

Tamil Text Summarization

Batch B group 19

gle

te

ie

petitions

sets

els

e

ussions

n

3

Work

id

I Wikipedia Arti...

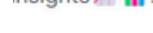
I text summar...

I Stopwords

i & Cats Images

ind Dog

D

Insights  ...

Predict Accid...

Active Events

Search

GAURAV · UPDATED 4 YEARS AGO

15 New Notebook Download (100 MB) :

Tamil Wikipedia Articles

127k cleaned wikipedia articles - with a train and test set to benchmark your LM

Data Card Code (3) Discussion (0) Suggestions (0)

About Dataset

This data set consists of 127k Wikipedia Articles which have been cleaned.

It has a Train set and Validation set, which were used to train and benchmark Language Models for Tamil in the repository [NLP for Tamil](#)

The scripts which were used to fetch and clean articles can be found [here](#)

Thanks to [Ravi](#) for sharing this data set

Feel free to use this data set creatively and for building better Language Models

Usability 6.47

License CC BY-SA 4.0

Expected update frequency Not specified

Tags Business, Finance

Data Explorer

Version 1 (500.46 MB)

train (1 directories)   

About this directory  Add Suggestion

Training Set

train

- train
- valid

train : 350+ Wikipedia articles
Valid : 90 Wikipedia articles

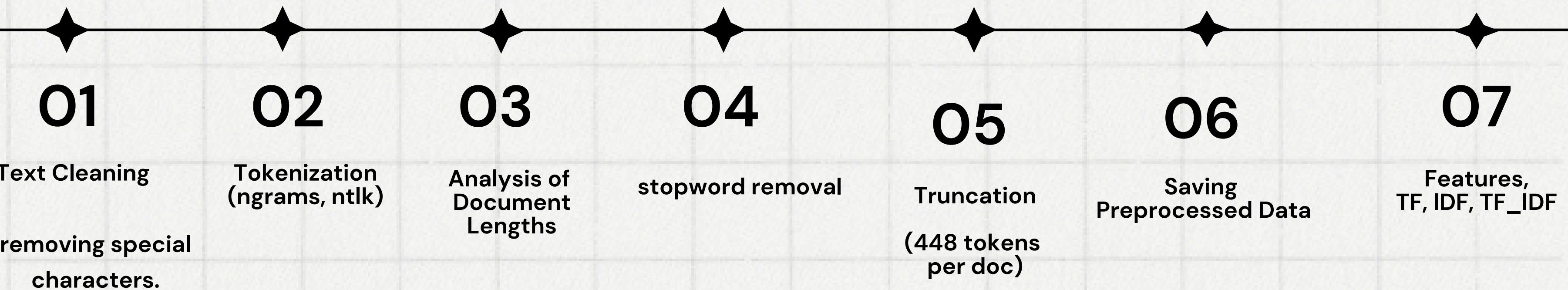
Train

 AA_wiki_00	06-05-2024 08:28 PM	Text Document
 AA_wiki_01	06-05-2024 08:28 PM	Text Document
 AA_wiki_02	06-05-2024 08:28 PM	Text Document
 AA_wiki_03	06-05-2024 08:28 PM	Text Document
 AA_wiki_04	06-05-2024 08:28 PM	Text Document
 AA_wiki_05	06-05-2024 08:28 PM	Text Document
 AA_wiki_06	06-05-2024 08:28 PM	Text Document
 AA_wiki_07	06-05-2024 08:28 PM	Text Document
 AA_wiki_10	06-05-2024 08:28 PM	Text Document
 AA_wiki_11	06-05-2024 08:28 PM	Text Document
 AA_wiki_12	06-05-2024 08:28 PM	Text Document
 AA_wiki_13	06-05-2024 08:28 PM	Text Document

Valid

 AA_wiki_08	06-05-2024 08:28 PM	Text Document
 AA_wiki_09	06-05-2024 08:28 PM	Text Document
 AA_wiki_14	06-05-2024 08:28 PM	Text Document
 AA_wiki_16	06-05-2024 08:28 PM	Text Document
 AA_wiki_17	06-05-2024 08:28 PM	Text Document
 AA_wiki_21	06-05-2024 08:28 PM	Text Document
 AA_wiki_31	06-05-2024 08:28 PM	Text Document
 AA_wiki_38	06-05-2024 08:28 PM	Text Document
 AA_wiki_44	06-05-2024 08:28 PM	Text Document
 AA_wiki_50	06-05-2024 08:28 PM	Text Document
 AA_wiki_51	06-05-2024 08:28 PM	Text Document
 AA_wiki_57	06-05-2024 08:28 PM	Text Document
 AA_wiki_59	06-05-2024 08:28 PM	Text Document

Preprocessing



Before Preprocessing

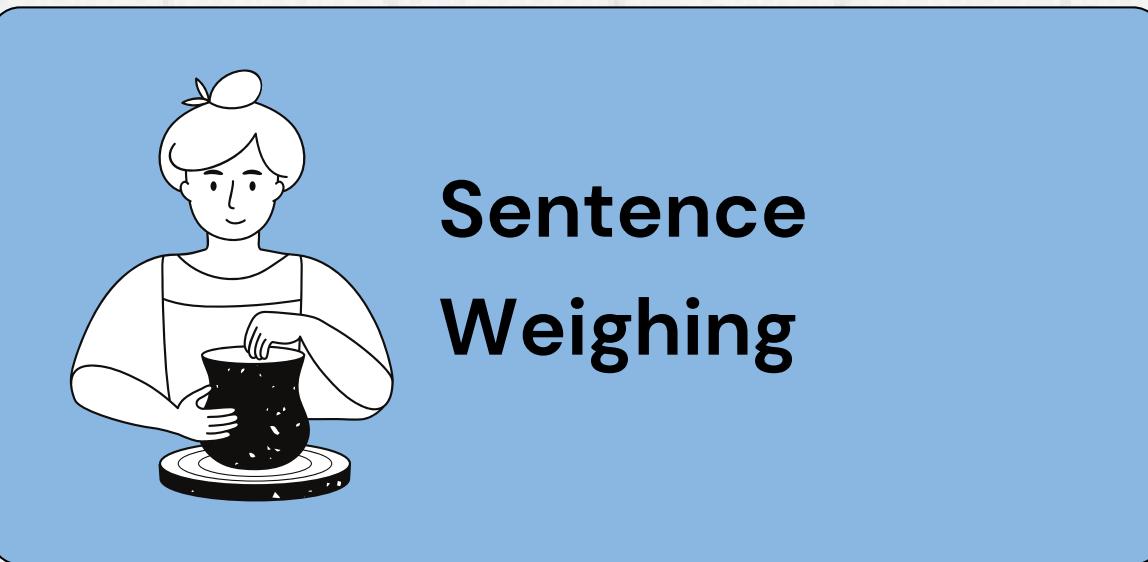
```
Number of docs after chunking : 156392
Number of docs with more than 448 tokens : 6968
Maximum number of words in single chunk : 1766
Mean Length: 109.68663358739578
```

After Preprocessing

```
Number of docs after chunking : 156392
Number of docs with more than 448 tokens : 0
Maximum number of words in single chunk : 448
Mean Length: 107.96570796460178
```



**Sentence
Scoring**



**Sentence
Weighing**

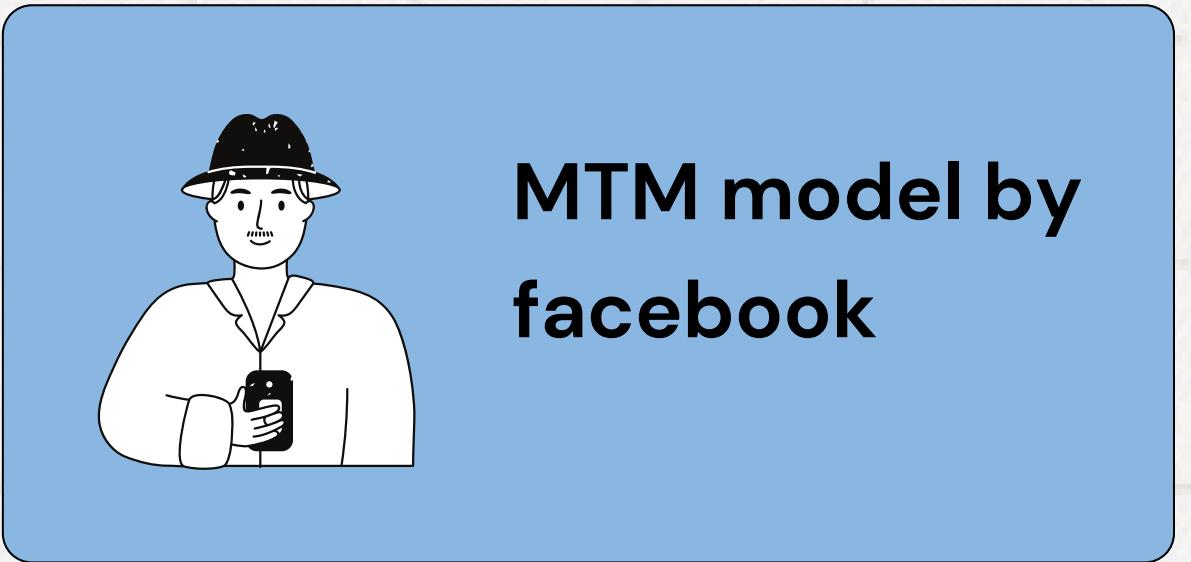


**Sentence
Clustering**

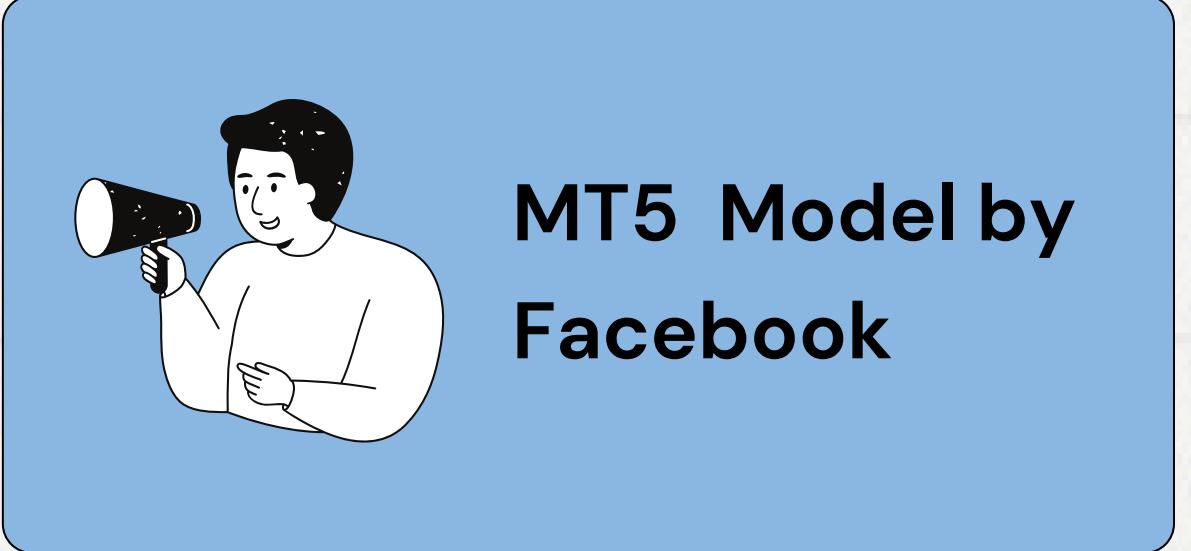
Approch

Extractive vs
Abstractive

Extractive



**MTM model by
facebook**



**MT5 Model by
Facebook**



mBART.

Abstractive

Sentence Scoring

The Sentence Scoring Method is a text summarization technique that evaluates the importance of each sentence within a document. It assigns scores to sentences based on various criteria, allowing for the selection of the most relevant sentences for summarization.

Methodology:

- Surface Score Components:
 - Position Score
 - Length Score
 - Paragraph Score
- Heading Score:

Implementation:

- Utilizes a combination of surface scores and heading scores to calculate the overall score for each sentence.
- Top-scoring sentences are selected to form the summary, ensuring the inclusion of key information.

Original : டேனியல் பெர்ல்டேனியல் பெர்ல் (அக்டோபர் 10, 1963 - பிப்ரவரி 1, 2002) அமெரிக்க யூதப் பத்திரிகையா

Summary : அவர் ரிச்சர்ட் ரெய்ட் ("கு பாம் வெடிப்பவர்"), அல்-கேடா மற்றும் பாகிஸ்தானின் உட்புற-சேவைகள் பு

Original : அதில் இருந்த செய்தியில் ஒரு பகுதி பின்வருமாறு:"நாங்கள் இன்னும் ஒரு நாள் அவகாசம் தருகிறோம்,

Summary : பின்னர் இந்தச் சுழற்சி தொடரும், மேலும் எந்த அமெரிக்கப் பத்திரிகையாளரும் பாகிஸ்தானில் நுழைய

Original : அந்த நேரத்தில் பெர்லின் தந்தை அவரது இஸ்ரேலியக் குடியுரிமை விசாரணையின் மீது ஏதேனும் தீய விடை

Summary : அவர்கள் ஜூலை 15, 2002 இல் குற்றவாளிகள் எனத் தீர்ப்பளிக்கப்பட்டனர், மேலும் ஷேக்குக்கு மரண தலைப்பில் நிறுவினார்கள்.

Original : HBO ஃபிலிம்ஸ் "த ஜர்னலிஸ்ட் அண்ட் த ஜிஹாடி: த மர்டர் ஆஃப் டேனியல் பெர்ல்" என்ற தலைப்பில் 79

Summary : அது அக்டோபர் 10, 2006 இல் HBO வில் ஓளிபரப்பானது. ஜூடியா பெர்ல் பின்னர் கலைஞர்கள், அரசாங்க

Original : அவரது இறப்புக்குப் பின்னர் விரைவில், பெர்லின் பெற்றோர் டேனியல் பெர்ல் ஃபவுண்டேசனை நிறுவினார்கள்.

Summary : அவரது இறப்புக்குப் பின்னர் விரைவில், பெர்லின் பெற்றோர் டேனியல் பெர்ல் ஃபவுண்டேசனை நிறுவினார்கள்.

```
class SentenceScoreCalculator:  
    def __init__(self, p, HEADING):  
        self.sCount = p.sCount  
        self.pCount = p.pCount  
        self.hWords = word_tokenize(HEADING)  
        self.counter = WordCounter(p.sCount).count(p.processed)  
        self.vocabulary = self.counter.wordDict.keys()  
        self.paragraphStructure = p.paragraphStructure  
        self.sentenceWords = p.processed  
  
    def score(self, index):  
        words = self.sentenceWords[index]  
        return self.surfaceScore(index, words) + self.headingScore(index,  
words)  
    def surfaceScore(self, index, words):  
        return (  
            self.positionScore(index, words) +  
            self.lengthScore(index, words) +  
            self.paragraphScore(index, words)  
        )  
  
    def paragraphScore(self, index, words):  
        return 1 - ( self.paragraphStructure[index] / self.pCount )  
  
    def headingScore(self, index, words):  
        if len(self.hWords) == 0:  
            return 0  
  
        wordsInHeading = len(filter(lambda w: w in self.hWords, words))  
        totalWords = ( math.log( len(words) ) + math.log( len(self.hWords) ) )  
  
        return float(wordsInHeading) / totalWords if totalWords > 0 else 0  
  
    def positionScore(self, index, words):  
        return 1 - ( index / self.sCount )  
  
    def lengthScore(self, index, words):  
        return len(words) / len(self.vocabulary)
```

Sentence Weighing

The Sentence Weighing Method is a technique used to evaluate the significance of sentences in a document for summarization purposes. It assigns weights to sentences based on their characteristics, such as word frequency and similarity to other sentences.

Methodology:

- Edit Distance Calculation:
- Calculates the Sentence Length Score Weight (LSW) based on the edit distance between sentences.
- Sentence Weight Calculation:
- Considers the frequency of words in each sentence relative to the total vocabulary size.

Implementation:

Combines the Sentence Length Score Weight (LSW) and sentence weight to rank sentences. Higher-ranked sentences are considered more important and are selected for the summary.

Original : அந்த நேரத்தில் பெர்லின் தந்தை அவரது இஸ்ரேலியக் குடியுரிமை விசாரணையின் மீது ஏதேனும் தீய விளைவுகளை ஏற்படுத்திவிடுமோ என்று பயந்தார். மார்ச் 21, 2002 இல், பாகிஸ்தானில், அகமது ஓமர் சாயித் சேக் மற்றும் மூன்று மற்ற சந்தேகத்திற்குரியவர்கள் டேனியல் பெர்லின் கடத்தல் மற்றும் கொலையில் அவர்களாது பங்களிப்புக்காகக் குற்றஞ்சாட்டப்பட்டனர். அவர்கள் ஜூலை 15, 2002 இல் குற்றவாளிகள் எனத் தீர்ப்பளிக்கப்பட்டனர், மேலும் ஷேக்குக்கு மரண தண்டனை விதிக்கப்பட்டது. ஷேக் அவரது மரணதண்டனைக்கு எதிராக மேல்முறையீடு செய்தார், ஆனால் அவரது வழக்கின் விசாரணை தொடர்ந்து 30 முறைகளுக்கும் மேலாக ஒத்தி வைக்கப்பட்டது, மேலும் தீர்மானமான தேதி எதுவும் நிர்ணயிக்கப்படவில்லை. மார்ச் 10, 2007 இல், ஓசாமா பின் லேடனின் கீழ் மூன்றாவது ஆணையிடுபவராகக் கூறப்படும் அல் கொய்தா செயலாளராகக் குற்றஞ்சாட்டப்படும் காலித் ஷேக் முகமது டேனியல் பெர்லின் கொலைக்கான அவரது போராளி நிலைத் திறனாய்வுத் தீர்ப்பாயத்திற்கு முன்பு, பொறுப்பேற்க வேண்டும் என வலியுறுத்தப்பட்டது. அவர் அவரைத் தலையைத் துண்டித்தார் என வலியுறுத்தப்பட்டது. அவரது தீர்ப்பாய விசாரணையின் போது ஒப்புதல் வாக்குமூலம் படிக்கையில், காலித் ஷேக் முகமதுவின் பதிவில் பின்வருவன் தொடர்ந்தது: நான் எனது ஆசிர்வாதம் பெற்ற வலது கரத்துடன் பாகிஸ்தான், கராச்சியில் அமெரிக்க யூத டேனியல் பெர்லின் தலையைத் துண்டித்தேன். இந்த ஒப்புதல் தொடர் வார்த்தை, அவரது இரகசியமான CIA விசாரணை மையத்தின் சர்ச்சைக்குரிய குறுக்கு விசாரணையில் இடம்பெற்ற வார்த்தைகளில் இருந்து 2002 ஆம் ஆண்டில் வெளிவந்தது. மார்ச் 19, 2007 இல், அகமது ஓமர் சாயித் ஷேக்கின் வழக்கறிஞர்கள், காலித் ஷேக் முகமதுவின் குறுக்கு விசாரணை அவர்களாது கட்சிக்காரருக்குச் சாதகமாக இருப்பதைக் கண்டனர். அவர்கள் பெர்லின் கொலையில் அவரது கட்சிக்காரர் பங்கு பெற்றிருக்கிறார் என்பதை எப்போதும் ஏற்றுக்கொள்கின்றனர், ஆனால் அவர்கள் எப்போதும் காலித் ஷேக் முகமதுதான் உண்மையான கொலைகாரர் என வாதிடுகின்றனர். அவர்கள் அவர்களாது கட்சிக்காரரின் மரண தண்டனையில் காலித் ஷேக் முகமதுவின் ஒப்புதலை மையமாக பங்களிக்க வைக்கத் திட்டமிடுகின்றனர். அப்போதைய-பாகிஸ்தான் ஜனாதிபதி பெர்வெஸ் முஷாரஃப் அவரது "இன் தலைஞ் ஆஃப் ஃபயர்" புத்தகத்தில் சில நேரத்தில் இரட்டை ஏஜன்டாக இருந்த M16 இன் ஏஜன்ட் மூலமாக பெர்ல் கொலை செய்யப்பட்டார் என்று குறிப்பிட்டிருந்தார். 2002 ஆம் ஆண்டில் பெர்லின் எழுத்துக்களின் தொகுப்பு ("அட்ஹோம் இன் த வேர்ல்ட்") அவர் இருந்த பின் வெளியிடப்பட்டது, அதில் அவரது "ஒரு எழுத்தாளராக அவரது செயற்கரிய திறன்" மற்றும் "த வால் ஸ்ட்ரீட் ஜர்னலின் "மையப் பத்தியில்" இடம்பெற்ற அவரது "திருப்பம் நிறைந்த கதைகளுக்கான பார்வை ஆகியவை விளக்கப்பட்டிருந்தது. " "அவற்றில் ஆறு கதைகள் இசைக் கலைஞர் ரஸ்ஸல் ஸ்டெயின்பர்க்கின் ஆல்பமான " " ஸ்டோரிஸ் ஃபரம் மை ஃபேவரிட் பிளான்ட்டில்" தழுவப்பட்டது, இது வயலின், பியானோ மற்றும் ரீடருக்கான முத்தொகுதி ஆகும்." பெர்லின் குடும்பம் மற்றும் நண்பர்களால் பெர்லின் குறிக்கோள் தொடர்வதற்காக "டேனியல் பெர்ல் ஃபவுண்டேசன்" உருவாக்கப்பட்டது, மேலும் அவர்கள் அவரது மரணத்திற்கு பெர்லின் பணி மற்றும் பண்பை வடிவமைத்த துணிவு, பாணி மற்றும் கொள்கைகள் ஆகியவற்றில் மூலகாரணம் என்னவாக இருக்கும் எனக் குறிப்பிட்டிருந்தனர். டேனியல் பெர்ல் உலக இசை நாட்கள் 2002 இல் இருந்து உலகம் முழுவதும் நடைபெற்று வருகின்றன, மேலும் 60 நாடுகளில் 1,500 க்கும் மேற்பட்ட இசை நிகழ்ச்சிகள் நடத்தப்பட்டுள்ளன. பெர்லின் விதவை மனைவி மரியன்னே பெர்ல் எ மைட்டி ஹார்ட் "எ மைட்டி ஹார்ட்" என்ற அவரது வாழ்க்கை நினைவுக்குறிப்பை எழுதினார், அதில் பெர்லின் முழுக்கதை மற்றும் அவரது வாழ்க்கையில் நிகழ்ந்த நிகழ்வுகள் இடம்பெற்றிருந்தது. அந்தப் புத்தகத்தைத் தழுவி ஏஞ்சலினா ஜூலை, இர்ப்பான் கான், ஆர்சீ பஞ்சாபி, வில் பேட்டோன் மற்றும் டேன் ஃபட்டர்மேன் ஆகியோர் நடிப்பில் திரைப்படமாக வெளியானது. செப்டம்பர் 1, 2003 ஆம் ஆண்டில், "ஊ கில்ட் டேனியல் பெர்ல்?" என்ற தலைப்பில் புத்தகம் வெளியிடப்பட்டது, இது பெர்னார்ட்-ஹென்றி லெவியால் எழுதப்பட்டது. எழுத்தாளர் "துப்பறியும் நாவலாக" எழுதியிருந்த அந்த புத்தகம், கொலை பற்றிய அதன் ஊகித்த முடிவுகள் சிலவற்றுக்காக, பாகிஸ்தானின் சில பாத்திரப்படைப்புகளுக்காக மற்றும் பெர்லின் இறுதி நிமிடங்களில் அவரது நினைவாக எழுத்தாளர் கற்பனையாக எழுதியிருந்த விசயங்களுக்காக சர்ச்சைக்குரியதாக இருந்தது. லெவி அந்த புத்தகத்திற்காக விமர்சிக்கப்பட்டார். அந்த புத்தகம் டோட் வில்லியம்ஸ் இயக்கத்தில் மற்றும் ஜோஷ் லூகாஸ் நடிப்பில் திரைப்படமாக எடுக்கப்பட்டு வருகிறது, அதில் டேனியல் பெர்லின் வாழக்கையின் இறுதி நாட்கள் இடம்பெறும்.

Summary : அவர்கள் ஜூலை 15, 2002 இல் குற்றவாளிகள் எனத் தீர்ப்பளிக்கப்பட்டனர், மேலும் ஷேக்குக்கு மரண தண்டனை விதிக்கப்பட்டது. "பெர்லின் குடும்பம் மற்றும் நண்பர்களால் பெர்லின் குறிக்கோள் தொடர்வதற்காக டேனியல் பெர்ல் ஃபவுண்டேசன்" உருவாக்கப்பட்டது, மேலும் அவர்கள் அவரது மரணத்திற்கு பெர்லின் பணி மற்றும் பண்பை வடிவமைத்த துணிவு, பாணி மற்றும் கொள்கைகள் ஆகியவற்றில் மூலகாரணம் என்னவாக இருக்கும் எனக் குறிப்பிட்டிருந்தனர். என்ற தலைப்பில் புத்தகம் வெளியிடப்பட்டது, இது பெர்னார்ட்-ஹென்றி லெவியால் எழுதப்பட்டது. எழுத்தாளர் "துப்பறியும் நாவலாக" எழுதியிருந்த அந்த புத்தகம், கொலை பற்றிய அதன் ஊகித்த முடிவுகள் சிலவற்றுக்காக, பாகிஸ்தானின் சில பாத்திரப்படைப்புகளுக்காக மற்றும் பெர்லின் இறுதி நிமிடங்களில் அவரது நினைவாக எழுத்தாளர் கற்பனையாக சர்ச்சைக்குரியதாக இருந்தது. லெவி அந்த புத்தகம் டோட் வில்லியம்ஸ் இயக்கத்தில் மற்றும் ஜோஷ் லூகாஸ் நடிப்பில் திரைப்படமாக எடுக்கப்பட்டு வருகிறது, அதில் டேனியல் பெர்லின் வாழக்கையின் இறுதி நாட்கள் இடம்பெறும்.

```
class SentenceScoreCalculator:  
    def __init__(self, p):  
        self.sCount = p.sCount  
        self.counter = WordCounter(p.sCount).count(p.processed)  
        self.vocabulary = self.counter.wordDict.keys()  
        self.nWords = len(self.vocabulary)  
        self.sentences = p.sentences  
        self.calcuuateEditDistance()  
  
    def calcuateEditDistance(self):  
        self.lsw = { }  
  
        for s in range(self.sCount):  
            maxLen = lambda s2: max(len(self.sentences[s]), len(self.sentences[s2]))  
            ed = lambda s2: editDistance(self.sentences[s], self.sentences[s2])  
            lsw = lambda s2: float(maxLen(s2) - ed(s2)) / maxLen(s2)  
  
            self.lsw[s] = sum(map(lsw, range(self.sCount)))  
  
    def sentenceWeight(self, index):  
        words = self.counter.wordsIn(index)  
        additionalOccurrences = lambda w: ( self.counter.fetchWordCount(w) -  
self.counter.fetchSentenceWordCount(index, w) )  
  
        return reduce(lambda s, w: additionalOccurrences(w) / self.nWords, words, 0)  
  
    def rank(self, index):  
        return(self.sentenceWeight(index) + self.lsw[index])
```

Sentence Clustering

The Sentence Clustering Method groups sentences into clusters based on their feature representations. It selects representative sentences from each cluster to form a coherent summary of the document.

Methodology:

- **Feature Representation:**
 - 1.Extracts features representing the importance of each term (word) in each sentence.
- **Clustering:**
 2. Uses the KMeans algorithm to partition sentences into clusters based on their feature representations.
 - 3.Sentence Selection:
 - 4.Selects representative sentences from each cluster based on their proximity to the cluster centroid.

Implementation:

Features are normalized to ensure equal contribution to clustering.

Representative sentences are selected to provide a comprehensive overview of the document's content.

```

def termWeightFN(s, counter, sCount, TYPE):
    sentencesWithWord = lambda w: filter(lambda sd: counter.isWordIn(sd, w),
                                          counter.sentenceDict)

    bool_ = lambda w: 1 if counter.isWordIn(s, w) else 0
    tF = lambda w: counter.fetchSentenceWordCount(s, w) if counter.isWordIn(s, w) else 0
    idF = lambda w: math.log( float(sCount) / len(sentencesWithWord(w)) )
                           if counter.isWordIn(s, w) else 0
    tFIdf= lambda w: tF(w) * idF(w)

    # BOOL
    if TYPE == 1:
        return (bool_)

    # TF
    elif TYPE == 2:
        return (tF)

    # IDF
    elif TYPE == 3:
        return (idF)

    # TF-IDF
    elif TYPE == 4:
        return (tFIdf)

def normalizeFeatures(features):
    noramlized = [ ]
    for f in features:
        noramlized.append(map(lambda ft: float(ft) / max(f) if max(f) > 0 else 0, f))
    return noramlized

def buildFeatures(preprocessed, counter, TYPE):
    vocabulary = counter.vocabulary()

    features = [ ]

    for s in range(preprocessed.sCount):
        sFeatures = map(termWeightFN(s, counter, preprocessed.sCount, TYPE), vocabulary)
        features.append(sFeatures)

    return normalizeFeatures(features)

def sentenceSelect(kmeans, preprocessed, features):
    summary = [ ]

    for i in set(kmeans.labels_):
        sentencesInCluster = reduce(lambda m,l: m + [l] if kmeans.labels_[l] == i else m,
                                    range(preprocessed.sCount), [ ])

        sortedByClosenessToCentroid = sorted(sentencesInCluster, key=lambda s:
euclideanDistance(kmeans.cluster_centers_[i], features[s]))

        summary.append(sortedByClosenessToCentroid[0])

    return summary

```

```

def buildFeatures(preprocessed, counter, TYPE):
    vocabulary = counter.vocabulary()

    features = [ ]

    for s in range(preprocessed.sCount):
        sFeatures = map(termWeightFN(s, counter, preprocessed.sCount, TYPE), vocabulary)
        features.append(sFeatures)

    return normalizeFeatures(features)

def sentenceSelect(kmeans, preprocessed, features):
    summary = [ ]

    for i in set(kmeans.labels_):
        sentencesInCluster = reduce(lambda m,l: m + [l] if kmeans.labels_[l] == i else m,
                                    range(preprocessed.sCount), [ ])

        sortedByClosenessToCentroid = sorted(sentencesInCluster, key=lambda s:
euclideanDistance(kmeans.cluster_centers_[i], features[s]))

        summary.append(sortedByClosenessToCentroid[0])

    return summary

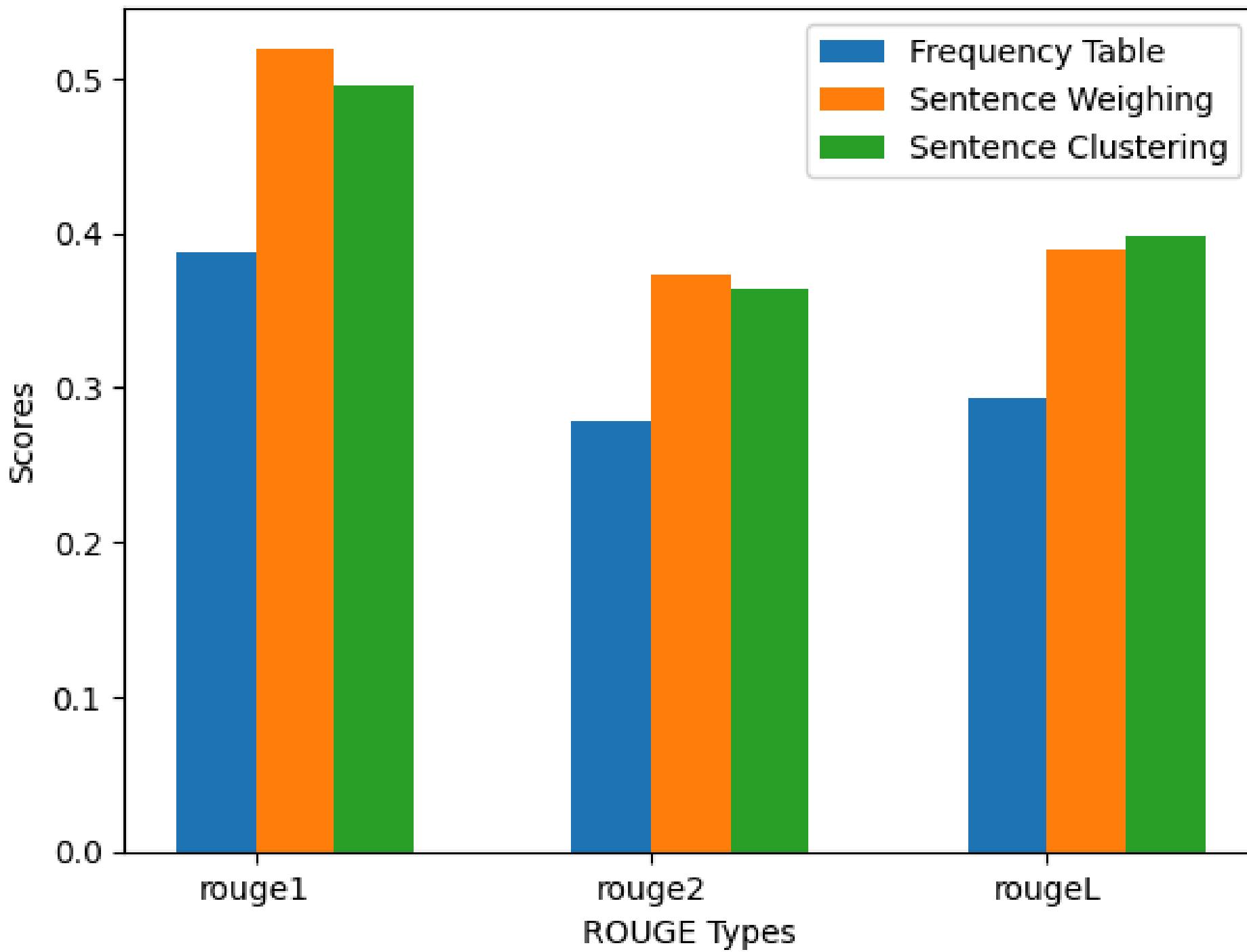
def generateSummary(PATH, NGRAMS, CLUSTERS, TYPE):
    # Preprocessing and Term Selection
    p = Preprocessor(PATH, NGRAMS).parse()
    counter = WordCounter(p.sCount).count(p.processed)
    # Term weighting and feature selection
    features = buildFeatures(p, counter, TYPE)
    # Sentence clustering
    kmeans = KMeans(n_clusters=CLUSTERS,
                     n_jobs=4,
                     precompute_distances=True,
                     init="k-means++",
                     random_state=500,
                     tol=1e-10,
                     copy_x=True)
    kmeans.fit(features)
    # Sentence selection
    summary = sentenceSelect(kmeans, p, features)
    return map(lambda s: p.sentences[s], sorted(summary))

```

பாண்டவர்பாண்டவர் எனப்படுவர்கள் மகாபாரதத்தில் வரும் மன்னன் பாண்டுவின் ஜந்து மகன்கள் ஆவார்கள். இவர்களுள் முதல் மூவரான தர்மன், பீமன் மாற்றும் அர்ஜனன் ஆகியோர் குந்தி மூலமும் கடைசி இருவரான நகுலன் மற்றும் சகாதேவன் ஆகியோர் மாத்ரி மூலமும் பிறந்தவர்கள் ஆவர். இவர்கள் ஜீவர் எனபதால் பஞ்ச பாண்டவர் என்றும் அழைக்கப்படுவர். இவர்களுக்கும், இவர்கள் பீரியப்பா திருதாஷ்டிரனின் மகன்களான கீகளரவர்களுக்கும் நடந்த போரான குருட்சேததிரப் போரே மகாபாரதத்தின் முக்கிய நிகழ்வாகும். யமுனை நதிக்கரையில் யாதுவ குழு ஒன்று செழிப்பான மதுரா எனும் நகரை அமைத்து குழு அடசி முறையை நடத்தி வந்தது. யாதுவர் ஆட்சிக் குழுவில் ஒருவரான சூரசேனரின் மகள் பிரதை (பிருதை, பிரதை), பிரதையை குந்தி நாட்டு மன்னர் குந்தி போஜன் தக்தெடுத்து குந்தி எனப் பெயரிட்டு வளர்த்து வந்தார். மன வயதையடைந்த குந்திக்கு சுயம்வரம் நடந்தது, சுயம்வரத்தில் கூடியிருந்தவர்களில் பாண்டுவை தேர்நடெடுத்தாள். பீஷ்மர், இரண்டாவதாக மத்திர நாட்டின் மன்னன் சலயனினை சகோதரி மாதுரியை பாண்டுவிற்கு மனம் முடித்து வைக்க விரும்பினார். சல்லியனுக்கு, அவரது தங்கையின் நிச்சயத்திற்கு மனியும், முத்தும், பவளமும் சீராகக் கூடுதார் பீஷ்மர், அவற்றை ஏற்றுக்கொண்டு மாதரியை பாண்டுவிற்கு மனம் முடித்துத் தந்தார் சலயரபல நாடுகளை வெற்றி கொண்டு கப்பத்தொகையைப் பெற்று வந்த பின் குந்தியாலும் மாத்ரியாலும் தூண்டப்பட்டு வனவாசத்தை நாடிச் சென்றார் பாண்டு. வேட்டையின் போது பாண்டுவின் அம்பு பெண்மானை முயங்கிக் கொண்டிருந்த ஆண் மானை தாக்கிவிடுகிறது. மானின் அருகில் சென்று பார்த்த போது பாண்டுவுக்கு உண்மை தெரிகிறது. கிண்டமா என்ற முனிவரும் அவரது மனைவியும் காட்டில் சுதந்திரமாக உலவி காதல் செய்யும் நோக்கில் தங்களது தவ வலிமையால் மான்களாக உருவும் மாறியிருந்தனர். இறக்கும் நேரத்தில் கிண்டமா முனிவர் "ஓரு ஆணும் பெண்ணும் காதல் புரிவதை ஆக்ரோசமாக தடுத்துவிட்டாய் உனக்கு காதல் சுகம் என்ன என்பது தெரியாமல் போகக் கடவுது எந்த பெண்ணையும் காதல்கொண்டு தொட்டால் உடனே இறந்து போவாய்" என சாபமிட்டார். ஒரு குழந்தைக்கு தகப்பன் ஆக முடியாதவன் அரசன் ஆகமுடியாது என வருந்தி பாண்டு அத்தினாபுரம் செல்ல மறுக்கு சதஸ்ரங்க வனத்தில் முனிவரகளுடன் தங்கிவிடுகிறான். இசெசய்தி அத்தினாபுரம் எடுகிறது. பாண்டு இல்லாத நிலையில் அத்தினாபுரத்தின் ஆட்சியை பீஷ்மர் திருதாட்டிரனுக்கு வழங்குகிறார். சில மாதங்களில் காந்தாரி கருதகரித்தாள் என்ற செய்தி பாண்டுவுக்கு தெரியவே ஆட்சியும் போய், ஒரு குழந்தைக்கு தந்தையும் ஆகமுடியாத நிலையில் மனமுத்தமும், சோர்வும், விரகதியும் அடைந்து பாண்டு ஒரு முடிவெடுத்தான். சுவேதகேகு முனிவரின் நியதிப்படி ஒரு பெண்ணின் கணவர் அவர் விரும்பும் ஒரு ஆணுடன் சேர்ந்து குழந்தை பெற்றுக் கொள்ளலாம், அதன்படி தன்னுடன் இருந்த குந்தியை அழைத்து, யாராவது ஒரு முனிவரின் மூலமாக ஒரு குழந்தையை பெற்றுக்கொள் என வேண்டினான். தேவர்களையே அழைக்க முடியும் போது என? முனிவர்களை அழைக்க வேண்டும் என கூறி, தர்மத்தின் தலைவன் யமன் மூலம் யுதிஷ்டிரன் (தர்மன்), மிகுந்த சக்தி படைத்த வாயு பகவான் மூலம் பீமன், தேவர்களின் தலைவனான இந்திரன் மூலம் அருச்சனன், என மூன்று குழந்தைகளை குந்தி பெற்றாள். பாண்டு வேறு ஒரு தேவனை அழைக்க சொன்ன போது " மாட்டேன் மூவருடன் இருந்தாயிற்று நான்காவதாக ஒருவருடன் இருந்தால் என்னை வேசி என்று பெசுவார்கள் அப்படித்தான் தரம் சொல்கிறது" என மறுத்துவிடுகிறாள். "நீ வேறு எந்த ஆணிடமும் செல்ல முடியாது" எனபதால் மாத்ரிக்காக ஒரு கேவனை அழைக்கச் சொன்னான். பாண்டு வேறு ஒரு தேவனை அழைக்கச் சொன்னாள். அஸவினி தேவர்கள் எனும் இரட்டையர்கள் மூலம் உலகத்திலேயே மிக அழகான நகுலனும், உலகத்திலேயே எல்லாம் அறிந்த அறிவாளியான சகாதேவனும் பிறந்தார்கள். இப்படியாக பிறந்தவர்களை பாண்டவர்கள் என்று அத்தினாபுரத்து மக்கள் அழைத்தனர்.

முனிவர்களை அழைக்க வேண்டும் என கூறி, தர்மத்தின் தலைவன் யமன் மூலம் யுதிஷ்டிரன் (தர்மன்), மிகுந்த சக்தி படைத்த வாயு பகவான் மூலம் பீமன், தேவர்களின் தலைவனான இந்திரன் மூலம் அருச்சனன், என மூன்று குழந்தைகளை குந்தி பெற்றாள். "நீ வேறு எந்த ஆணிடமும் செல்ல முடியாது" எனபதால் மாத்ரிக்காக ஒரு கேவனை அழைக்கச் சொன்னான். பாண்டு வேறு ஒரு தேவனை அழைக்கச் சொன்ன போது " மாட்டேன் மூவருடன் இருந்தாயிற்று நான்காவதாக ஒருவருடன் இருந்தால் என்னை வேசி என்று பெசுவார்கள் அப்படித்தான் தரம் சொல்கிறது" என மறுத்துவிடுகிறாள்.

Comparison of Summarization Methods



Advantages

Sentence Scoring {

- Provides a structured approach to sentence selection.
- Considers both the document structure and the relevance to the heading.
- Allows for customizable scoring criteria based on specific requirements.

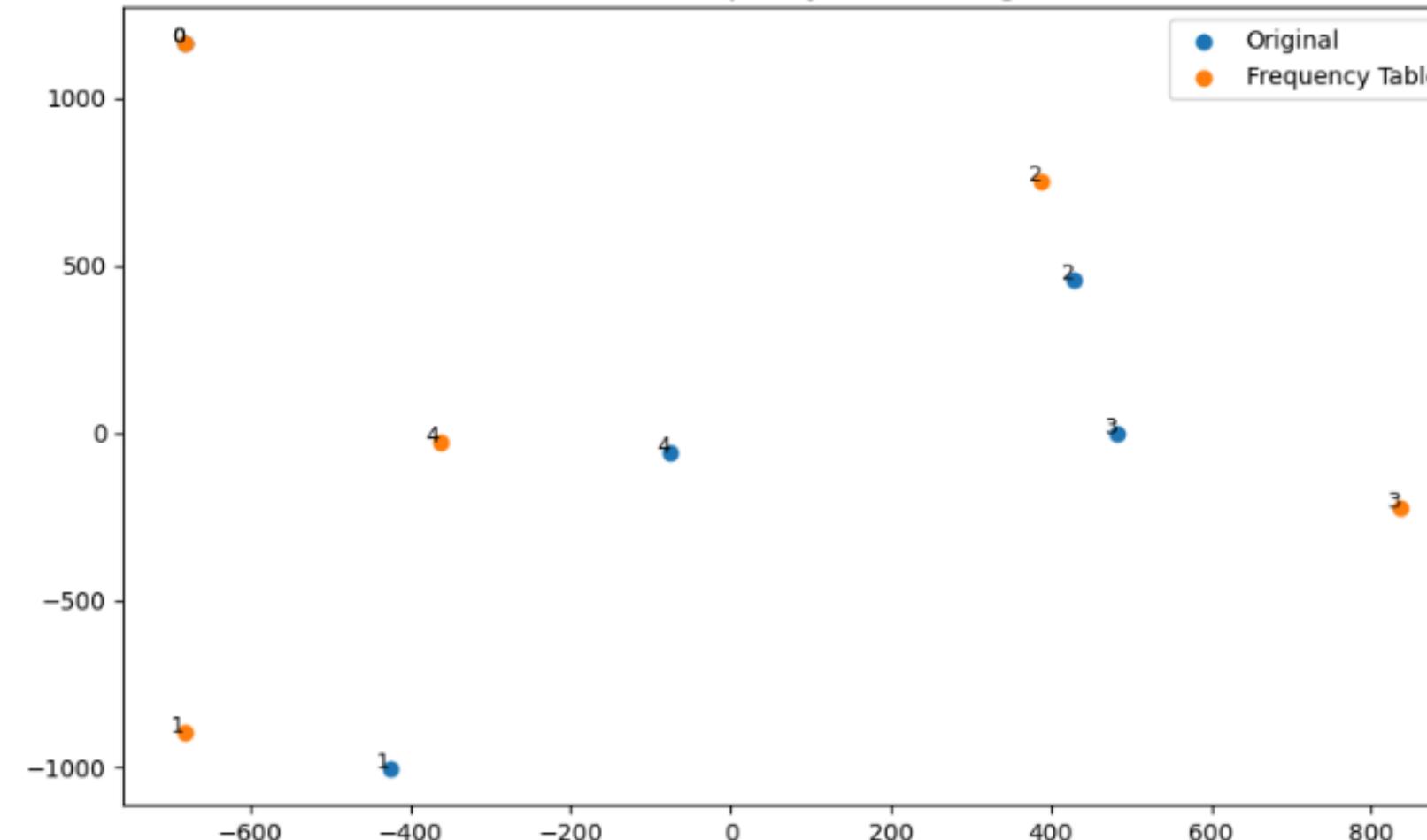
Sentence Weighing {

- Captures both structural similarity and word frequency in sentence evaluation.
- Provides a quantitative measure of sentence importance.
- Offers flexibility in choosing different weighting schemes based on requirements.

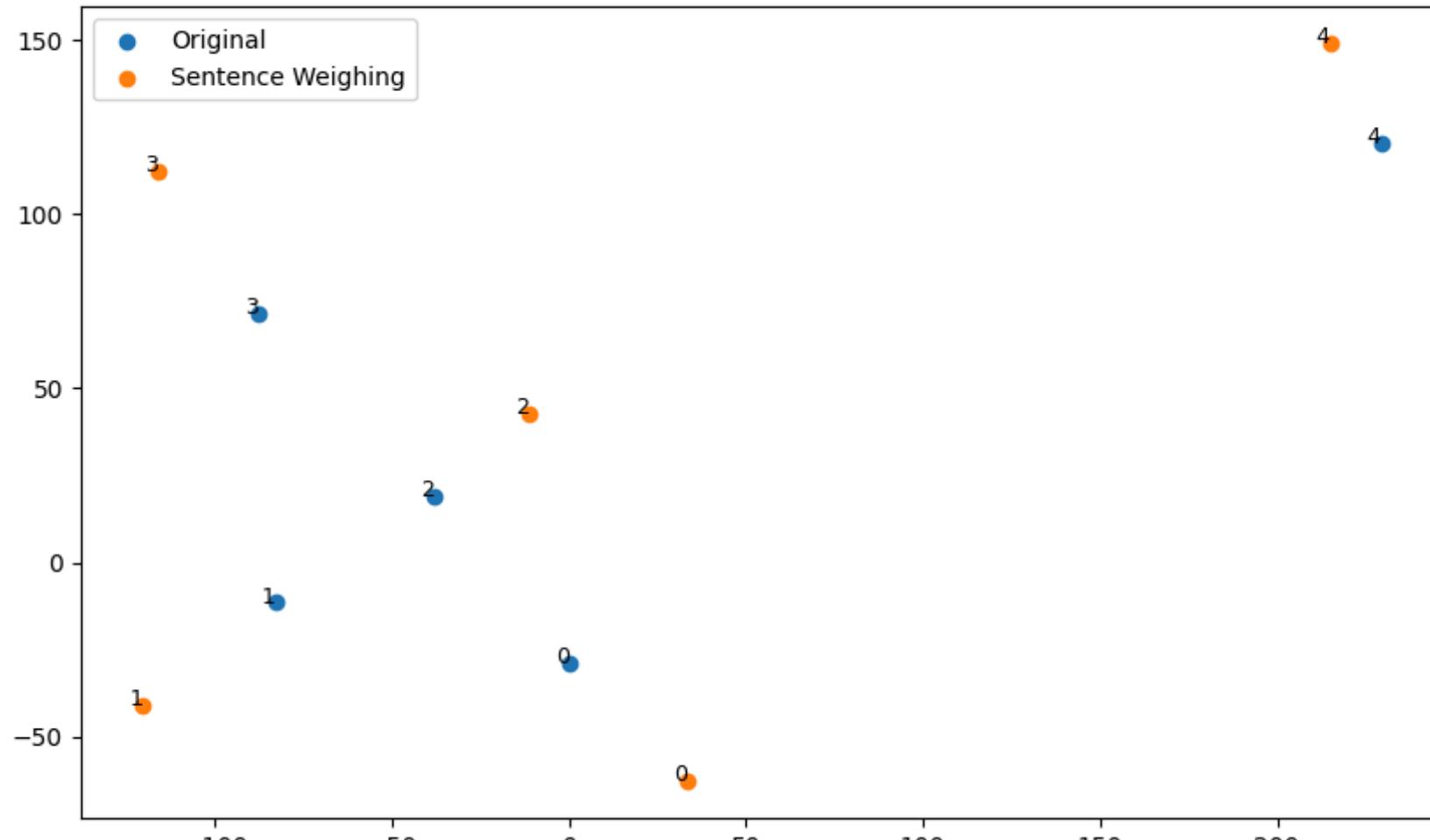
Sentence Clustering {

- Identifies key themes or topics within the document.
- Allows for the selection of diverse and representative sentences.
- Provides a structured and coherent summary based on clustering.

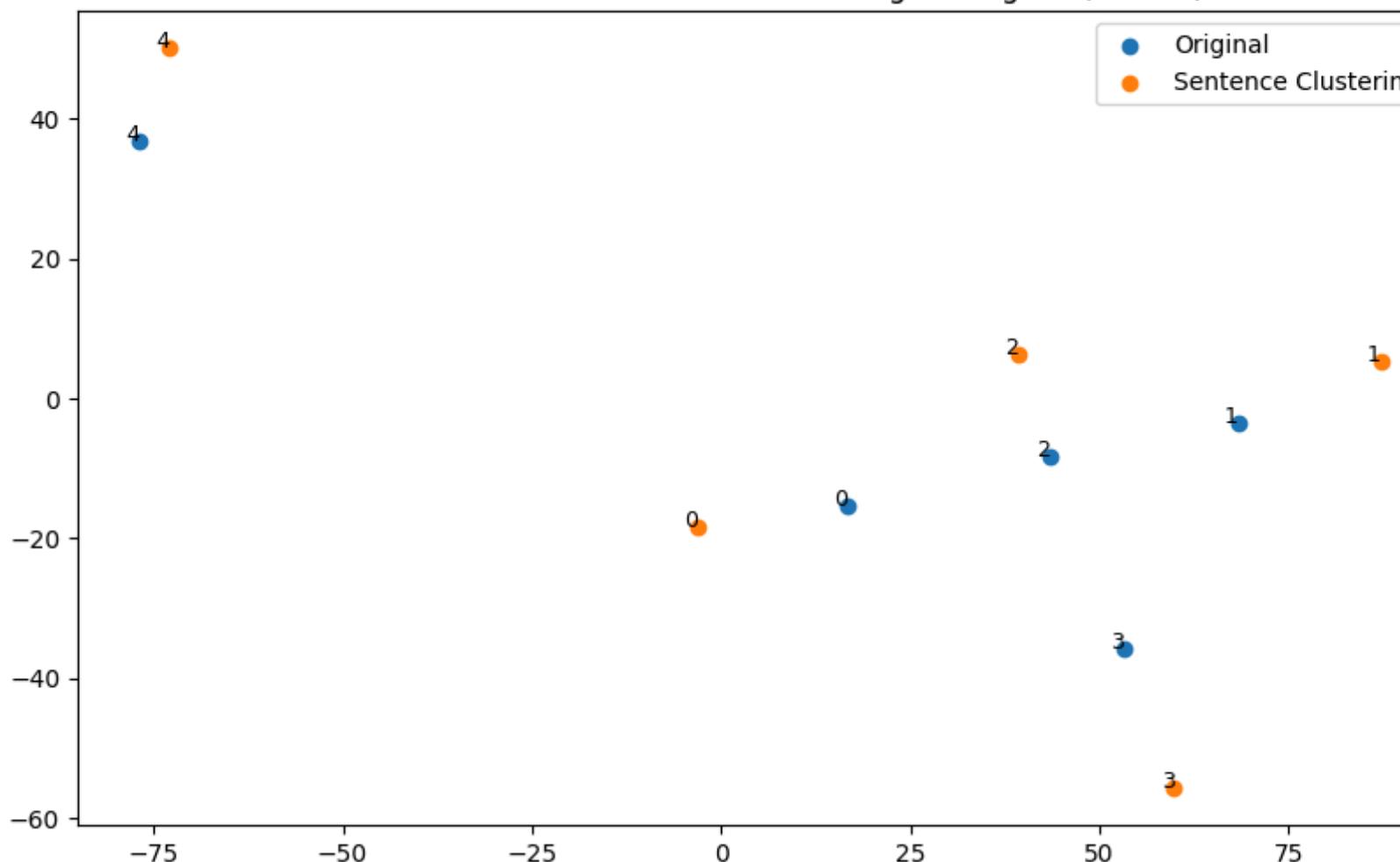
t-SNE Visualization: Frequency Table vs Original (Subset)



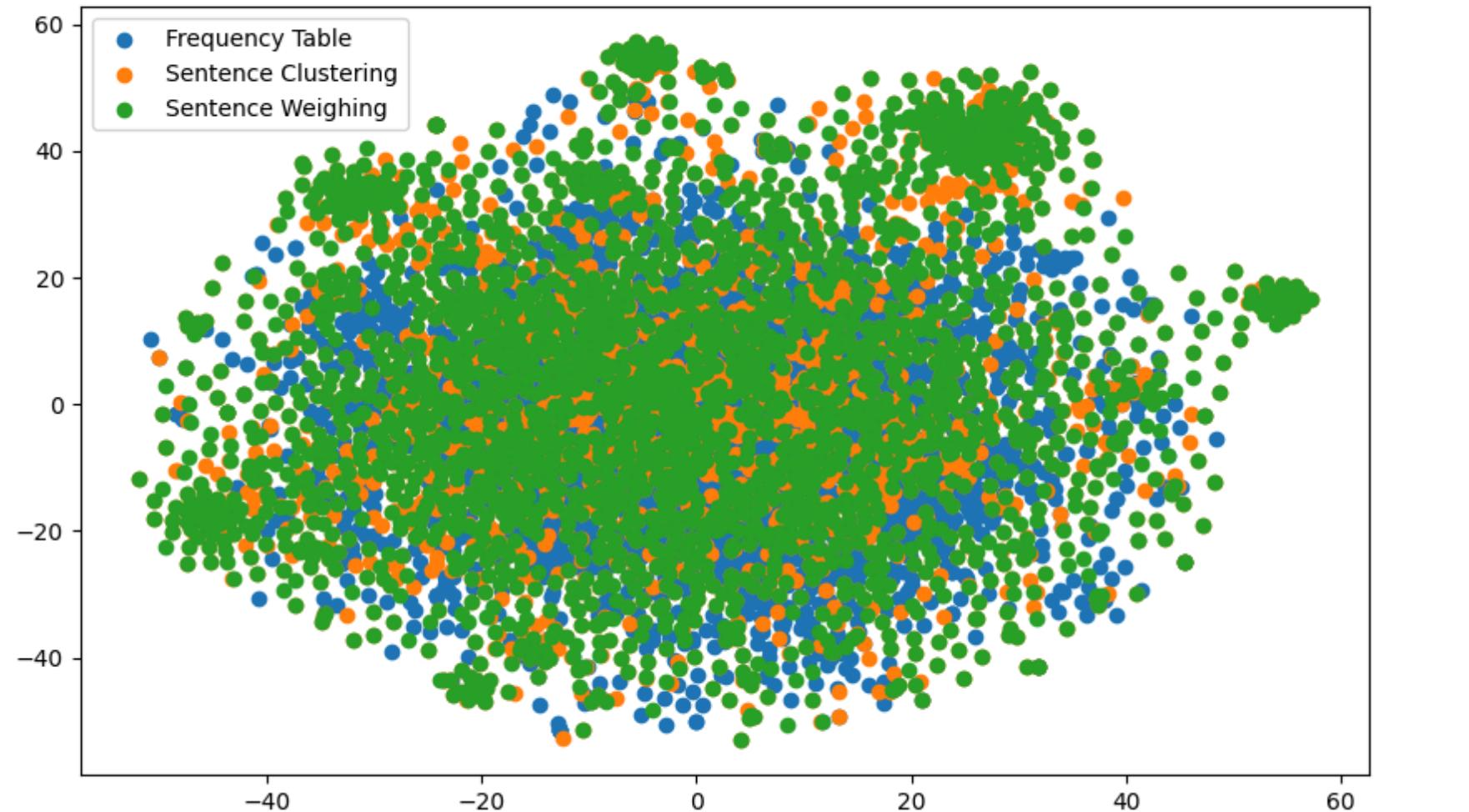
t-SNE Visualization: Sentence Weighing vs Original (Subset)



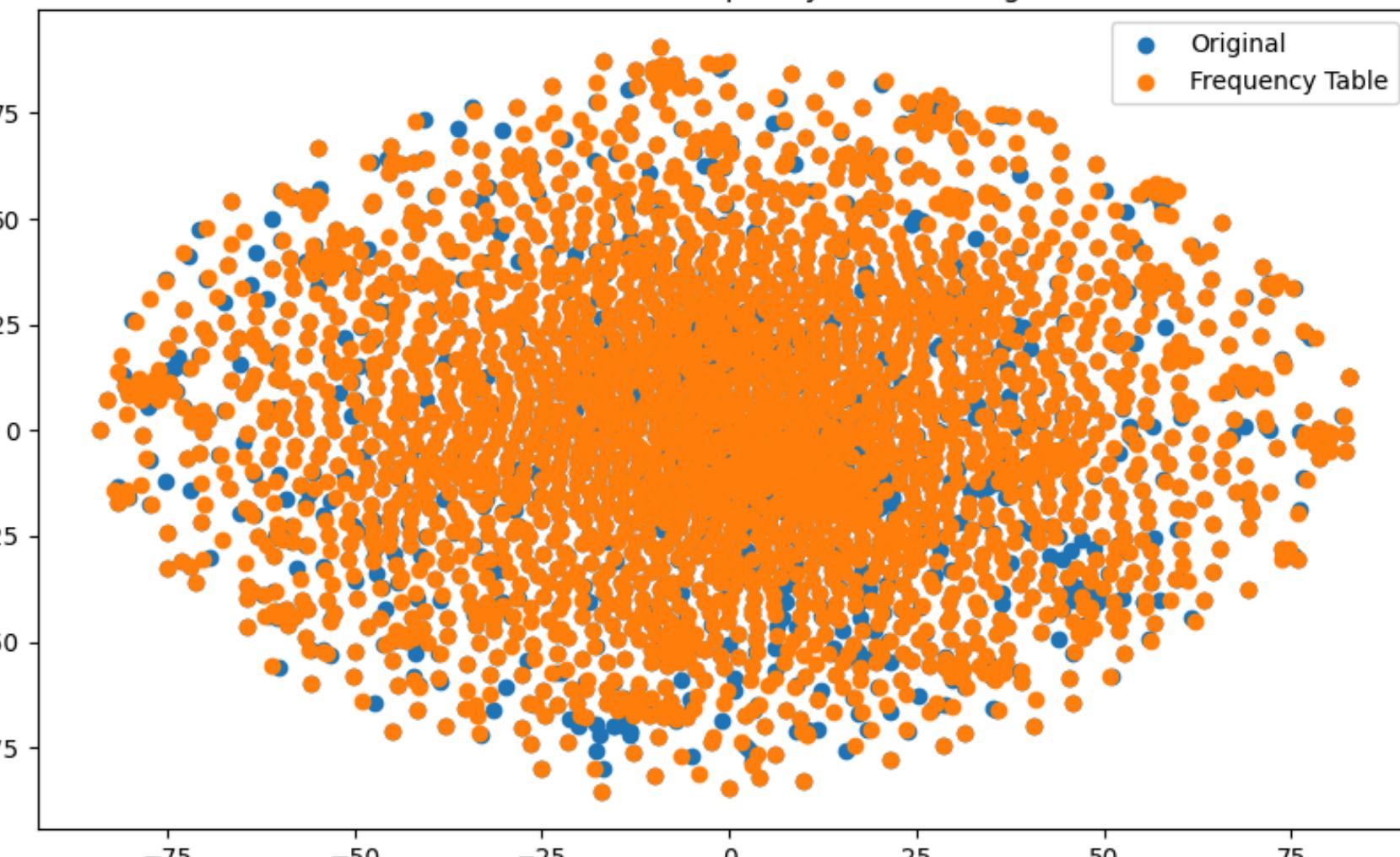
t-SNE Visualization: Sentence Clustering vs Original (Subset)



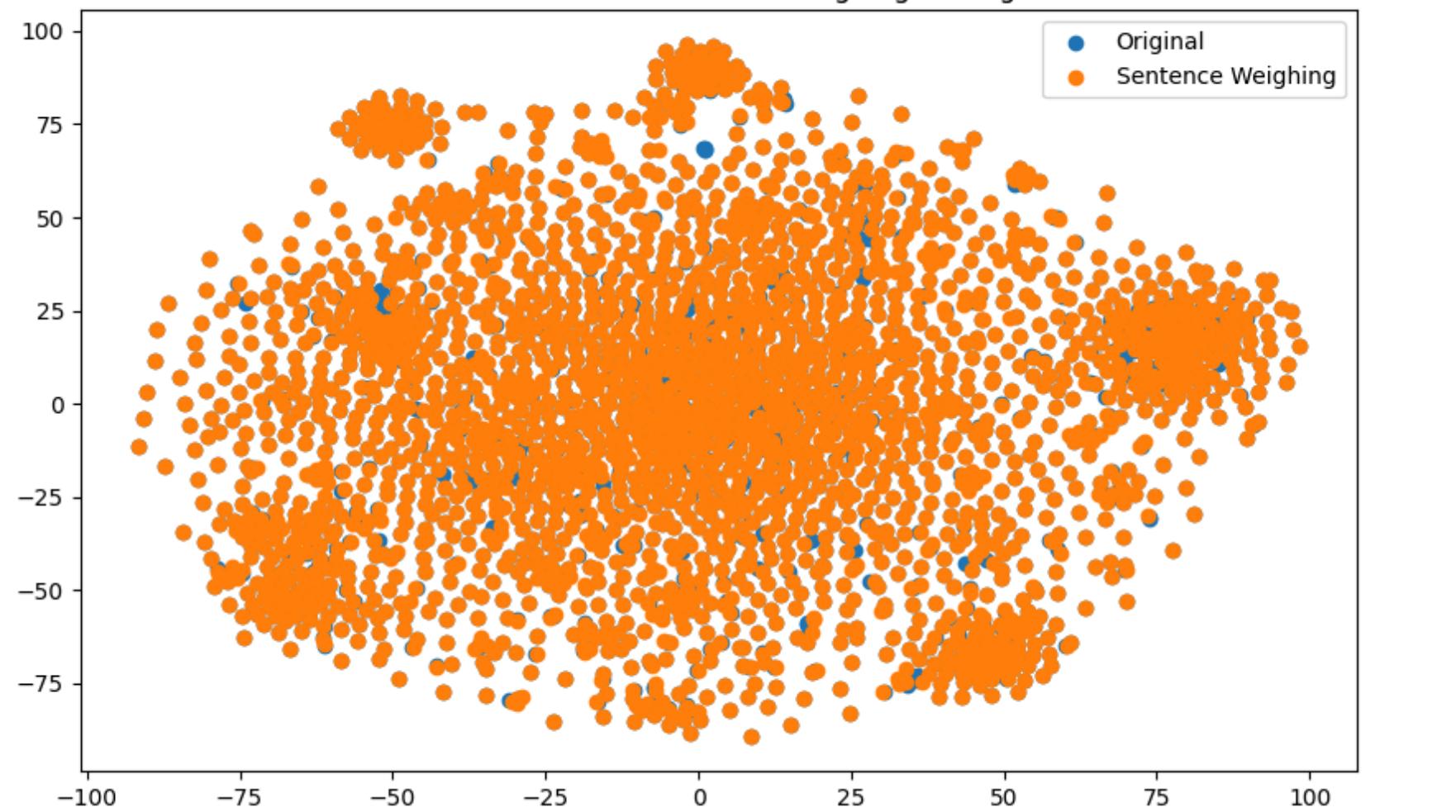
t-SNE Visualization of Summaries



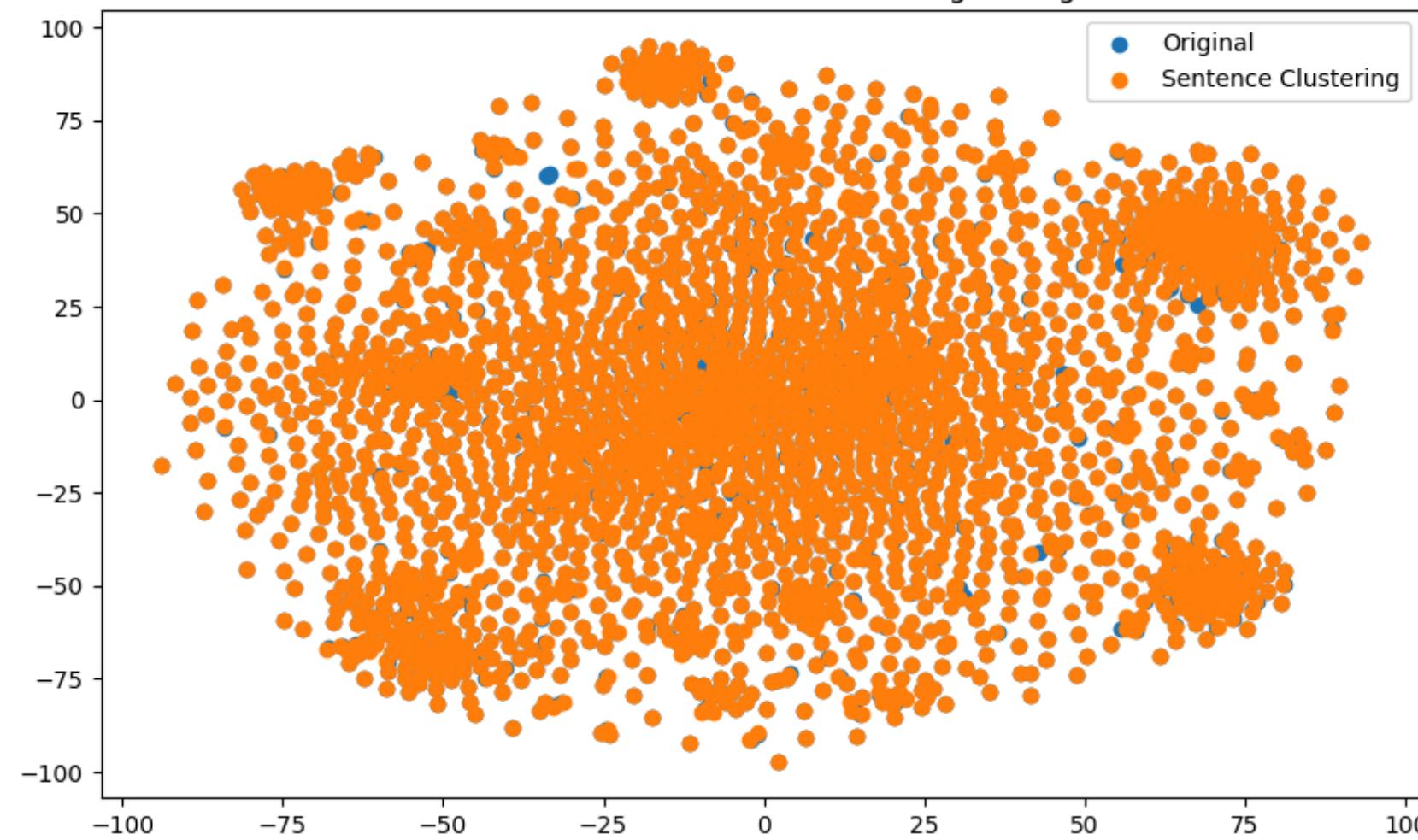
t-SNE Visualization: Frequency Table vs Original



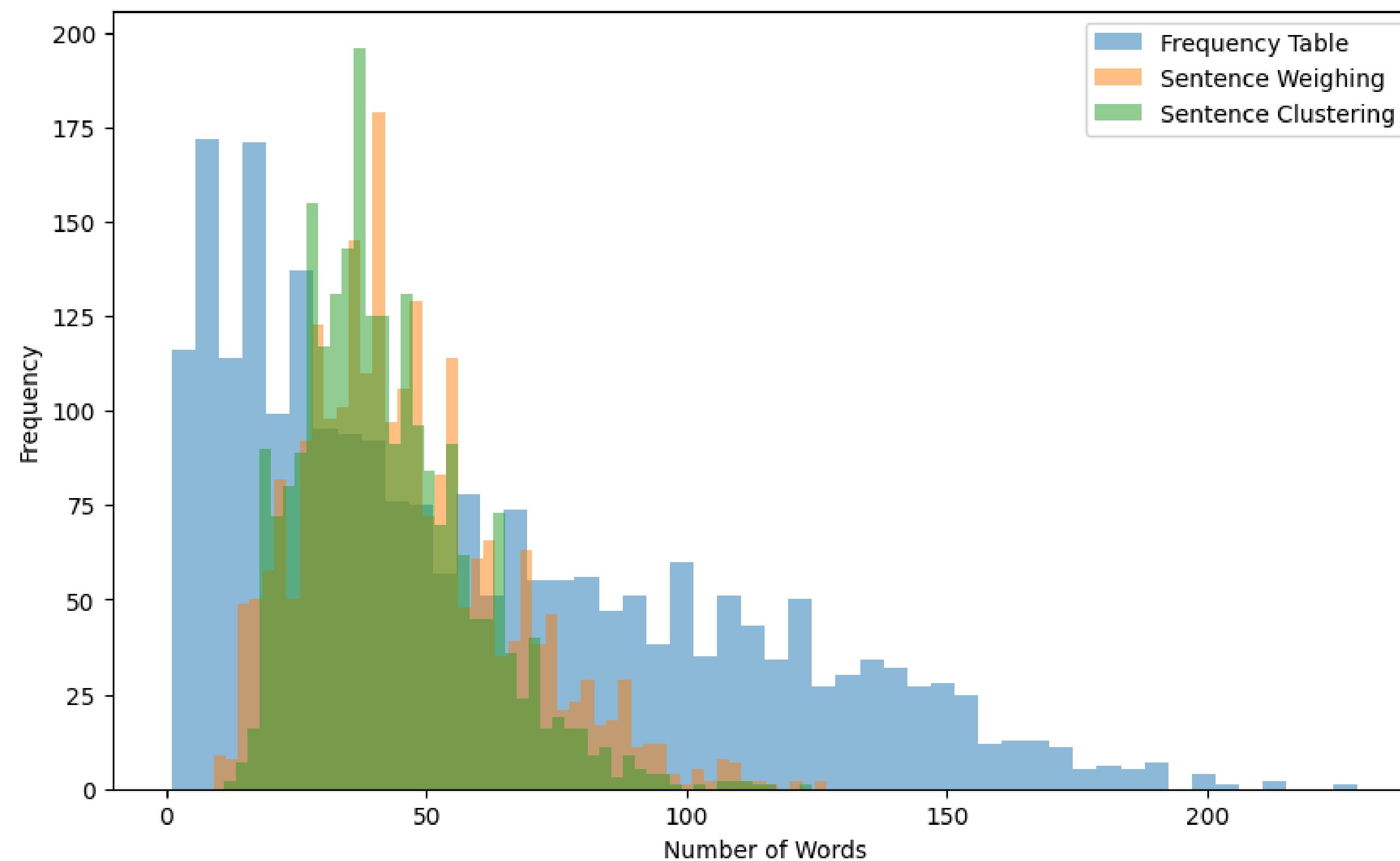
t-SNE Visualization: Sentence Weighing vs Original



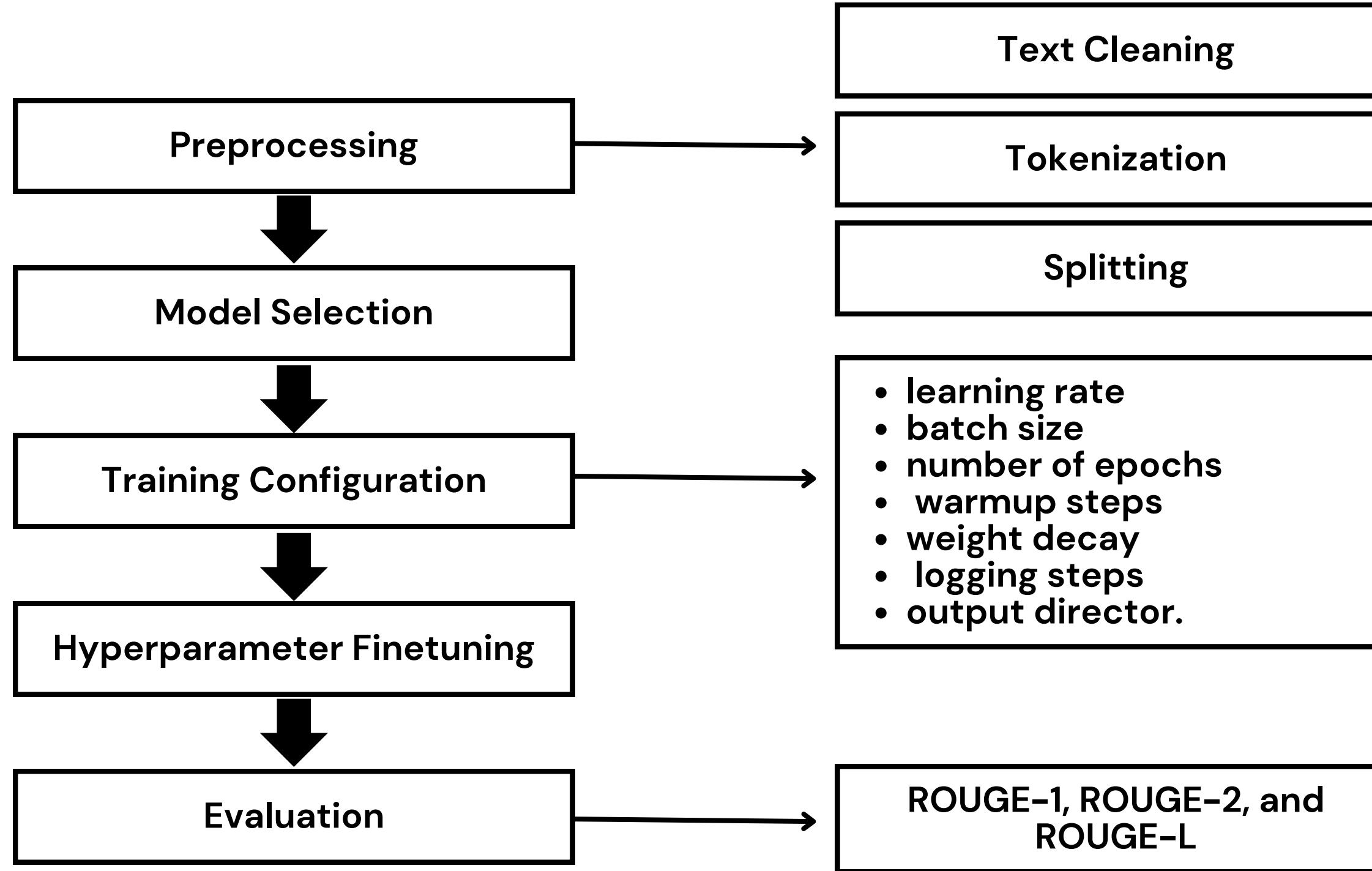
t-SNE Visualization: Sentence Clustering vs Original



Distribution of Number of Words in Summaries



Finetuning mBART & mT5



Abstractive Results

mBART

இது குறித்து அவர் பிபிசி தமிழிடம் கூறுகையில், "இத்தீர்ப்பை மிகச் சிறந்த முற்போக்கான தீர்ப்பாக பார்க்கிறேன்.

அடிப்படை உரிமை என்ன என்பதை மிகவும் தீவிரமாக இத்தீர்ப்பு விளக்கியுள்ளது" என்றார்.

"இந்திய அரசியலமைப்பின் 21-ஆவது விதியை மிகவும் ஆழமாக நீதிமன்றம் விளக்கியுள்ளது என்றும்,

ஏற்கனவே இரு வேறு வழக்குகளில் தனி நபர் அந்தரங்கத்தை அடிப்படை உரிமை பாதுகாக்காது எனக் குறிப்பிட்ட தீர்ப்புகளைத் திருத்தி அந்த உரிமையை தற்போது உச்ச நீதிமன்றம் பாதுகாத்துள்ளது" என்று என்.ராம் கூறினார்.

"ஆதார் பதிவு விவகாரத்தில் இந்த தீர்ப்பு நிச்சயமாக பிரதிபலிக்கும் என்று கூறும் அவர், ஆதார் முறையைத் திணிக்க முயற்சிக்கும் மத்திய அரசின் எண்ணம் இனி கடினமாக இருக்கும்" என்றார். "நெருக்கடி காலத்தில் நீதிபதி எச்.ஆர். கண்ணா அளித்த தீர்ப்பு ஏற்படுத்திய மாற்றத்தைப் போல இந்தத் தீர்ப்பும் சமூகத்தில் மாற்றத்தை ஏற்படுத்தலாம் என்று சிலர் கருதுவதாகவும், மொத்தத்தில் இது ஒரு முக்கியத்துவம் நிறைந்த தீர்ப்பாகும்"

என்றும் என்.ராம் தெரிவித்தார். பிற செய்திகள் : சமூக ஊடகங்களில் பிபிசி தமிழ்

Summarized Tamil Text: "இந்திய அரசியலமைப்பின் 21-ஆவது விதியை மிகவும் ஆழமாக நீதிமன்றம் விளக்கியுள்ளது என்றும், ஏற்கனவே இரண்டு வேறு வழக்குகளில் தனி நபர் அந்தரங்கத்தை அடிப்படை உரிமை பாதுகாக்காது எனக் குறிப்பிட்ட தீர்ப்புகளைத் திருத்தி அந்த உரிமையை தற்போது உச்ச நீதிமன்றம் பாதுகாத்துள்ளது" என்று என்.

Abstractive Results

M2M100

இது குறித்து அவர் பிபிசி தமிழிடம் கூறுகையில், "இத்தீர்ப்பை மிகச் சிறந்த முற்போக்கான தீர்ப்பாக பார்க்கிறேன்.

அடிப்படை உரிமை என்ன என்பதை மிகவும் தீவிரமாக இத்தீர்ப்பு விளக்கியுள்ளது" என்றார்.

"இந்திய அரசியலமைப்பின் 21-ஆவது விதியை மிகவும் ஆழமாக நீதிமன்றம் விளக்கியுள்ளது என்றும்,

ஏற்கனவே இரு வேறு வழக்குகளில் தனி நபர் அந்தரங்கத்தை அடிப்படை உரிமை பாதுகாக்காது எனக் குறிப்பிட்ட தீர்ப்புகளைத் திருத்தி அந்த உரிமையை தற்போது உச்ச நீதிமன்றம் பாதுகாத்துள்ளது" என்று என்.ராம் கூறினார்.

"ஆதார் பதிவு விவகாரத்தில் இந்த தீர்ப்பு நிச்சயமாக பிரதிபலிக்கும் என்று கூறும் அவர், ஆதார் முறையைத் திணிக்க முயற்சிக்கும் மத்திய அரசின் எண்ணம் இனி கடினமாக இருக்கும்" என்றார். "நெருக்கடி காலத்தில் நீதிபதி எச்.ஆர். கண்ணா அளித்த தீர்ப்பு ஏற்படுத்திய மாற்றத்தைப் போல இந்தத் தீர்ப்பும் சமூகத்தில் மாற்றத்தை ஏற்படுத்தலாம் என்று சிலர் கருதுவதாகவும், மொத்தத்தில் இது ஒரு முக்கியத்துவம் நிறைந்த தீர்ப்பாகும்"

என்றும் என்.ராம் தெரிவித்தார். பிற செய்திகள் : சமூக ஊடகங்களில் பிபிசி தமிழ்

Summarized Tamil Text: அடிப்படை உரிமை என்ன என்பதை மிகவும் தீவிரமாக இத்தீர்ப்பு விளக்கியுள்ளது" என்றார். "இந்திய அரசியலமைப்பின் 21-ஆவது விதியை மிகவும் ஆழமாக நீதிமன்றம் விளக்கியுள்ளது என்றும், ஏற்கனவே இரு வேறு வழக்குகளில் தனி நபர் அந்தரங்கத்தை அடிப்படை உரிமை பாதுகாக்காது என குறிப்பிட்ட தீர்ப்புகளைத் திருத்தி அந்த உரிமையை தற்போது உச்ச நீதிமன்றம் பாதுகாத்துள்ளது" என்று என்.

Abstractive Results

MT5

இது குறித்து அவர் பிபிசி தமிழிடம் கூறுகையில், "இத்தீர்ப்பை மிகச் சிறந்த முற்போக்கான தீர்ப்பாக பார்க்கிறேன்.

அடிப்படை உரிமை என்ன என்பதை மிகவும் தீவிரமாக இத்தீர்ப்பு விளக்கியுள்ளது" என்றார்.

"இந்திய அரசியலமைப்பின் 21-ஆவது விதியை மிகவும் ஆழமாக நீதிமன்றம் விளக்கியுள்ளது என்றும்,

ஏற்கனவே இரு வேறு வழக்குகளில் தனி நபர் அந்தரங்கத்தை அடிப்படை உரிமை பாதுகாக்காது எனக் குறிப்பிட்ட தீர்ப்புகளைத் திருத்தி அந்த உரிமையை தற்போது உச்ச நீதிமன்றம் பாதுகாத்துள்ளது" என்று என்.ராம் கூறினார்.

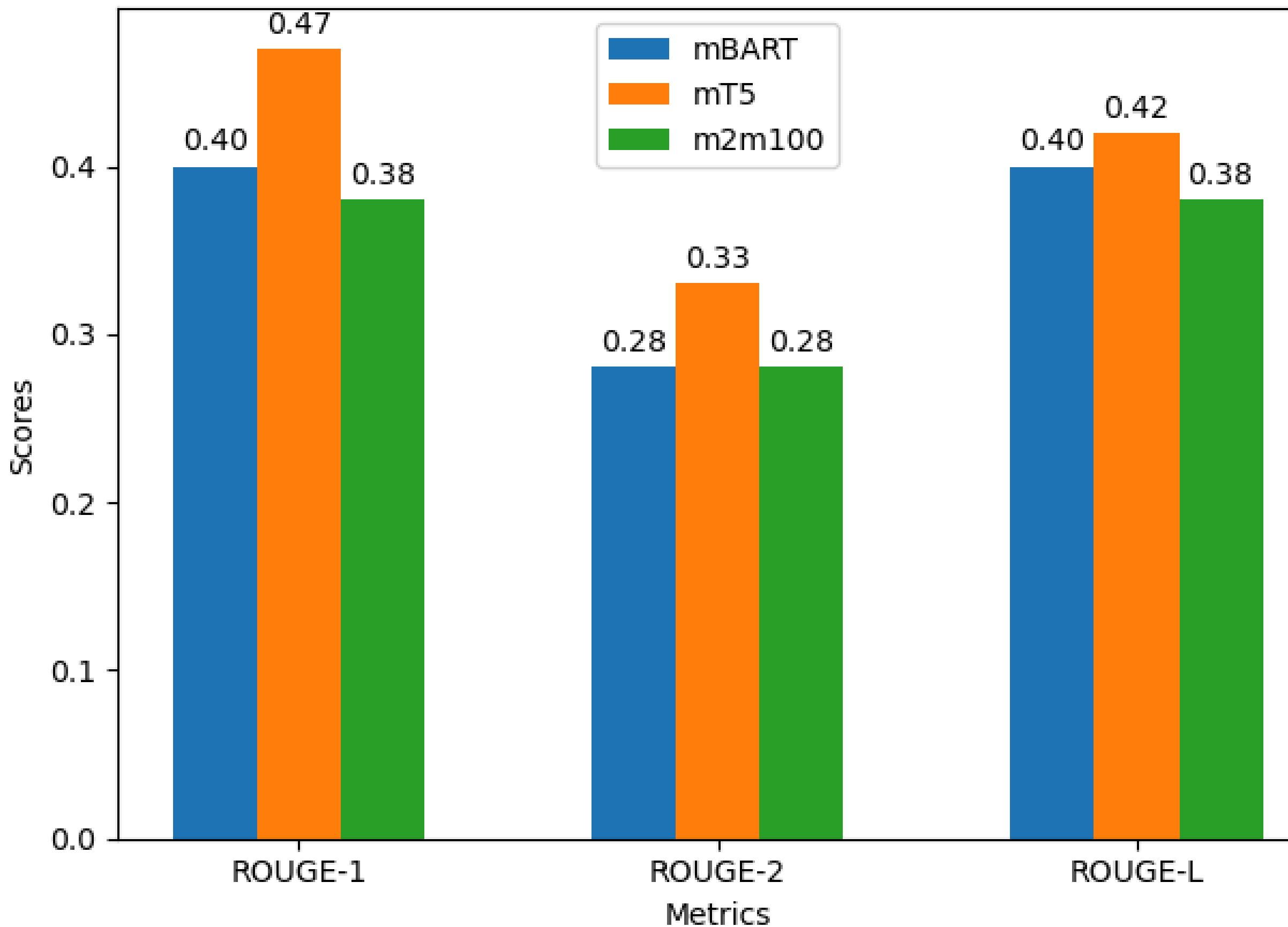
"ஆதார் பதிவு விவகாரத்தில் இந்த தீர்ப்பு நிச்சயமாக பிரதிபலிக்கும் என்று கூறும் அவர், ஆதார் முறையைத் திணிக்க முயற்சிக்கும் மத்திய அரசின் எண்ணம் இனி கடினமாக இருக்கும்" என்றார். "நெருக்கடி காலத்தில் நீதிபதி எச்.ஆர். கண்ணா அளித்த தீர்ப்பு ஏற்படுத்திய மாற்றத்தைப் போல இந்தத் தீர்ப்பும் சமூகத்தில் மாற்றத்தை ஏற்படுத்தலாம் என்று சிலர் கருதுவதாகவும், மொத்தத்தில் இது ஒரு முக்கியத்துவம் நிறைந்த தீர்ப்பாகும்"

என்றும் என்.ராம் தெரிவித்தார். பிற செய்திகள் : சமூக ஊடகங்களில் பிபிசி தமிழ்.

Summary: அடிப்படை உரிமை என்ன என்பதை மிகவும் தீவிரமாக இத்தீர்ப்பு விளக்கியுள்ளது" என்றார். "நெருக்கடி காலத்தில் நீதிபதி எச்.ஆர். பிற செய்திகள் : சமூக ஊடகங்களில் பிபிசி தமிழ்.

MODELS	ROUGE-1	ROUGE-2	ROUGE-L
mBART	0.40	0.28	0.40
mT5	0.47	0.33	0.42
m2m100	0.38	0.28	0.38

Performance of Models on Tamil Text Summarization



Performance of Models on Tamil Text Summarization

