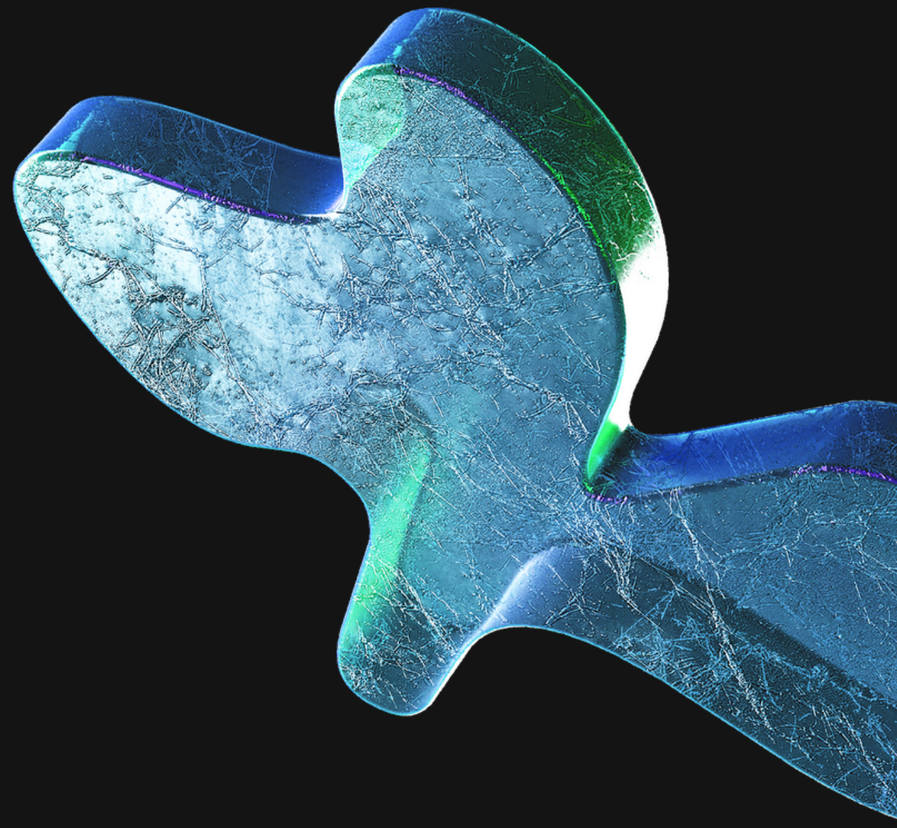
The Dancing Distributions Team

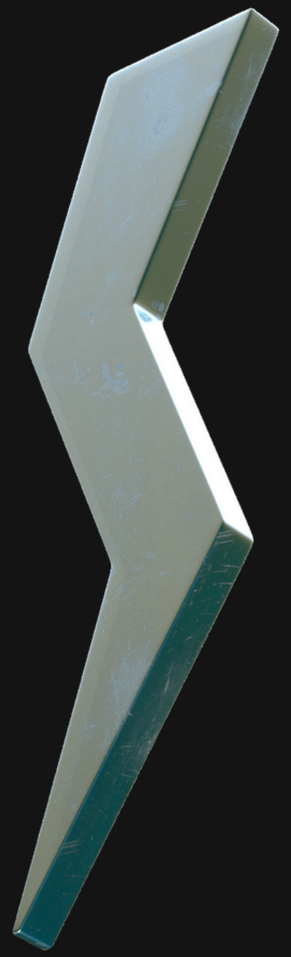
Italian NLP Corpus



Sentiment Analysis/Classification/Prediction Problems

Benedek Körösparti
Carolina Leuzzi
Mario Mirabile

Outline of the project

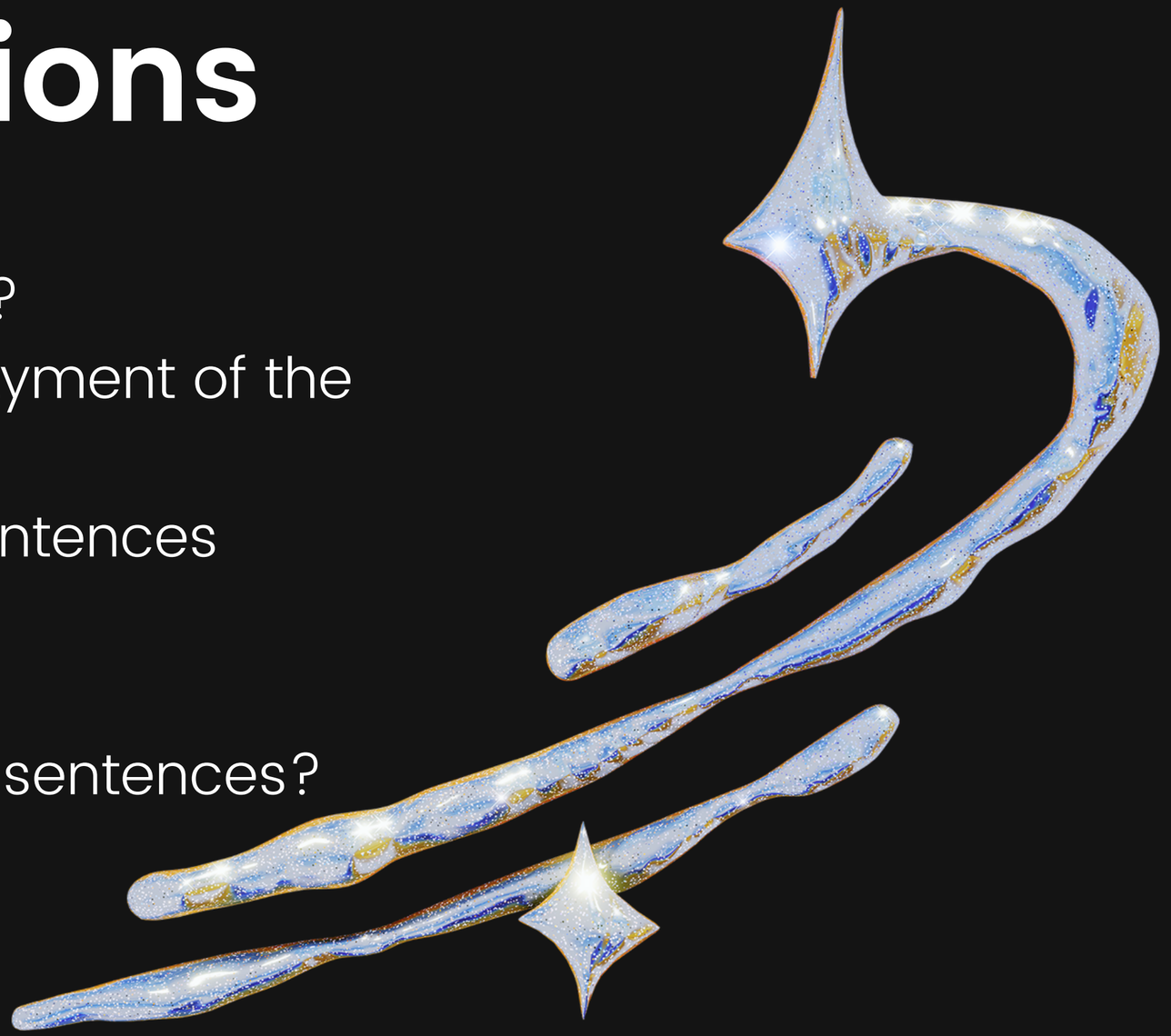


I. Overview of the project:

- It focuses on analyzing the "Italian NLP Corpus - Classification/Sentiment Analysis" dataset with 1,123 Italian sentences and 1,200 English sentences (we added new questions).
- Sentences rated on a complexity scale from 1 to 7 by human judges.
- Two different treebanks used: Italian Universal Dependency Treebank (IUDT) and Wall Street Journal section of the Penn Treebank.
- Aims: explore sentence complexity and the relationship between language and judgment scores.
- Utilized classification, sentiment analysis, and prediction techniques to identify patterns and differences.
- Project contributes to natural language processing field and enhances understanding of language complexity across languages.

II. Main questions

- What are the datasets about?
- Is it possible to score the sentyment of the sentences?
- Is it possible to classify the sentences according to the language?
- Is it possible to predict the judgments/complexity of the sentences?



III. Preparation, Preprocessing & Exploratory Data Analysis (EDA)

- We start with preprocessing activities on data.
- Drop irrelevant columns to cleanse data.
- Conduct exploratory analysis with prepared data.
- Visualize distributions using plots after analysis.

IV. LET'S START!

- Importing Libraries
  - Importing libraries for various functionalities such as data analysis, visualization, natural language processing, machine learning, and deep learning.

- Data Loading
  - Loading datasets from two URLs using pandas' read_csv() function and storing them in dataframes.

- Data Exploration
  - Using a function to print unique values from all columns in the loaded dataframes.

- Data Cleaning
  - Checking for missing values in the English and Italian datasets and printing the results.

```python
# Python Standard Libraries
import string
from collections import Counter

# Third-Party Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.sentiment import SentimentIntensityAnalyzer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.preprocessing import LabelEncoder, MinMaxScaler, Standar
from tensorflow import keras
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Flatten, I
from tensorflow.keras.regularizers import l1
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlate
from keras.utils import np_utils
from keras import regularizers
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
import nlpaug.augmenter.word as naw
from wordcloud import WordCloud
```

```python
df_en = pd.read_csv(url_1)
df_en.head()
```

	ID	SENTENCE	judgement1	judgement2
0	951586555	Amcast Industrial Corp. said it plans to repur...	4	2
1	951587546	GDP is the total value of a nation's output of...	3	1
2	951587247	Town & Country Ford in Charlotte, N.C., still ...	1	2
3	951586819	A couple in Rockford, Ill., raised $ 12,591 ea...	1	2
4	951586503	Yesterday the company	3	5

```python
df_it = pd.read_csv(url_2)
df_it.head()
```

	ID	SENTENCE	judgement1	judgement2
0	951583956	Quanto alla camminata incerta, va attribuita, ...	3	7
1	951584097	Campione di rugby una delle vittime, un altro ...	2	4
2	951583629	Costo dalle 100.000 alle 150.000 il mq.	2	3
3	951583242	A Valona, dove ieri è stata convocata un'altra...	1	5
4	951583156	Il mito di Allende, più di 14	4	5

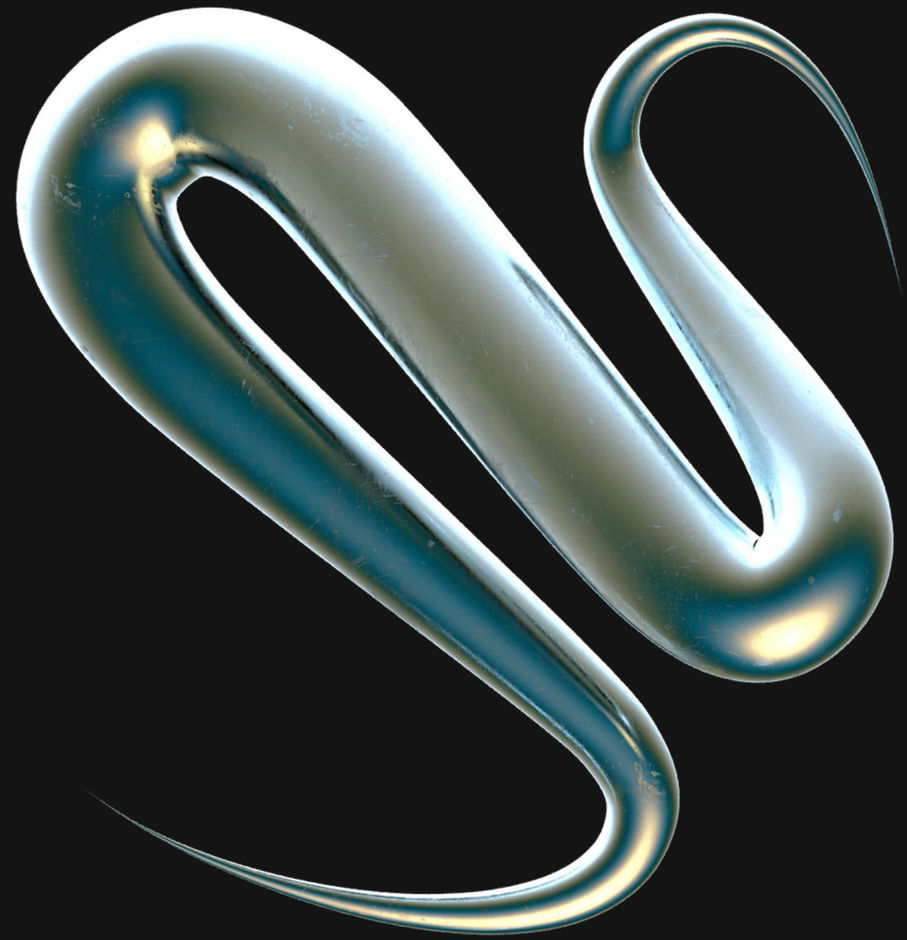
```python
# Load the datasets
df_1 = pd.read_csv(url_1)
df_2 = pd.read_csv(url_2)

# Function to print column names and unique values
def print_unique_values(df):
    for column in df.columns:
        unique_values = df[column].unique()
        print(f"Column name: {column}")
        print(f"Unique values: {unique_values}\n")

print("Dataset 1:")
print_unique_values(df_1)

print("Dataset 2:")
print_unique_values(df_2)
```

```
Column name: judgement1
Unique values: [4 3 1 2 6 7 5]

Column name: judgement2
Unique values: [2 1 5 4 3 7 6]

Column name: judgement3
Unique values: [1 3 5 2 4 7 6]
```

V. Latent Dirichlet Allocation (LDA)

LDA helps uncover underlying themes and patterns in the English and Italian datasets.

By assigning topics to each sentence, it becomes easier to categorize and analyze the content of the texts, facilitating further exploration and interpretation of the textual data.



English Dataset

The English dataset consists of 4,231 unique words and has been divided into 7 topics.
Each topic represents a group of words that are highly associated with each other based on their occurrence in the dataset.
For each topic, the top 15 words that contribute to that topic the most are displayed.
The topics cover various themes such as corporate affairs, stock market, finance, and market trading.

```
4231
7
THE TOP 15 WORDS FOR TOPIC #0
['corp', 'mr', 'board', 'named', 'vice', 'new', 'officer', 'federal', 'sales', 'billion', 'chairman', 'said', 'chief', 'executive', 'president']


THE TOP 15 WORDS FOR TOPIC #1
['based', 'acquired', 'prices', 'shares', 'higher', 'year', 'stock', 'mr', '10', 'company', 'new', 'said', 'corp', 'market', 'million']


THE TOP 15 WORDS FOR TOPIC #2
['pay', 'bank', 'corp', 'shearson', 'hutton', 'lehman', 'quarter', 'index', 'mr', 'treasury', 'year', 'million', 'company', 'said', 'billion']


THE TOP 15 WORDS FOR TOPIC #3
['group', 'board', 'time', 'president', 'priced', 'sales', 'issue', 'year', 'mr', '15', 'yield', 'company', 'shares', 'said', 'million']


THE TOP 15 WORDS FOR TOPIC #4
['income', '32', 'months', 'fell', 'rose', 'securities', 'company', 'earlier', 'quarter', 'cents', 'net', 'said', 'year', 'share', 'million']


THE TOP 15 WORDS FOR TOPIC #5
['debt', 'agreed', 'offering', 'city', 'time', 'index', 'shares', 'september', 'investment', '000', 'corp', 'price', 'new', 'said', 'million']
...
THE TOP 15 WORDS FOR TOPIC #6
['said', 'company', 'shares', '50', 'yesterday', 'share', 'mr', 'cents', 'closed', 'composite', 'york', 'new', 'exchange', 'trading', 'stock']
```

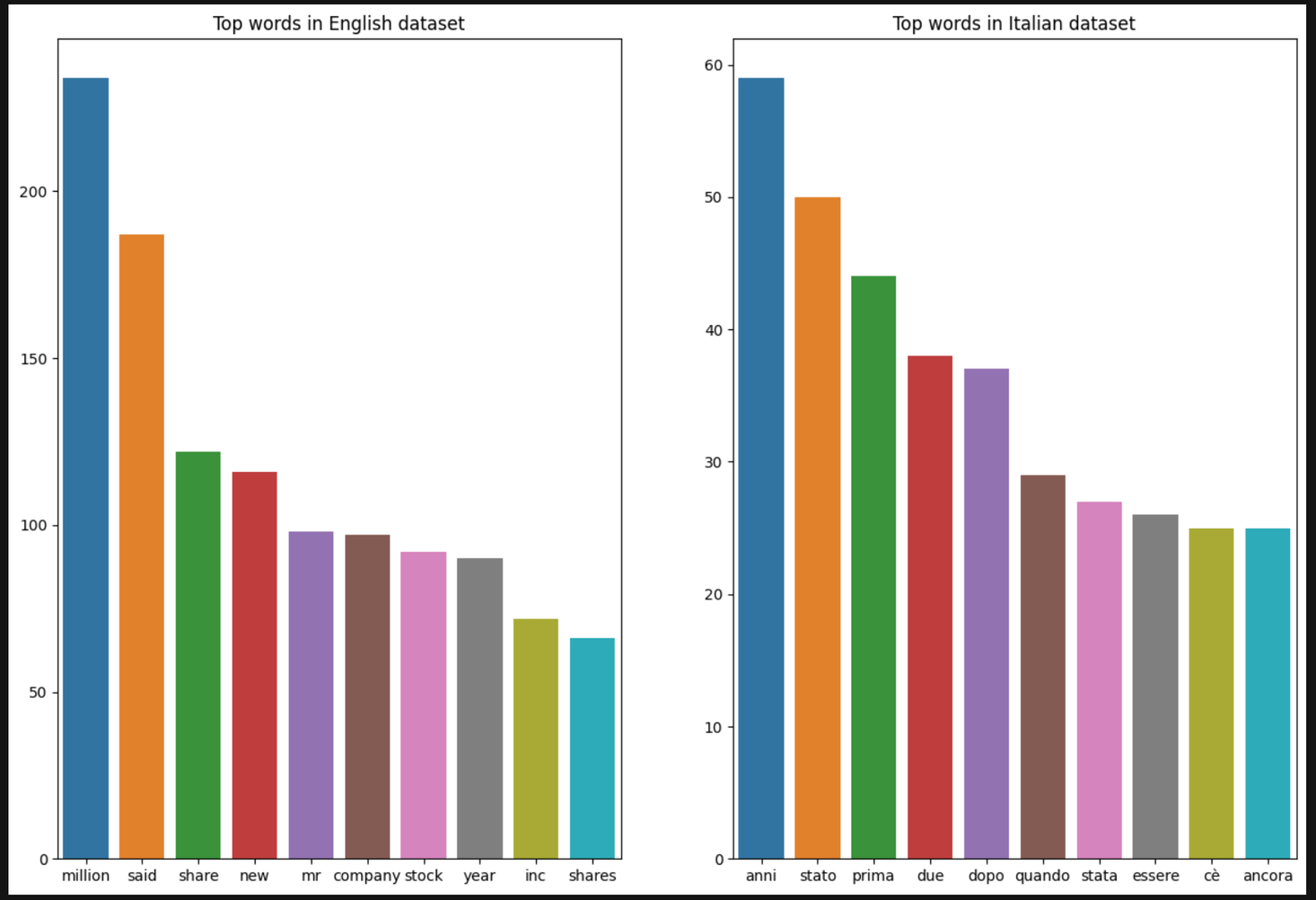
Italian Dataset

The Italian dataset contains 5,823 unique words and has also been divided into 7 topics. Each topic represents a cluster of words that are closely related to each other within the dataset.
The top 15 words contributing to each topic are displayed
The topics cover a range of subjects including politics, finance, social issues, and daily life.

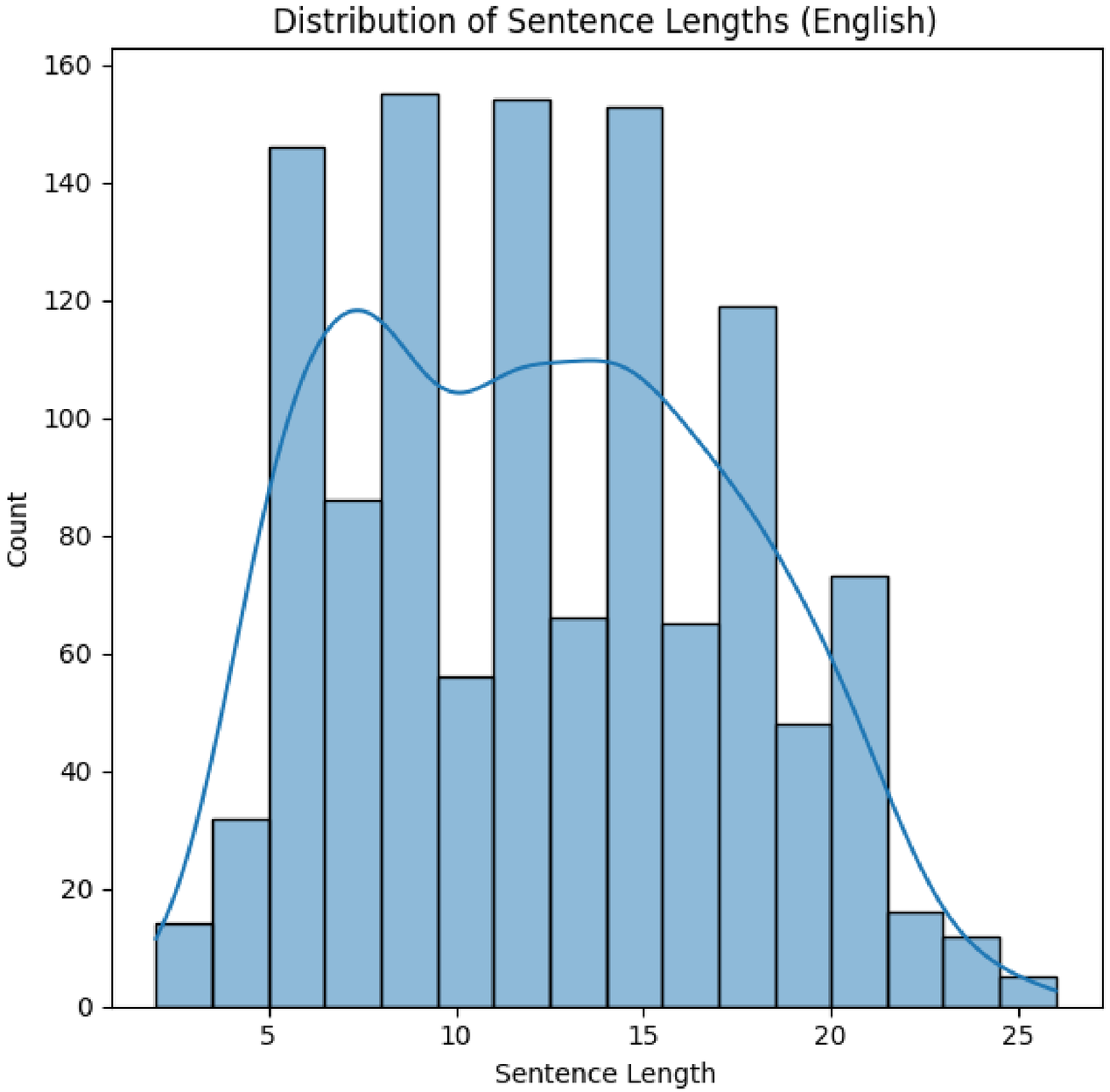
```
5823
7
THE TOP 15 WORDS FOR TOPIC #0
['persone', 'ora', 'stati', 'altri', 'rivolta', 'cento', 'italia', 'dopo', 'governo', 'società', 'dollari', 'primo', 'stato', 'albania', 'anni']


THE TOP 15 WORDS FOR TOPIC #1
['10', 'ancora', 'oro', 'fine', 'secondo', 'parco', 'stato', 'prima', 'italia', 'dopo', 'primo', 'mezzo', 'presidente', 'città', 'poco']


THE TOP 15 WORDS FOR TOPIC #2
['presidente', 'volta', 'invece', 'però', 'ora', 'momento', 'parte', 'tempo', 'cinque', 'così', 'tre', 'anni', 'stato', 'prima', 'quando']


THE TOP 15 WORDS FOR TOPIC #3
['solo', 'stessa', 'dopo', 'ancora', 'prima', 'stato', 'volte', 'ieri', 'oggi', 'sempre', '000', 'fa', 'lire', 'due', 'anni']


THE TOP 15 WORDS FOR TOPIC #4
['ultima', 'finanziarie', 'valona', 'anno', 'fine', 'donna', 'soltanto', 'certo', 'molto', 'dopo', 'senza', 'qualche', 'anni', 'forse', 'stata']


THE TOP 15 WORDS FOR TOPIC #5
['metri', 'prima', 'verso', 'parte', 'proprio', 'sotto', 'albanese', 'due', 'dopo', 'meno', 'altri', 'poi', 'essere', 'anni', 'stato']
...
THE TOP 15 WORDS FOR TOPIC #6
['mai', 'ormai', 'ancora', 'tre', 'dice', 'stata', 'anni', 'stato', 'qui', 'giorno', 'vita', 'ogni', 'cinema', 'prima', 'due']
```

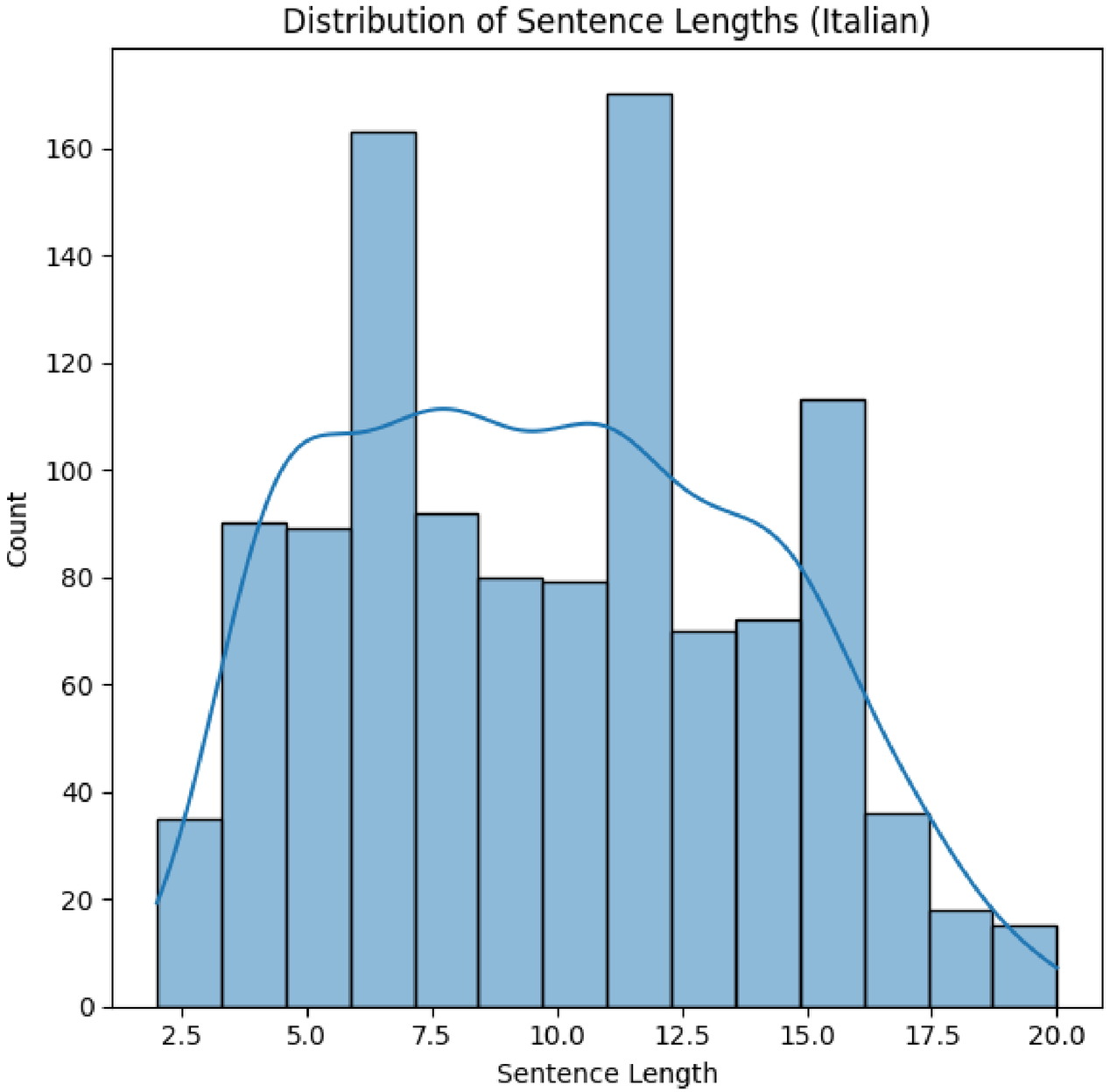
VI. Further preprocessing and EDA

```python
# Function to get top n words
def get_top_n_words(corpus, n=10):
    # Flatten the list of words and find the frequency of each word
    words = [item for sublist in corpus for item in sublist]
    freq_dist = nltk.FreqDist(words)
    return freq_dist.most_common(n)

# Check the shape of the datasets
print(f"English dataset shape: {df_en.shape}")
print(f"Italian dataset shape: {df_it.shape}")

# Check a few random rows from the datasets
print("\nA few random rows from English dataset:")
print(df_en.sample(5))

print("\nA few random rows from Italian dataset:")
print(df_it.sample(5))

# Get the top 10 common words in each dataset
top_words_en = get_top_n_words(df_en['SENTENCE'], 10)
top_words_it = get_top_n_words(df_it['SENTENCE'], 10)

print("\nTop 10 common words in English dataset:")
for word, freq in top_words_en:
    print(f"{word}: {freq}")

print("\nTop 10 common words in Italian dataset:")
for word, freq in top_words_it:
    print(f"{word}: {freq}")

# Visualize the top 10 common words in each dataset
fig, ax = plt.subplots(1, 2, figsize=(15, 10))

# English
words_en = [w[0] for w in top_words_en]
counts_en = [w[1] for w in top_words_en]
sns.barplot(x=words_en, y=counts_en, ax=ax[0])
ax[0].set_title('Top words in English dataset')

# Italian
words_it = [w[0] for w in top_words_it]
counts_it = [w[1] for w in top_words_it]
sns.barplot(x=words_it, y=counts_it, ax=ax[1])
ax[1].set_title('Top words in Italian dataset')

plt.show()
```

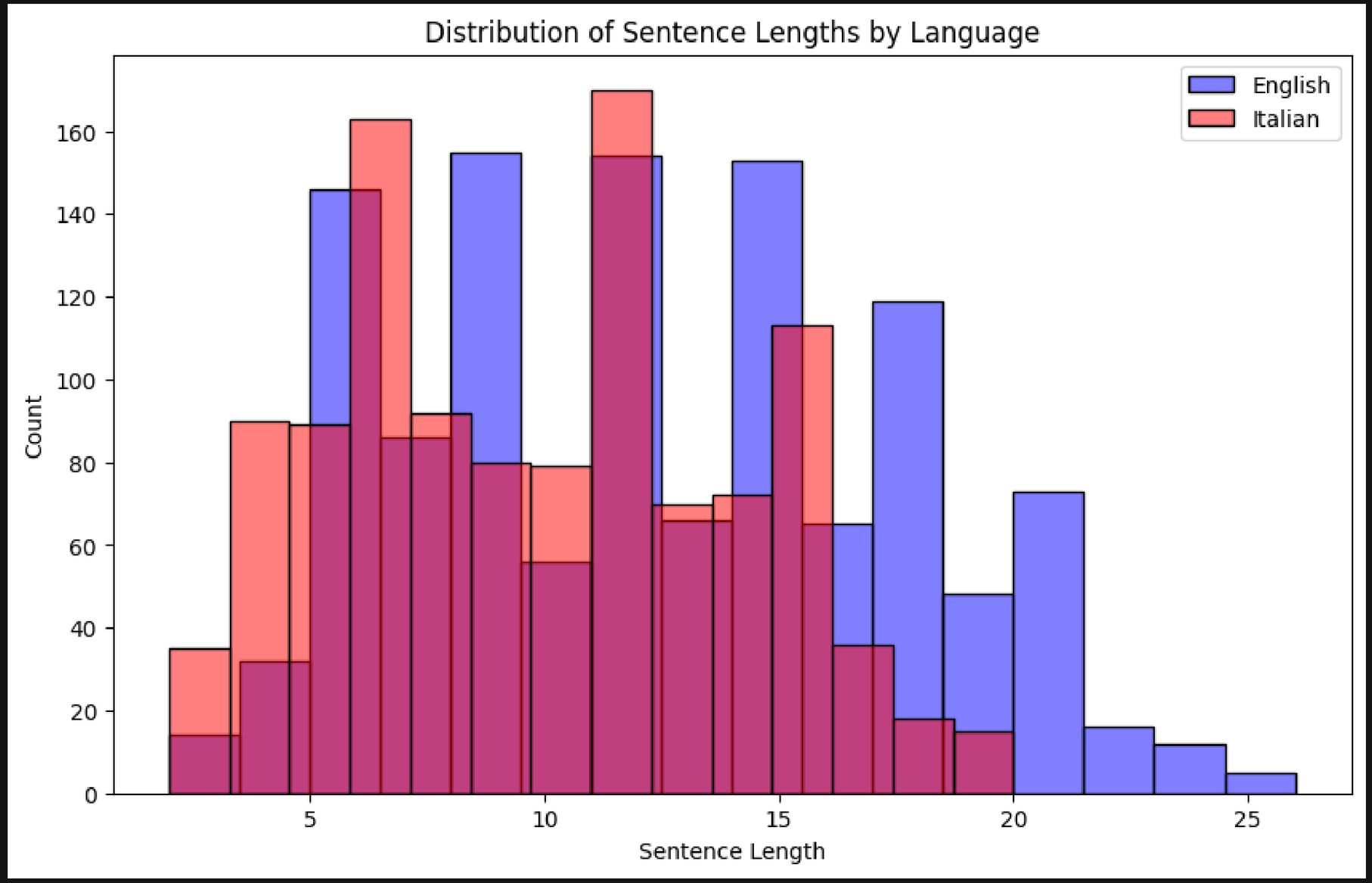


Average Sentence Lenghts Distribution



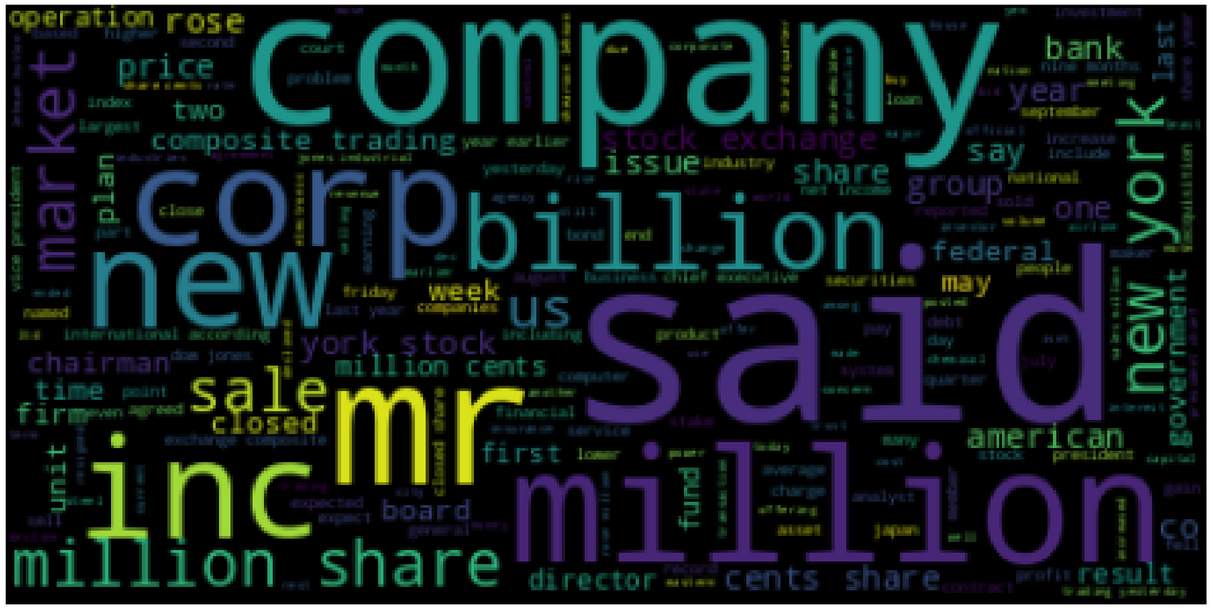


Distribution of sentence lengths by language



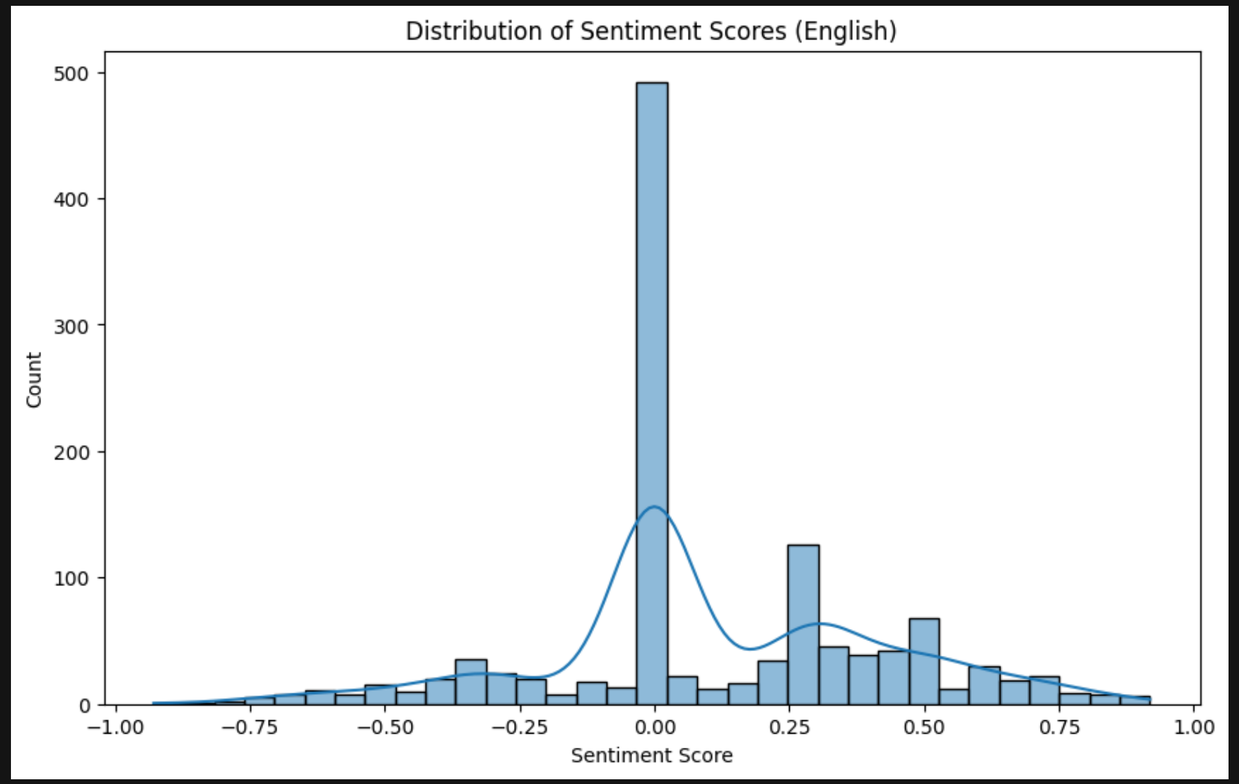
Word Clouds

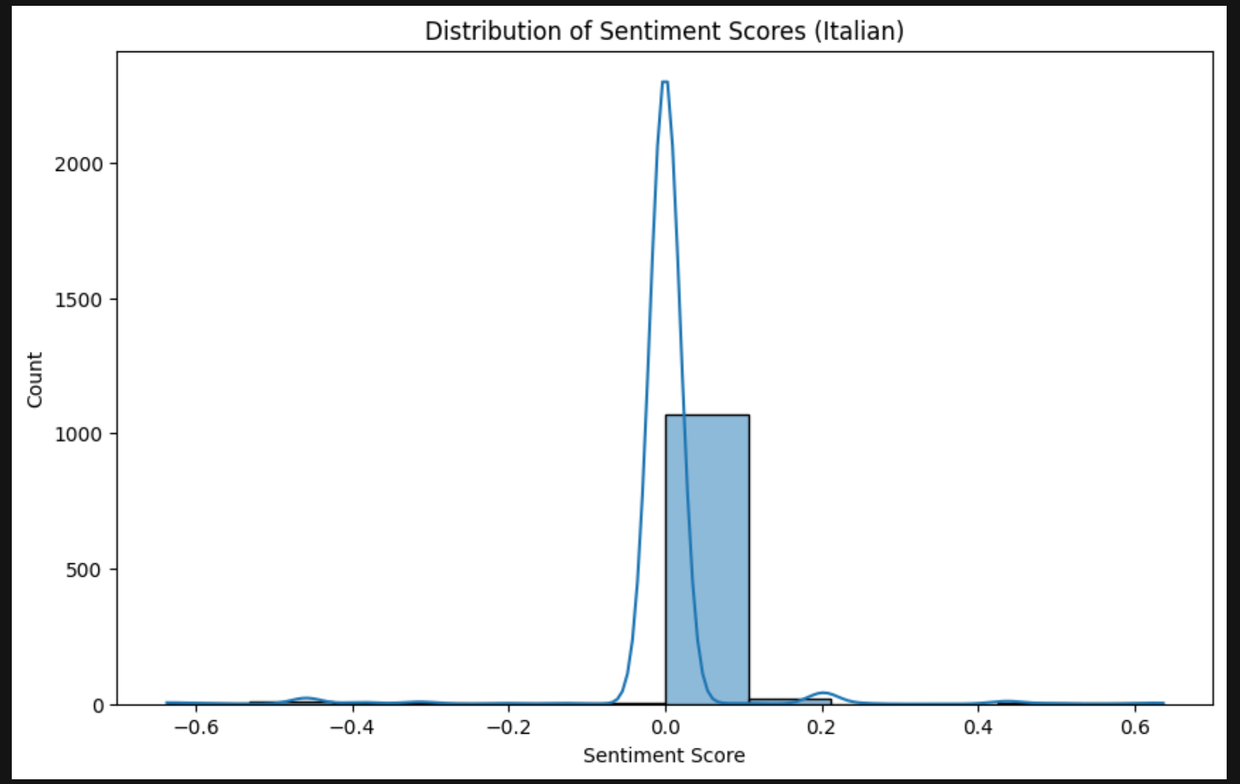
Word Cloud (English)



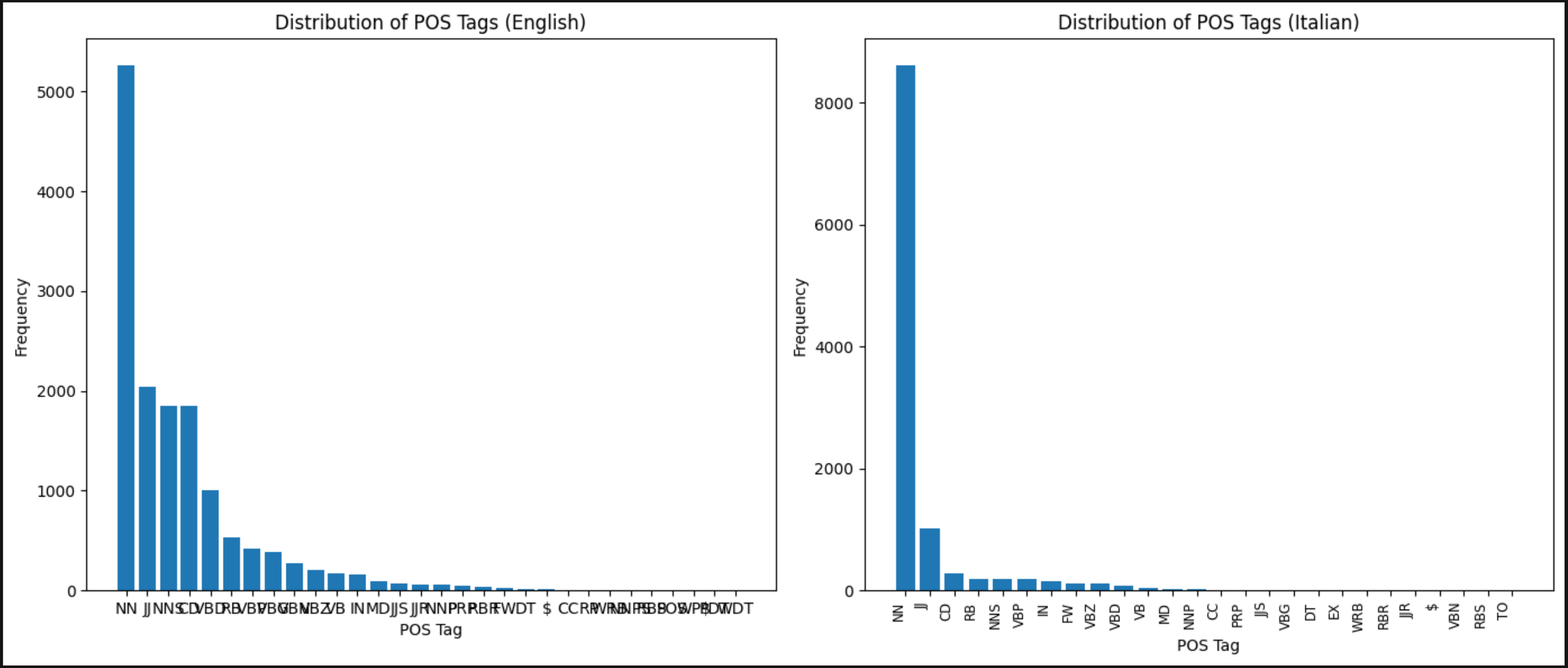
Word Cloud (Italian)

Distribution of the Sentiment Scores





Distribution of POS tags



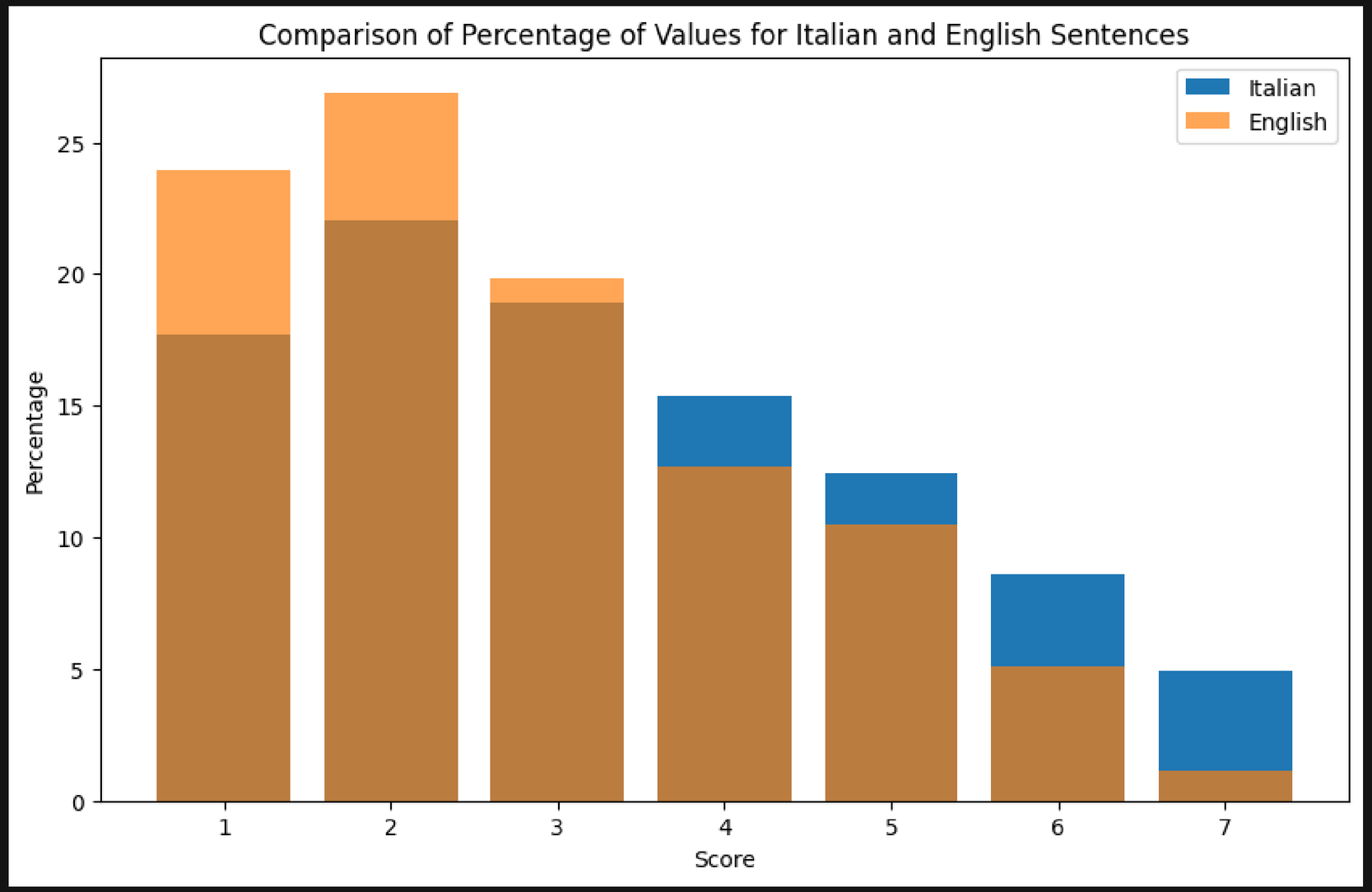
Comparison of Percentage of Judgs (%) for IT & ENG

For Italian sentences:

The most common score is 2, with a percentage of approximately 22.06%.

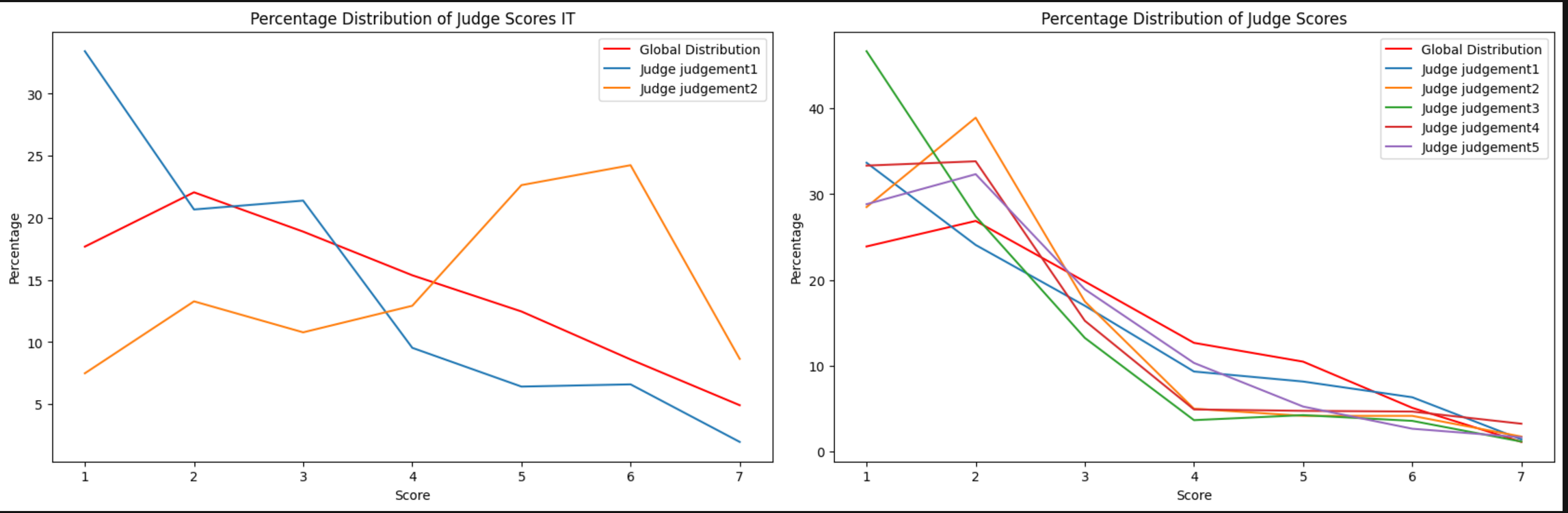
For English sentences:

The most common score is 2, with a percentage of approximately 26.88%.

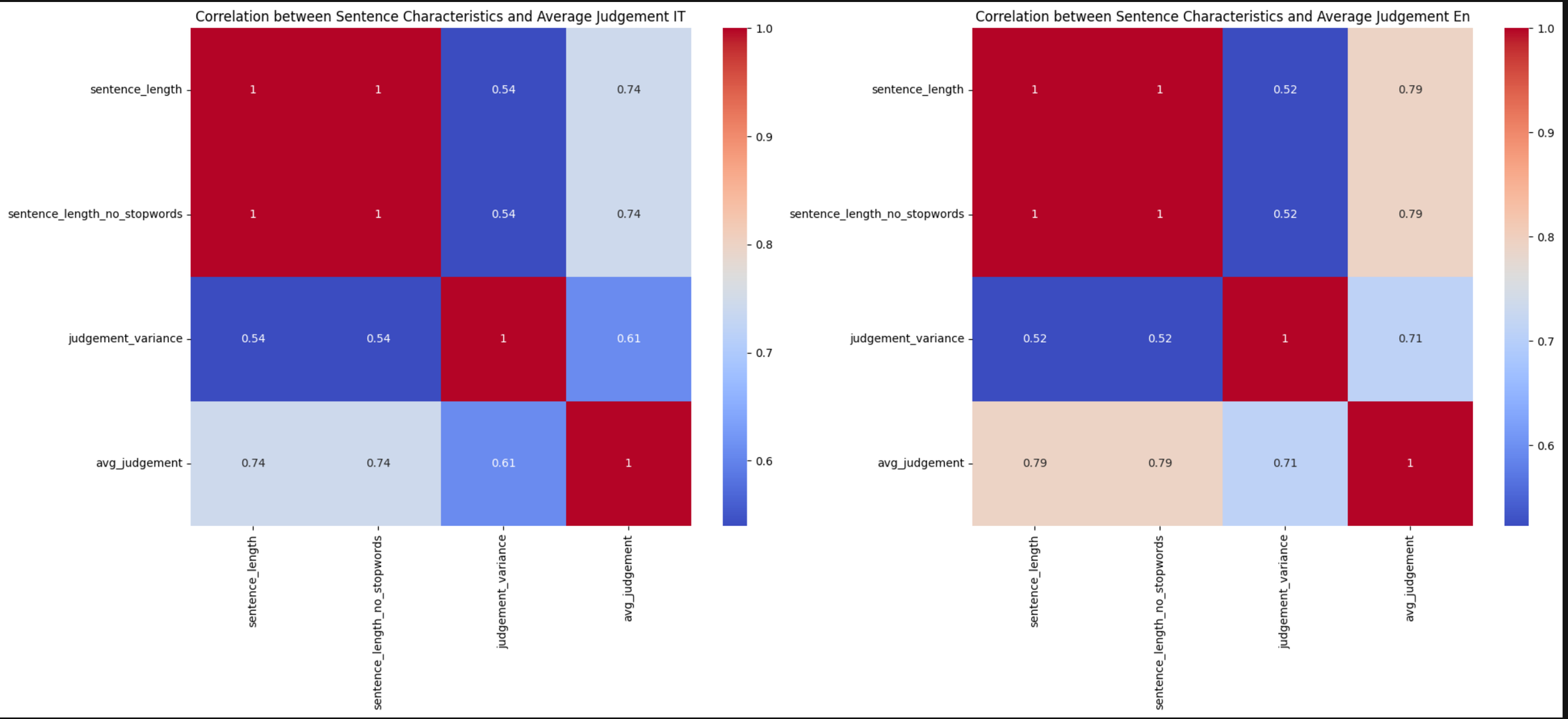


Score Distribution Analysis

Compared global score distribution with individual judges' distributions, revealing variations and subjectivity in evaluation process, providing insights into language evaluation dynamics and biases.



Sentence Metrics and Correlations



CLASSIFYING AND PREDICTING LANGUAGE AND COMPLEXITY.

IS IT POSSIBLE?

VII. Linear Regression

In this context, we are trying to predict a continuous output (the average judgment) based on certain input variables, which fits the definition of a regression problem.

NOT EXCITING RESULTS...

Linear Regression model:

- **English dataset accuracy: 63.36%**
- **Italian dataset accuracy: 55.08%**

VIII. Random Forest

We wanted to classify the specific judgments.

Each judgment is a discrete number from 1 to 7, so this is a multi-class classification problem.

We used a Random Forest classifier, which is used in ML because it handles multi-class classification tasks well.

POOR RESULTS!

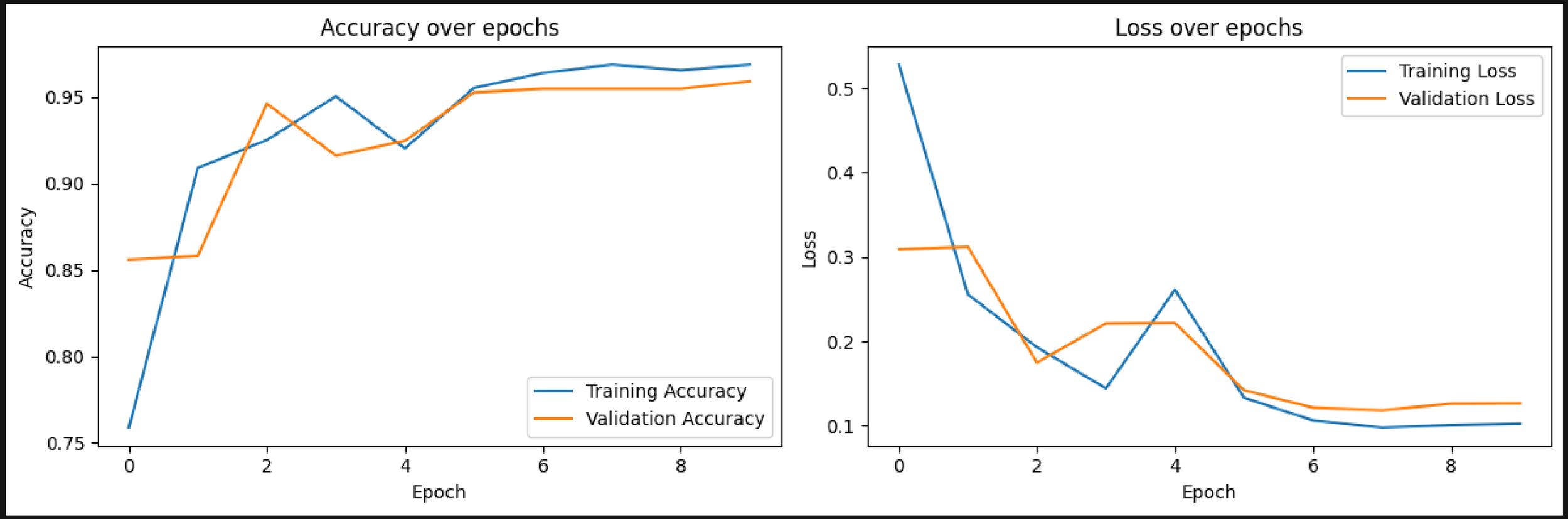
Random Forest model.

- **English dataset accuracy: 29.29%**
- **Italian dataset accuracy: 22.47%**

WHILE TRYING TO UNDERSTAND THE CHARACTERISTICS OF OUR DATA, WE DECIDED TO CHANGE OUR APPROACH.

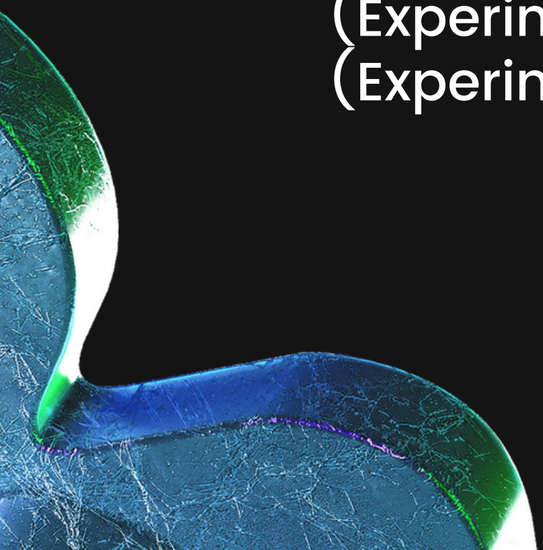
IX. Binary Classification Problem: English or Italian?

RNN with LSTM layer used for distinguishing English and Italian sentences.
Model performance evaluated using loss and accuracy metrics.
INTERESTING RESULTS: > 0.95 validation accuracy and ≈ 0.10 loss



X. RNNs and FNNs. A Difficulty Prediction Problem

- 1º attempt: Sentence Difficulty Prediction using RNN:
  - LSTM-based RNN used to predict sentence difficulty.
  - Performance assessed based on loss, MAE, and R^2 score.
  - Final test loss: 0.0108 (Experiment 2.1), 0.0260 (Experiment 2.2)



- 2º attempt: Sentence Difficulty Prediction using FNN:
  - FNN model comprising multiple layers used for similar task.
  - Performance evaluated using same metrics as previous experiment.
  - Final test loss: 0.0135, MAE: 0.0909, R^2 Score: 0.645 (Experiment 3.1)
  - Final test loss: 0.0210, MAE: 0.1170, R^2 Score: 0.581 (Experiment 3.2)

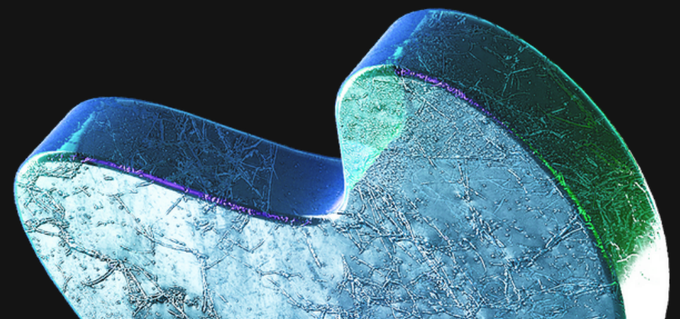
Comparing RNNs and FNNs for Sentence Complexity Prediction

RNNS

- The code implements an LSTM model, a type of RNN, for predicting sentence difficulty based on text analysis.
- Lower loss, MAE, and test error values indicate accurate predictions. Higher R2 scores suggest better capture of underlying patterns.
- The English model outperforms the Italian model, with higher R2 score and lower MAE, indicating more effective capture of the relationship between input sentences and difficulty levels.

FNNS

- The model demonstrates accurate sentence difficulty predictions with lower loss, MAE, and test error values.
- English model outperforms Italian model with higher R2 score and lower MAE on the test set.
- The difference in performance could be due to language characteristics or dataset size. RNNs have feedback connections, while FNNs only move data forward through layers.



XI. Fine Tuning & Overfitting Mitigation

English dataset:

- Training accuracy: Starts at 0.26, reaches 0.92 by the end.
- Validation accuracy: Starts at 0.36, fluctuates around 0.40-0.45.
- Training loss: Decreases steadily from 1.63 to 0.26.
- Validation loss: Increases from 1.43 to 1.63.

Italian dataset:

- Training accuracy: Starts at 0.28, reaches 0.94 by the end.
- Validation accuracy: Starts at 0.38, fluctuates around 0.40-0.41.
- Training loss: Decreases steadily from 1.64 to 0.31.
- Validation loss: Increases from 1.52 to 1.63.

Dropout regularization, and Input and Output layers improves performance by reducing overfitting.

Models show increased training accuracy, reduced gap with validation accuracy, and less overfitting to training data.

Further improvements can be made by adjusting hyperparameters, and using data augmentation and other regularization techniques? Let's try!

Enhancing FNN Models with Data Augmentation and L2 Regularization

- Data Augmentation
  - Data is augmented using synonym replacement.
  - Training and testing sets prepared from augmented dataframes.

- L2 Regularization
  - Introduction of L2 regularization to mitigate overfitting.
  - Models retrained with L2 regularization and performance evaluated again.

- Outcomes:
  - Effective use of data augmentation and L2 regularization in FNN models.
  - Potential improvements in model performance and generalization highlighted.
  - Importance of these techniques in improving sentence complexity prediction and reducing overfitting demonstrated.

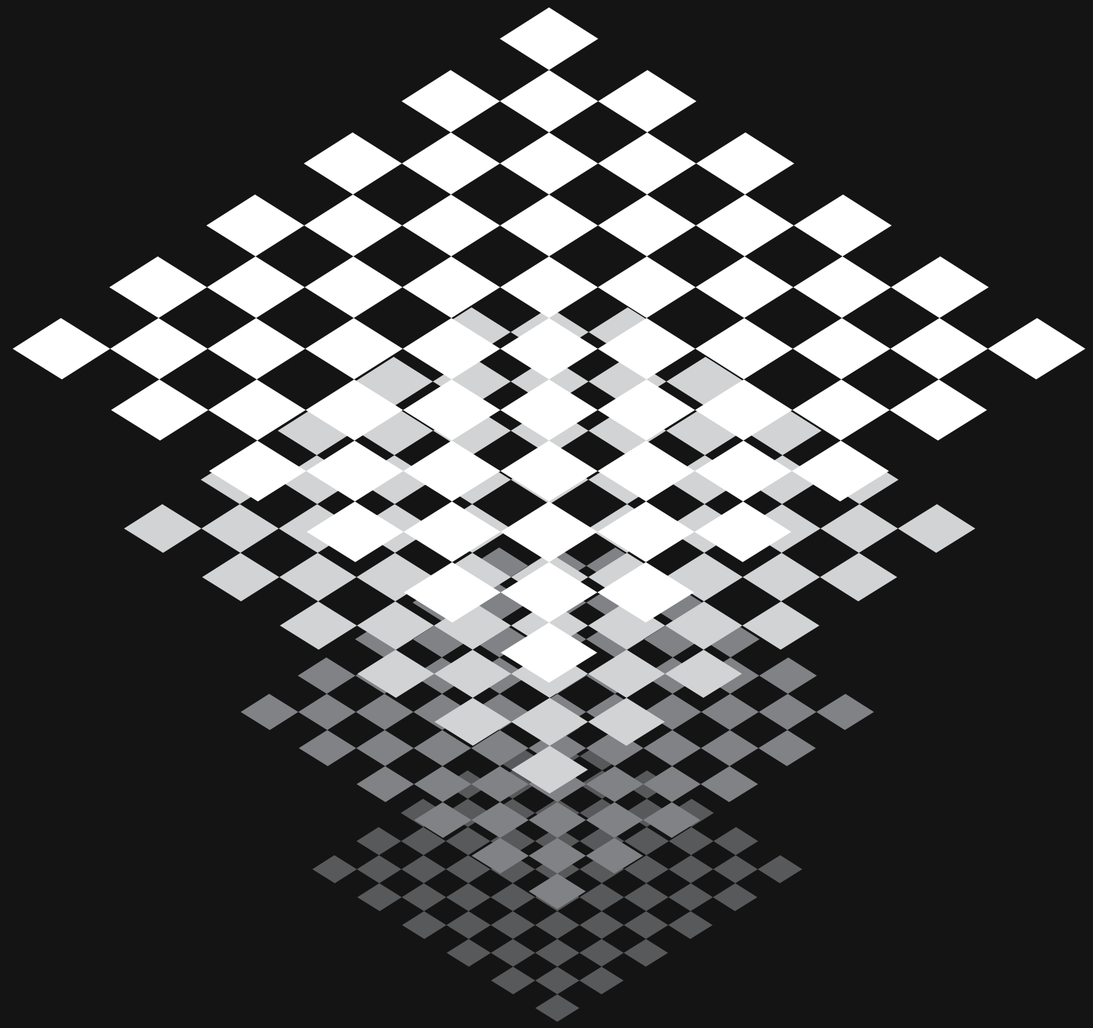
Possible space for improvements? Let's try again!

Enhancing FNN Models with Batch Normalization and Regularization

- Batch normalization and L2 regularization
  - Design of models for both languages with batch normalization and L2 regularization.

- Training and Evaluation
  - Implementation of early stopping and learning rate reduction.
  - Performance evaluation based on accuracy metrics.

- Outcomes:
  - Identified effective techniques and areas for improvement.



XII. Final considerations

Technical part

• While **RNNs** reach **over 95% of accuracy in classifying the language** of the sentences, other models achieved **70% accuracy on English and Italian in predicting complexity**, indicating language robustness.
When overfitting observed, we tried to mitigate it with specific techniques: s data augmentation, (L1 and L2) regularization, and Output, Input, and Batch layers for better generalization.
• Synonym replacement a**ugmented data**, enhancing model generalization.
• Techniques like batch normalization, dropout, L2 regularization, and Adam optimizer i**mproved stability and convergence**.

Judges and scores

- Human judgments of sentence difficulty vary, revealing **subjectivity in assessment**.
- Limited judge information raises consistency and **reliability concerns.**
- Expert labeling offers objectivity, but subjective analysis introduces **bias.**
- Clear guidelines and **standardized criteria** are needed to reduce subjectivity.

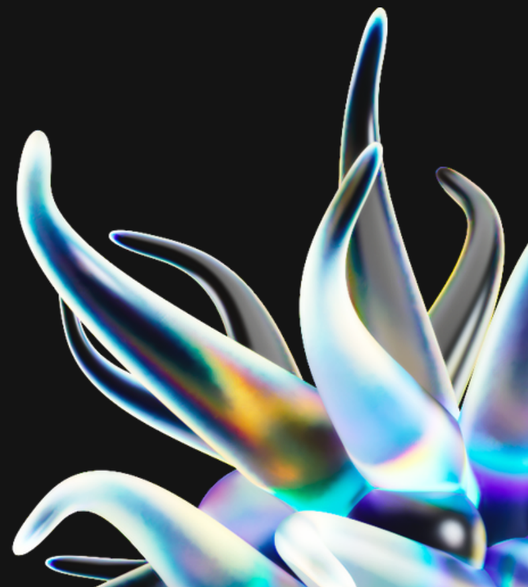
Possible Applications

- The algorithm benefits language learning, **adaptive reading,** filtering, speech therapy, and accessibility support.
- **Personalized recommendations** engage users by considering difficulty levels.
- The algorithm optimizes **filtering and recommendations** beyond language learning.
- Implementation challenges include universal scale establishment and diverse **training data acquisition.**

XIII. References

Brunato D., De Mattei L., Dell'Orletta F., Iavarone B., Venturi G. (2018) "Is this Sentence Difficult? Do you Agree?". In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), 31–4 November, Bruxelles.

Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC). Italian Natural Language Processing Lab. http://www.italianlp.it/.



Thank you for your attention!