# 

# 

# **Tweet Similarity Analysis with Transformer Embeddings**

Nfidsa Halima

Djait Ikram

[Tweet Similarity Analysis with Transformer Embeddings](#_jaq6eut7ajgv)

[1. Introduction:](#_leoaxxllv0kf)

[2. Methodology:](#_acdey51k9pqg)

[2.1 Data Preparation:](#_rkq881ft9z3j)

[2.2 Data Preprocessing:](#_dkcl4fqnsa35)

[2.3 Model Architecture:](#_ion3vbl6zr4s)

[2.4 Evaluation:](#_3er3reg0nlfc)

[3. Model Architecture:](#_m4me24eus1qg)

[3.1 Embedding Layer:](#_704fvkmctcoz)

[3.2 Transformer Encoder:](#_sucp3sv2bdwq)

[3.3 Feature Extraction:](#_t1kjgw2osb9)

[3.4 Manhattan Distance Calculation:](#_u5mkkww9maxt)

[3.5 Dense Layer:](#_yrpxq4px78u4)

[4. Results:](#_jz8botp6bnba)

## 

## **1. Introduction:**

The objective of this project is to develop a model capable of analyzing the semantic similarity between pairs of tweets and providing a similarity score indicating the likelihood that they originated from the same user. This model utilizes transformer embeddings for text representation and distance calculations.

## **2. Methodology:**

### **2.1 Data Preparation:**

* Tweet Pair Generation: Pairs of tweets are randomly sampled from the dataset to create a training and testing set. Techniques like stratification are considered to ensure a balanced representation of pairs from the same user and different users.
* Labeling: Each tweet pair is labeled based on whether they come from the same user or different users. Same-user pairs are labeled as 1, indicating high similarity, while different-user pairs are labeled as 0, indicating low similarity.

### **2.2 Data Preprocessing:**

* Text Cleaning: The text data undergoes preprocessing steps including lowercasing, punctuation removal, stopwords removal, and stemming to ensure consistency and improve model performance.

### **2.3 Model Architecture:**

* Embedding Layer: Pre-trained GloVe embeddings are used to represent words in the tweet text.
* Transformer Encoder: A pre-trained BERT transformer model is employed to encode the tweet text and capture contextual information.
* Feature Extraction: The output of the transformer encoder serves as tweet representations.
* Manhattan Distance Calculation: Manhattan distance is calculated between the representations of tweet pairs to measure similarity.
* Dense Layer: A dense layer with sigmoid activation is added to produce a similarity score between 0 and 1.

### **2.4 Evaluation:**

* Evaluation Metrics: Precision, Recall, and F1 Score are computed to evaluate the model's performance on the testing set. Precision measures the proportion of correctly identified same-user pairs out of all pairs predicted as same-user. Recall measures the proportion of correctly identified same-user pairs out of all actual same-user pairs. F1 Score is the harmonic mean of precision and recall, providing a balanced performance measure.

## **3. Model Architecture:**

### **3.1 Embedding Layer:**

* Utilized pre-trained GloVe embeddings for word representation, ensuring that semantic information is preserved in tweet text.

### **3.2 Transformer Encoder:**

* Employed a pre-trained BERT transformer model to encode tweet text, capturing contextual information and improving text understanding.

### **3.3 Feature Extraction:**

* Extracted features from the output of the transformer encoder to represent tweet pairs in a vectorized format suitable for distance calculation.

### **3.4 Manhattan Distance Calculation:**

* Calculated the Manhattan distance between tweet representations to measure the similarity between tweet pairs.

### **3.5 Dense Layer:**

* Added a dense layer with sigmoid activation to produce similarity scores between 0 and 1, providing a quantitative measure of tweet pair similarity.

## **4. Results:**

* Quantitative Metrics on Testing Set:
  + Precision: 0.44
  + Recall: 0.5
  + F1 Score: 0.47