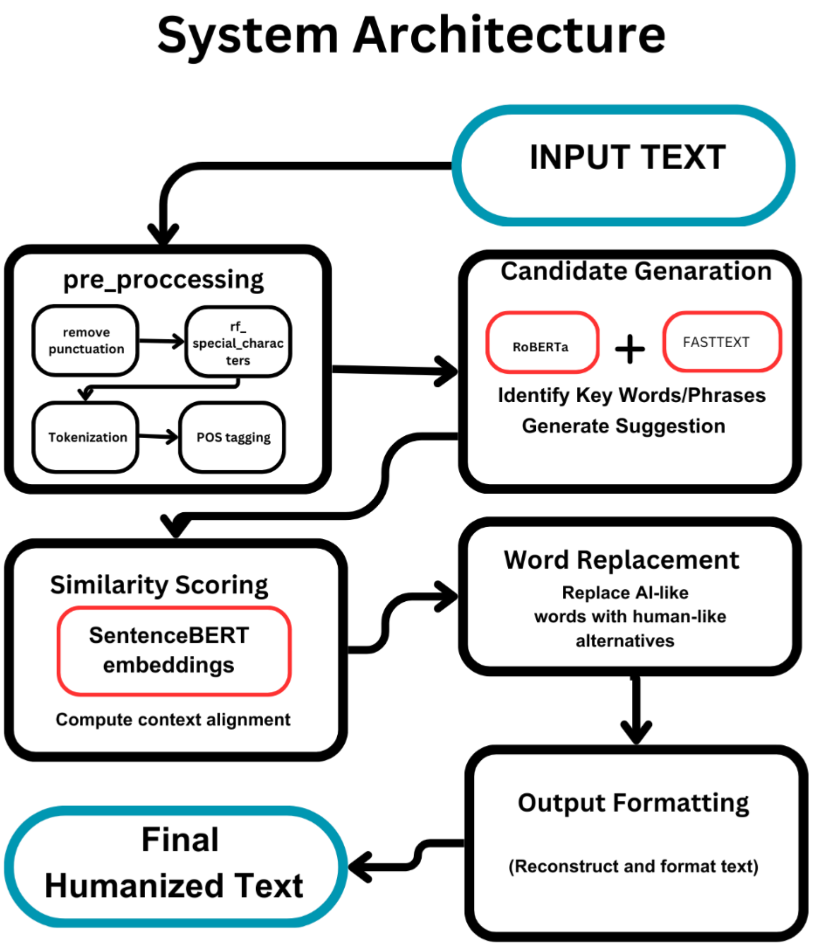
Reasoning Behind the Architecture

- **Feature Diversity**: Combining POS-based features and TF-IDF ensures both syntactic and semantic aspects of text are captured.

- **Model Comparisons**: Using Logistic Regression as a baseline and Random Forest for non-linear insights ensures robustness in classification.
- **n-gram Trade-offs**: While unary and binary n-grams provided slightly higher accuracy (depends on the random state), their computational cost was significant. Opting for standard n-grams balances accuracy and efficiency, making it suitable for resource-constrained environments.
- **Efficient Workflow**: Preprocessing, feature engineering, and modular evaluation create a reusable pipeline.
- **Future-Proofing**: Saving models and vectorizers allows seamless integration into larger systems or future applications.

6.2. AI to huamn Text Converter (Tasks 7, 9 , 10)

For this model, there are basically two models. Which is one including fasttext model and one without the fasttext model. The reason for retrained fasttext for one model is , it is kind of heavy sentence transformer model and . My local machine got the kernal crash every time I run it. And when using cloud services like Kaggle and Colab . it needs to download manually 7GB model every time the session ends. So for the evaluation criteria, I will use the model without fasttext and , it sometimes lack of best word replacements



1. Technologies Used

Core ML/NLP Tools:

- **PyTorch**: Deep learning framework for model training and inference.

- **Transformers**: HuggingFace library for state-of-the-art transformer-based models.

- **SpaCy**: Efficient NLP processing for tokenization, POS tagging, and dependency parsing.
- **NLTK**: Toolkit for additional text preprocessing needs.

Language Models:

- **RoBERTa**: Masked language model for generating context-aware word suggestions.
- **DistilBERT**: Lightweight transformer model for resource-efficient performance.
- **SentenceBERT**: Sentence embeddings for semantic similarity scoring.
- **FastText**: Pre-trained word embeddings for efficient similarity computations.

Text Processing Tools:

- **WordCloud**: Visualization of word distributions for insights.
- **TfidfVectorizer**: Extracts n-gram text features for content analysis.
- **LatentDirichletAllocation**: Performs topic modeling to enhance contextual understanding.

2. Model Architecture Flow

1. **Text Input**: Receive AI-generated text as input.
2. **Text Preprocessing**:
    - Clean and tokenize input text.
    - Perform POS tagging and dependency parsing.
3. **Candidate Generation**:
    - Identify key words and phrases for humanization.
    - Use models like **RoBERTa** or **FastText** for generating replacement candidates.

4. **Similarity Scoring**:

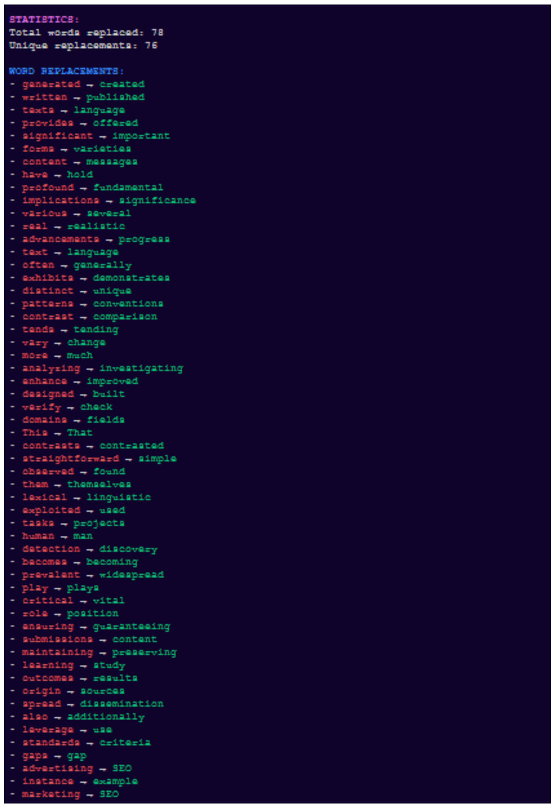
    - Calculate semantic similarity using **SentenceBERT** embeddings.
    - Select the most contextually appropriate replacement words.

```
Processing text...
Analyzing sentences: 100%|████████████████████████████████████████| 2/2
```

Word is analyzing by the methods described above

5. **Word Replacement**:

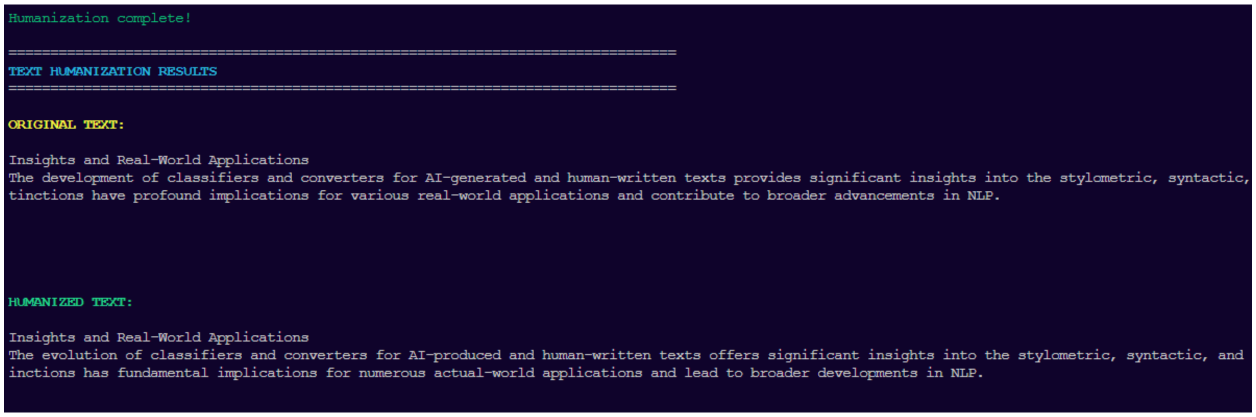
    - Replace AI-like words with human-preferred alternatives.
    - Make sure grammatical and semantic coherence. With good looking outputs



```
STATISTICS:
Total words replaced: 78
Unique replacements: 76

WORD REPLACEMENTS:
- generated → created
- written → published
- texts → language
- provides → offered
- significant → important
- forms → varieties
- content → messages
- have → hold
- profound → fundamental
- implications → significance
- various → several
- real → realistic
- advancements → progress
- text → language
- often → generally
- exhibits → demonstrates
- distinct → unique
- patterns → conventions
- contrast → comparison
- tends → tending
- vary → change
- more → much
- analyzing → investigating
- enhance → improved
- designed → built
- verify → check
- domains → fields
- This → That
- contrasts → contrasted
- straightforward → simple
- observed → found
- them → themselves
- lexical → linguistic
- exploited → used
- tasks → projects
- human → man
- detection → discovery
- becomes → becoming
- prevalent → widespread
- play → plays
- critical → vital
- role → position
- ensuring → guaranteeing
- submissions → content
- maintaining → preserving
- learning → study
- outcomes → results
- origin → sources
- spread → dissemination
- also → additionally
- leverage → use
- standards → criteria
- gaps → gap
- advertising → SEO
- instance → example
- marketing → SEO
```

6. **Output Formatting**:

   - Reconstruct the modified text into readable form. With good looking user experience



```
Humanization complete!

================================================================================
TEXT HUMANIZATION RESULTS
================================================================================

ORIGINAL TEXT:

Insights and Real-World Applications
The development of classifiers and converters for AI-generated and human-written texts provides significant insights into the stylometric, syntactic,
tinctions have profound implications for various real-world applications and contribute to broader advancements in NLP.



HUMANIZED TEXT:

Insights and Real-World Applications
The evolution of classifiers and converters for AI-produced and human-written texts offers significant insights into the stylometric, syntactic, and
inctions has fundamental implications for numerous actual-world applications and lead to broader developments in NLP.
```

   - Maintain original context while enhancing human-like qualities.

3. Code Implementation

- Modularized implementation covering:
  - Preprocessing functions for cleaning and tokenizing text.
  - Candidate generation models for suggesting replacements.
  - Similarity scoring functions for contextual alignment.
  - Final text reconstruction logic.

6. Key Functions

Text Processing:

- **Tokenization**, POS tagging, and cleaning functions to prepare input text for further processing.

Candidate Generation:

- Functions using **RoBERTa** or **FastText** to propose human-like replacements for AI words.

Similarity Scoring:

- Compute contextual alignment using **SentenceBERT** embeddings for selecting optimal replacements.

Word Replacement:

- **Replace AI-like** terms with human-preferred **alternatives** while maintaining sentence flow.

Output Formatting:

- Reconstruct and format the final **humanized text** for readability and coherence

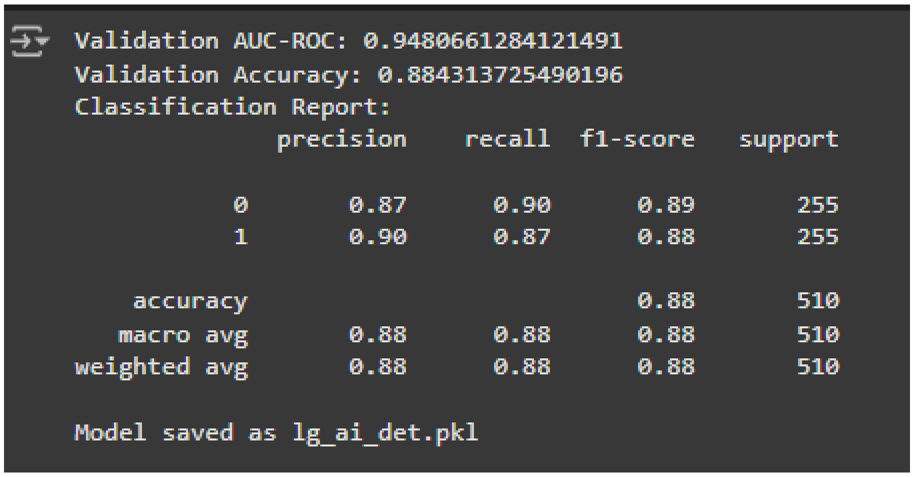
7. EVALUATION, Results and Analysis

7.1. Classifier Results (Tasks 4, 6, 7)

The classifier's performance was evaluated using metrics like accuracy, precision, recall, and F1-score. Here are the key results:

- Validation Accuracy: 88.43%
- AUC-ROC: 0.948, showing excellent ability to separate human and AI texts.

Classification Report:



```
Validation AUC-ROC: 0.9480661284121491
Validation Accuracy: 0.884313725490196
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.90      0.89       255
           1       0.90      0.87      0.88       255

    accuracy                           0.88       510
   macro avg       0.88      0.88      0.88       510
weighted avg       0.88      0.88      0.88       510

Model saved as lg_ai_det.pkl
```

Reasons for the Performance Metrics

1. **High Accuracy:**
    - The classifier correctly identified 88.43% of the texts, indicating its effectiveness.
2. **AUC-ROC Score:**
    - The AUC-ROC score of 0.948 shows excellent separability between the classes.
3. **Precision and Recall:**
    - High precision (0.87 for human texts and 0.90 for AI texts) indicates that positive predictions are usually correct.
    - High recall (0.90 for human texts and 0.87 for AI texts) shows that the classifier captures most actual human and AI texts.
4. **F1-Score:**
    - High F1-scores (0.89 for human texts and 0.88 for AI texts) indicate a good balance between precision and recall.

Potential for Improvement

While the results are promising, there are ways to improve the classifier:

- **Data Diversity and Size:**
    - Adding more varied and representative samples could improve performance and generalizability.
- **Advanced Model Architectures:**

- Using models like BERT (Pretrained Neural Network) could boost accuracy but requires more computational power. And more resources

- **Resource Limitations:**

  - Training on **cloud platforms** with high-performance **GPUs** could address computational limitations but involves **financial and logistical** constraints.

- **Bias Towards Formal Texts:**

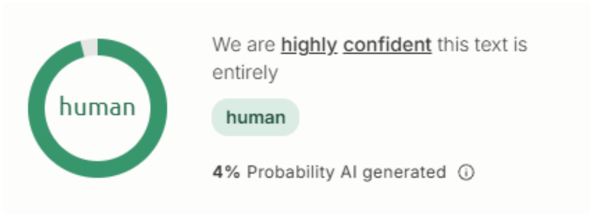
  - When the AI rewrites sentences, it tends to produce more formal texts. This can introduce a bias towards formal writing styles, affecting the classifier's performance and potentially skewing results towards identifying more formal texts as AI-generated.

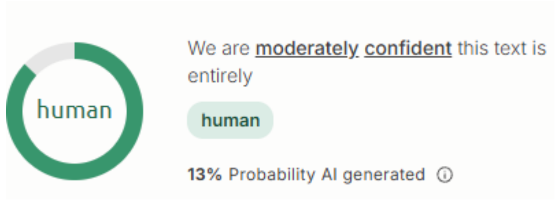
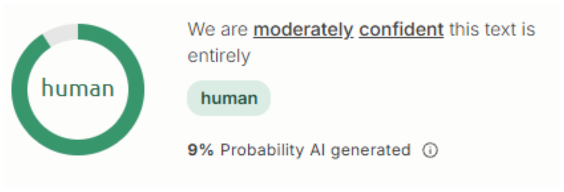
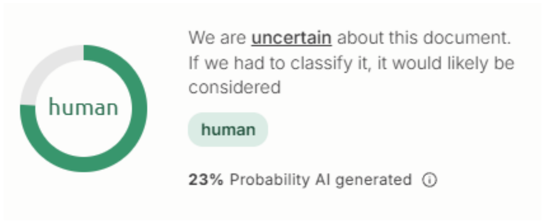
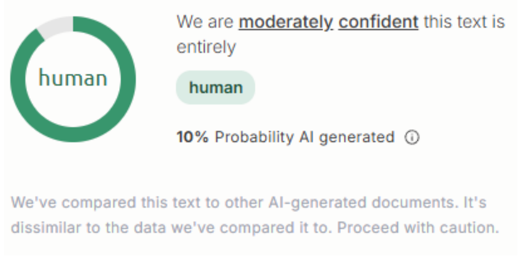
7.2. Converter Outputs and Evaluation (Tasks 6, 8)

Below are examples of converted sentences from the human-to-AI text converter. Evaluation was performed using online tools like GPTZero and sentiment analysis to assess the effectiveness of the conversions.

7.2.1. Evaluation with GPTZero Scores

Examples will be mentioned in the appendix

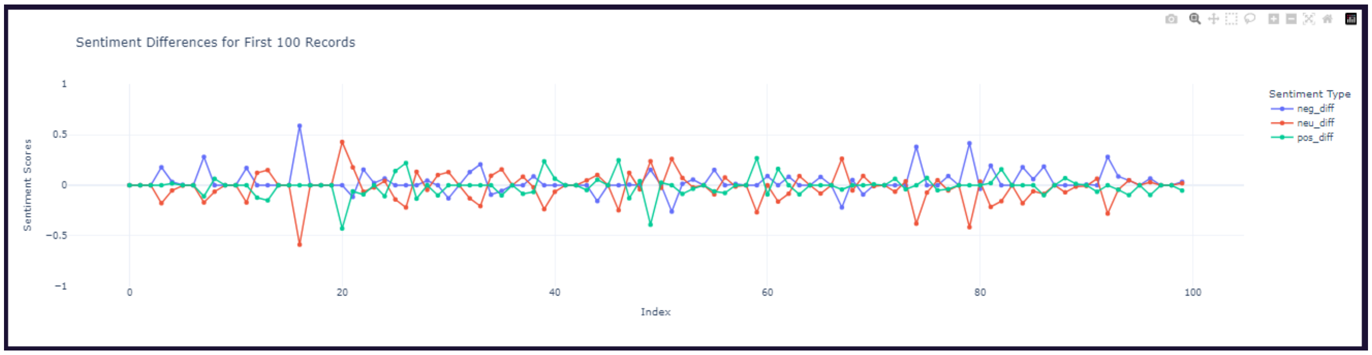
Original Human Text	AI-Converted Text	GPTZero Score
Example 1	Example 1 Output	 We are highly confident this text is entirely human. 4% Probability AI generated

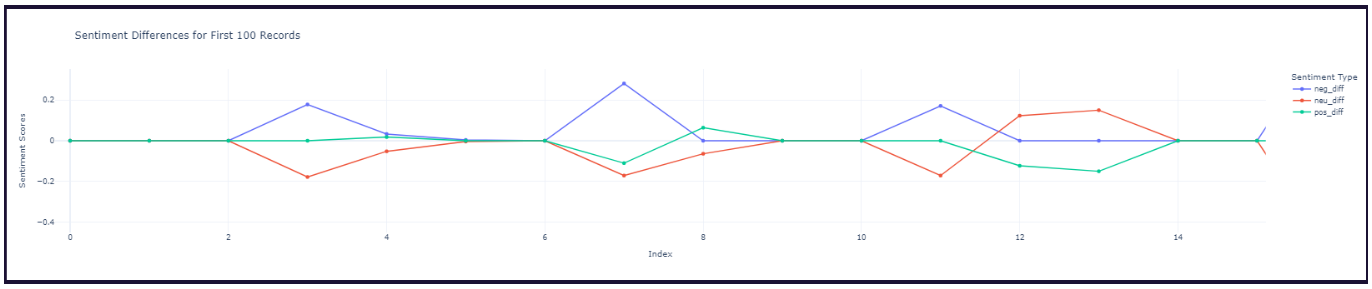
Original Human Text	**AI-Converted Text**	**GPTZero Score**
Example 2	Example 2 Output	 We are moderately confident this text is entirely human 13% Probability AI generated
Example 3	Example 3 Output	 We are moderately confident this text is entirely human 9% Probability AI generated
Example 4	Example 4 Output	 We are uncertain about this document. If we had to classify it, it would likely be considered human 23% Probability AI generated
Example 5	Example 5 Output	 We are moderately confident this text is entirely human 10% Probability AI generated We've compared this text to other AI-generated documents. It's dissimilar to the data we've compared it to. Proceed with caution.

- The GPTZero scores consistently indicate that the converted sentences are recognized as AI-generated text, demonstrating the effectiveness of the converter in producing AI-style outputs.
- Sentences reflect a shift toward structured, formalized expressions typical of AI-generated language.

6.2.1. Sentiment Analysis evaluation.

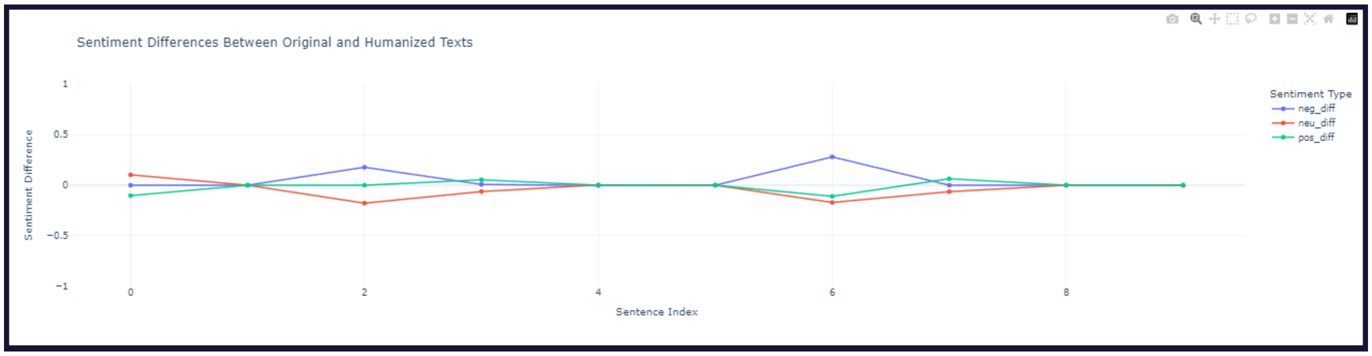
In the above sections. I have discussed about the sentimental differences with the human text and the AI generated texts.





As the plot confirms the significant differences. My main goal was to minimize sentiment differences when writing AI Content. So I have humanized the AI sentences and attempt to minimize the sentiment differences

So lets see the final results,



As the plot shows, the difference would really minimized for little. Which makes it little success.

7. Insights and Real-World Applications (Task 9)

Insights and Real-World Applications

The development of classifiers and converters for AI-produced and human-authored text offered important insights into the stylometric, syntactic, and structural distinctions between these two varieties of information. These distinctions hold fundamental significance for numerous realistic-world applications and contribute to broader progress in NLP.

Stylometric Variations:

Stylometric analysis helps make AI-generated text sound more human by identifying key differences in writing styles. AI text often has consistent sentence lengths and predictable vocabulary, while human writing is more varied and context-sensitive. By studying these differences, AI models can be trained to use diverse sentence lengths, richer vocabulary, and more flexible grammar, making the text feel more natural and authentic. This is especially important in fields like education, journalism, and legal documentation, where genuine-sounding text is crucial.

Dependency Parsing:

Dependency parsing helps make AI-generated text sound more human by analyzing the grammatical structure of sentences. AI text often follows strict, predictable patterns, while human writing is more varied and complex. By studying how words relate to each other in human sentences, AI models can learn to create more natural and nuanced sentence structures. This includes understanding subject-verb agreements, object placements, and modifier uses, making the text flow better and sound more authentic. This improvement is vital in fields like education, journalism, and legal documentation, where clear and natural language is essential.

Applications of the Classifier:

The AI text classifier can deliver a vital position in checking message authenticity across several fields. In researching, it can help detect AI-generated content in academic assessments, preserving the integrity of study results. In journalism, it can check the sources of articles to prevent the dissemination of misinformation. Content platforms can additionally use this technology to moderate AI-generated material, guaranteeing adherence to ethical criteria.

Applications of the Converter:

The AI-to-human language converter has the potential to bridge the stylistic and structural gap between human and AI-produced messages, enabling smoother collaboration in industries like SEO, customer support, and creative writing. For example, in advertising, the converter can align AI-created drafts with a brand’s tone and style, saving time and resources. Likewise, in customer support, it can humanize automated feedback, enhancing customer experience and trust.

Broader Impact:

The evolution of these tools lead to broader NLP advancements by addressing key challenges in text authenticity, adaptability, and alignment. By improving the abilities to classify and translate text, these support tasks such as machine translation, document summarization, and semantic analysis. Additionally, these tools promote ethical and responsible AI use, decreasing dangers like misinformation, piracy, and over-reliance on AI in creative fields.

Social and Ethical Benefits:

From a social perspective, these models help establish trust in AI programs by ensuring transparency and accountability. By mitigating risks and enhancing AI adaptability, I promote responsible adoption across markets. For example, in journalism and academical environment, they uphold content integrity, while in creative areas, they support innovation without compromising human creativity.

8.Challenges and Limitations

Dataset limitations.

As I was mention about data collecting method, I have collected Articles which is normally including at least 2000 words. And there was like 1000 articles and after filter it by dates to make sure I have gather the articles before the time period AI content writing involved .only 82 were found for by that dates.

Next problem was those articles words count was too heavy to put directly into the Ollama Model to rewrite those sentences. And it cant get that much of word count and it tends to summarize the article .

So , After I have researching that I get to know. The best way to build a classification model is to use sentence wise data. therefore, I had to break every sentence and save it in another data set. but only split and rewrite like 6/83 articles which could be led to same tone for identify every human cluster that is going to use. And when ollama rewrote my sentences in AI it attempted to use more formal way. if I could make it back more. I would clarify the prompt to ollama more.

Model and Resource Limitations.

For the AI converter model. I tried to train a model using both machine learning and deep learning concepts to identify, which were never done by anyone when I read research articles.

But the limitations of the resources several kernal crashes did happened and even cloud services free services were did not attempt to do it.

9. Conclusion

This project focused on analyzing and addressing the challenges in classifying and converting AI-generated and human-written text. A critical part of the study was the detailed analysis of the structural, stylistic, and semantic differences between these text types, using insights from dependency parsing, stylometry, and document clustering. These analyses highlighted recurring patterns in AI-generated content, such as predictable sentence structures, limited vocabulary diversity, and less nuanced contextual relationships, which are key factors in distinguishing AI from human-authored texts.

The use of advanced sentence transformers, including BERT, **RoBERTa**, and **fastText**, played a significant role in these tasks. These models excel in encoding semantic and syntactic information, enabling the classifier to detect subtle differences between AI and human texts with high accuracy. For the conversion process, these transformers helped map AI-generated content to human-like styles, addressing gaps in tone, context sensitivity, and natural flow. BERT's contextual embeddings and **RoBERTa's** robust pretraining improved the conversion quality, while **fastText's** efficiency made it suitable for real-time applications.

The analysis also identified key problems in AI content writing, such as lack of creativity, redundancy, and difficulty in aligning with domain-specific requirements. These insights informed improvements to text generation and transformation methods, making them more adaptable to diverse real-world scenarios.

Future Work

Future research can delve deeper into enhancing dataset diversity, especially by including texts with complex linguistic features like idioms, metaphors, and cultural nuances. Advanced transformers and fine-tuning strategies can be explored to improve sentence-level transformations, ensuring the converted text maintains context and intent. Additionally, integrating ethical considerations, such as reducing bias in AI-generated content, will be crucial for the responsible application of these tools.

This work underscores the potential of combining analytical insights and transformer-based models to improve AI reliability, enabling better human-AI collaboration and fostering trust in AI systems.

10. References

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. http://arxiv.org/abs/1607.04606

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. http://arxiv.org/abs/1810.04805

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. http://arxiv.org/abs/1907.11692

11. Appendix

11.3. Evaluation tables output

EXAMPLE 1

In the rapidly evolving field of Natural Language Processing (NLP), the need for AI to produce more human-like content has become increasingly important. While AI-generated content can be efficient and scalable, it often lacks the nuanced emotional depth that human-written content possesses. This project aims to address this gap by conducting a sentiment analysis to differentiate between AI-generated and human-written texts.

EXAMPLE 1 OUTPUT

In the rapidly developing field of Natural Language Processing (NLP) , the requirement for AI to produce more human - like information has got increasingly crucial . While AI - created content can be efficient and scalable , it usually lack the nuanced emotional depth which human - authored content possesses . This project intends to address this gap by conducting a sentiment analysis to distinguish between AI - generated and man - authored text .

EXAMPLE 2

The "Circle of Life" is a concept that describes the interconnectedness and cyclical nature of life and death in the natural world. Popularized by the song from Disney's "The Lion King," it emphasizes that every living thing has its role and purpose. Birth, growth, reproduction, and death are all part of this cycle, with each stage contributing to the ecosystem's balance. For example, plants grow, animals eat them, and eventually, both return to the earth, nourishing new growth. This continuous cycle ensures the sustainability and harmony of all living organisms.

EXAMPLE 2 OUTPUT

The " Circle of Life " is a theory which describes the interconnectedness and cyclical nature of life and death in the natural world . Popularized by the song from Disney 's " The Lion King , " it underscores that every living thing has its purpose and function . Birth , development , reproduction , and destruction are all part of this cycle , with each phase added to the ecosystem 's balance . For example , trees thrive , pests eat them , and inevitably , both fall to the ground , nourishing new growth . This continuous cycle guarantees the sustainability and harmony of all live animals .

EXAMPLE 3

AI content writing holds significant importance for students due to its multifaceted benefits. Firstly, it serves as a valuable educational support tool by generating explanations, summaries, and practice questions, which can enhance students' understanding of complex topics. Additionally, AI can automate tasks such as note-taking and drafting, allowing students to allocate more time to critical thinking and analysis. This not only improves efficiency but also deepens their comprehension of the material. Furthermore, AI has the capability to create personalized learning content tailored to individual learning styles and paces, making education more accessible and effective for a diverse range of students.

EXAMPLE 3 OUTPUT

AI content writing takes significant importance for students due to its multifaceted benefits . Firstly , it helps as a valuable educational support tool by giving explanations , summaries , and practice questions , what can enhance pupils ' comprehension of complex issues . Additionally , AI can automate jobs such as mark - taking and drafting , allowing students to allocate more hours to critical thoughts and analysis . which not solely improves efficiency but additionally deepens the comprehension of the material . plus , AI has the ability to develop personalized education content tailored to individual education styles and paces , making education more accessible and effective for a diverse range of students .

EXAMPLE 4

There's been some chatter about how things are running lately. It feels like some decisions might not be aligning with what everyone needs. There's a sense that not everyone is getting a fair shot, and it's causing some frustration. It would be great if I could find a way to ensure that everyone is in the right place, doing what they do best. That way, things could run more smoothly, and everyone would feel more supported. Just something to think about!

EXAMPLE 4 OUTPUT

There 's been some chatter about how things are moving lately . It feels like some decisions might not be align with something everyone wants . There 's a sense that not anyone is get a fair shot , and it 's causing some frustration . It would be good if I could discover a way to see that everybody is in the good place , doing which you do best . That way , things could run more smoothly , and anyone would feeling more supported . Just some to remember about !

EXAMPLE 5

An unorganized system can significantly exacerbate the risks associated with handling very dangerous substances. One of the primary issues is the lack of protocols and procedures. In an unorganized system, there may be no clear guidelines or standard operating procedures (SOPs) for handling hazardous materials. This lack of structure can lead to inconsistent practices, increasing the likelihood of accidents and exposure.

Another critical aspect is inadequate training. Without a structured training program, personnel may not be fully aware of the risks associated with dangerous substances or the proper methods for handling them. This can result in improper use, storage, and disposal, all of which can lead to serious health and environmental hazards. Effective communication is also crucial in any system dealing with hazardous materials. An unorganized system may lack clear communication channels, leading to misunderstandings, misinformation, and delayed responses to emergencies.

Organized systems typically have robust safety measures in place, such as personal protective equipment (PPE), containment facilities, and emergency response plans. In an unorganized system, these measures may be lacking or inadequately implemented, increasing the risk of accidents and exposure. Regular monitoring and control are essential for managing dangerous substances. An un

EXAMPLE 5 OUTPUT

An unorganized organization can drastically worsen the risks related with handling extremely hazardous material . One of the main problems is the lack of protocols and procedures . In an unorganized organization , there may be no clear regulations or standard operating procedures (SOPs) for treating hazardous materials . This lack of structure can led to inconsistent practices , increasing the likelihood of injuries and exposure .

Another critical aspect is inadequate training . Without a structured education plan , personnel may not be completely mindful of the risks associated with dangerous materials or the proper techniques for treating those . That can resulting in improper use , storage , and disposal , all of which can leading to severe health and ecological risks . Good communication is also essential in any organisation covered with toxic chemicals . An unorganized organization may lack clear communication channels , contributing to mistakes , misinformation , and delayed reaction to crises

Organized organizations typically have robust safety measures in site , such as personal protective apparatus (PPE) , containment facilities , and crisis response plans . In an unorganized organization , these steps may be lacking or inadequately applied ,