```
# Provided data
data = {
    "incorrect_text": [
        "அவன் பாடலா எழுதினான்",  # Grammar issue
        "தம்பி நடமாடுடி",         # Spelling issue
        "இது சரியே இருக்கின்றது", # Grammar issue
        "உனக்குபட்ட ஆசிரியை நல்லவனே",  # Spelling issue
        "அவரே காட்டு போகிறேன்",  # Grammar issue
        "நான் வருகிறேன் விரைவாக",  # Spelling issue
        "அவரோ பல குணங்கள் கொண்டவர்", # Grammar issue
        "வானிலை மிகவும் பயங்கரமாக இருந்தது",  # Spelling issue
        "அவர் அழகாக உளர்ந்தார்",  # Grammar issue
        "அவர்கள் பள்ளியில் படிப்பார்கள்",  # Spelling issue
        "அவருக்கு நல்ல மனம் உள்ளார்",  # Grammar issue
        "பிறந்த நாள் வந்துவிட்டது",  # Grammar issue
        "அவரும் சந்தோஷமாக இருக்கிறார்",  # Spelling issue
        "அவர் ஒரு நல்ல நண்பர்",  # Grammar issue
        "நான் இந்த புத்தகத்தை படிக்கிறேன்",  # Grammar issue
        "எனக்கு பொருள் தெரியவில்லை",  # Grammar issue
        "பரிசு நான் பெற்றேன்",  # Spelling issue
        "நாம் அங்கு செல்லவேண்டும்",  # Grammar issue
        "இவை எல்லாம் நமது கனவுகள்"   # Grammar issue
    ],
    "correct_text": [
        "அவன் பாடல் எழுதினான்",  # Corrected
        "தம்பி நடமாடி",          # Corrected
        "இது சரியாக இருக்கிறது",   # Corrected
        "உனக்கு பட்ட ஆசிரியை நல்லவனாக உள்ளார்",  # Corrected
        "அவரும் காட்டு போகிறேன்",  # Corrected
        "நான் விரைவாக வருகிறேன்",  # Corrected
        "அவருக்கு பல குணங்கள் உள்ளன",  # Corrected
        "வானிலை மிகவும் பயங்கரவாதமாக இருந்தது",  # Corrected
        "அவர் அழகாக அழுந்தார்",  # Corrected
        "அவர்கள் பள்ளியில் படிக்கின்றனர்",  # Corrected
        "அவருக்கு நல்ல மனசுக்கொண்டவர்",  # Corrected
        "பிறந்த நாள் வந்துவிட்டது",  # Corrected
        "அவரும் சந்தோஷமாக இருக்கின்றார்",  # Corrected
        "அவர் ஒரு நல்ல நண்பராக உள்ளார்",  # Corrected
        "நான் இந்த புத்தகத்தை படித்து முடித்தேன்",  # Corrected
        "எனக்கு பொருள் புரிந்தது",  # Corrected
        "பரிசு நான் பெற்றேன்",  # Corrected
        "நாம் அங்கு செல்வோம்",  # Corrected
        "இவை எல்லாம் நமது கனவுகள் ஆகும்"  # Corrected
    ]
}


import re
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences


# Create DataFrame
```

```
df = pd.DataFrame({
    "text": data["incorrect_text"] + data["correct_text"],  # Combine both incorrect and co
    "label": [0] * len(data["incorrect_text"]) + [1] * len(data["correct_text"])  # 0 for i
})

# Tokenization and text preprocessing
def clean_text(text):
    # Remove unwanted characters and keep only Tamil letters and spaces
    text = re.sub(r'[^அ-ஹ்ாெ-ூ\s]', '', text)
    return text.strip()

df['cleaned_text'] = df['text'].apply(clean_text)

# Tokenizer and padding
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(df['cleaned_text'])
X = tokenizer.texts_to_sequences(df['cleaned_text'])
X = pad_sequences(X, padding='post', maxlen=50)  # Limiting to 50 words per sentence

y = np.array(df['label'])  # Labels

# Build the model (with trainable embeddings)
model = Sequential()
model.add(Embedding(input_dim=len(tokenizer.word_index) + 1,
                    output_dim=50,  # You can adjust the embedding dimension size
                    input_length=50,
                    trainable=True))  # Make embeddings trainable
model.add(LSTM(64, return_sequences=False))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X, y, epochs=20, batch_size=2, validation_split=0.2)

# Function to check and correct grammar
def check_and_correct(text):
    cleaned_text = clean_text(text)
    sequence = tokenizer.texts_to_sequences([cleaned_text])
    padded_sequence = pad_sequences(sequence, padding='post', maxlen=50)
    prediction = model.predict(padded_sequence)[0][0]

    if prediction > 0.5:
        return f"Corrected text: {cleaned_text}"
    else:
        return f"Grammatical issue detected: {cleaned_text}"

# Test the function
test_text = "அவன் பாடல எழுதினான் "
result = check_and_correct(test_text)
print(result)

test_text = "நான் வருகிறேன் விரைவா"
result = check_and_correct(test_text)
print(result)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90: U
  warnings.warn(
```

```
Epoch 1/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 3s 38ms/step - accuracy: 0.4566 - loss: 0.6965 - \
Epoch 2/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - accuracy: 0.7671 - loss: 0.6260 - \
Epoch 3/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.5969 - loss: 0.6779 - \
Epoch 4/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.6539 - loss: 0.6307 - \
Epoch 5/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - accuracy: 0.7256 - loss: 0.6454 - \
Epoch 6/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.7199 - loss: 0.6324 - \
Epoch 7/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.7195 - loss: 0.6125 - \
Epoch 8/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - accuracy: 0.6656 - loss: 0.6394 - \
Epoch 9/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - accuracy: 0.7968 - loss: 0.5986 - \
Epoch 10/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - accuracy: 0.7153 - loss: 0.5905 - \
Epoch 11/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.5238 - loss: 0.6872 - \
Epoch 12/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.5166 - loss: 0.6773 - \
Epoch 13/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.6013 - loss: 0.6534 - \
Epoch 14/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - accuracy: 0.7167 - loss: 0.6370 - \
Epoch 15/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.6694 - loss: 0.6415 - \
Epoch 16/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.6969 - loss: 0.6235 - \
Epoch 17/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - accuracy: 0.6344 - loss: 0.6562 - \
Epoch 18/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - accuracy: 0.7580 - loss: 0.5939 - \
Epoch 19/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - accuracy: 0.6649 - loss: 0.6678 - \
Epoch 20/20
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - accuracy: 0.4957 - loss: 0.7069 - \
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step
Grammatical issue detected: அவன் படல எழதனன்
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Grammatical issue detected: நன் வரகறன் வரவ
```

```
pip install tensorflow pandas numpy scikit-learn
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.10
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local
```

```
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in /usr/local/lib/python3
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/di
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.2
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/
Requirement already satisfied: tensorboard<2.18,>=2.17 in /usr/local/lib/python3
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/loca
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/d
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/loc
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/di
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.1
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-pack
```

```
import re
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.model_selection import train_test_split

# Provided data
data = {
    "incorrect_text": [
        "அவன் பாடலா எழுதினான்",  # Grammar issue
        "தம்பி நடமாடுடி",         # Spelling issue
        "இது சரியே இருக்கின்றது", # Grammar issue
```

```
        "உனக்குபட்ட ஆசிரியை நல்லவனே",  # Spelling issue
        "அவரே காட்டு போகிறேன்",  # Grammar issue
        "நான் வருகிறேன் விரைவாக",  # Spelling issue
        "அவரோ பல குணங்கள் கொண்டவர்", # Grammar issue
        "வானிலை மிகவும் பயங்கரமாக இருந்தது",  # Spelling issue
        "அவர் அழகாக உளர்ந்தார்",  # Grammar issue
        "அவர்கள் பள்ளியில் படிப்பார்கள்",  # Spelling issue
        "அவருக்கு நல்ல மனம் உள்ளார்",  # Grammar issue
        "பிறந்த நாள் வந்துவிட்டது",  # Grammar issue
        "அவரும் சந்தோஷமாக இருக்கிறார்",  # Spelling issue
        "அவர் ஒரு நல்ல நண்பர்",  # Grammar issue
        "நான் இந்த புத்தகத்தை படிக்கிறேன்",  # Grammar issue
        "எனக்கு பொருள் தெரியவில்லை",  # Grammar issue
        "பரிசு நான் பெற்றேன்",  # Spelling issue
        "நாம் அங்கு செல்லவேண்டும்",  # Grammar issue
        "இவை எல்லாம் நமது கனவுகள்"   # Grammar issue
    ],
    "correct_text": [
        "அவன் பாடல் எழுதினான்",  # Corrected
        "தம்பி நடமாடி",          # Corrected
        "இது சரியாக இருக்கிறது",   # Corrected
        "உனக்கு பட்ட ஆசிரியை நல்லவனாக உள்ளார்",  # Corrected
        "அவரும் காட்டு போகிறேன்",  # Corrected
        "நான் விரைவாக வருகிறேன்",  # Corrected
        "அவருக்கு பல குணங்கள் உள்ளன",  # Corrected
        "வானிலை மிகவும் பயங்கரவாதமாக இருந்தது",  # Corrected
        "அவர் அழகாக அழுந்தார்",  # Corrected
        "அவர்கள் பள்ளியில் படிக்கின்றனர்",  # Corrected
        "அவருக்கு நல்ல மனசுக்கொண்டவர்",  # Corrected
        "பிறந்த நாள் வந்துவிட்டது",  # Corrected
        "அவரும் சந்தோஷமாக இருக்கின்றார்",  # Corrected
        "அவர் ஒரு நல்ல நண்பராக உள்ளார்",  # Corrected
        "நான் இந்த புத்தகத்தை படித்து முடித்தேன்",  # Corrected
        "எனக்கு பொருள் புரிந்தது",  # Corrected
        "பரிசு நான் பெற்றேன்",  # Corrected
        "நாம் அங்கு செல்வோம்",  # Corrected
        "இவை எல்லாம் நமது கனவுகள் ஆகும்"  # Corrected
    ]
}

# Enhanced text preprocessing
def enhanced_clean_text(text):
    # Remove unwanted characters but keep Tamil letters, spaces, and punctuation
    text = re.sub(r'[^அ-ஹ்ளயீ\s.,!?]', '', text)
    # Normalize multiple spaces
    text = re.sub(r'\s+', ' ', text)
    # Add space after punctuation if missing
    text = re.sub(r'([.,!?])([^\s])', r'\1 \2', text)
    return text.strip()

# Create DataFrame with augmented data
def create_augmented_dataset(data):
    # Original data
    df = pd.DataFrame({
        "text": data["incorrect_text"] + data["correct_text"],
        "label": [0] * len(data["incorrect_text"]) + [1] * len(data["correct_text"])
    })

    # Data augmentation: Create variations of correct sentences
    augmented_texts = []
```

```
    augmented_labels = []

    for text in data["correct_text"]:
        # Add slight variations while maintaining correctness
        words = text.split()
        if len(words) > 3:
            # Swap adjacent words (if it doesn't affect grammar)
            aug_text = ' '.join(words[::-1])  # Reverse word order
            augmented_texts.append(aug_text)
            augmented_labels.append(1)

    # Add augmented data to DataFrame
    df_augmented = pd.DataFrame({
        "text": augmented_texts,
        "label": augmented_labels
    })

    return pd.concat([df, df_augmented], ignore_index=True)

# Create augmented dataset
df = create_augmented_dataset(data)
df['cleaned_text'] = df['text'].apply(enhanced_clean_text)

# Enhanced tokenization
tokenizer = Tokenizer(num_words=10000, oov_token='<OOV>')
tokenizer.fit_on_texts(df['cleaned_text'])
X = tokenizer.texts_to_sequences(df['cleaned_text'])
X = pad_sequences(X, padding='post', maxlen=50)
y = np.array(df['label'])

# Split data with stratification
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Enhanced model architecture
def build_enhanced_model(vocab_size, embedding_dim=100, maxlen=50):
    model = Sequential([
        Embedding(vocab_size + 1, embedding_dim, input_length=maxlen),
        Bidirectional(LSTM(128, return_sequences=True)),
        Dropout(0.3),
        Bidirectional(LSTM(64)),
        Dropout(0.3),
        Dense(64, activation='relu'),
        Dropout(0.3),
        Dense(32, activation='relu'),
        Dense(1, activation='sigmoid')
    ])

    # Use Adam optimizer with custom learning rate
    optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
    model.compile(
        optimizer=optimizer,
        loss='binary_crossentropy',
        metrics=['accuracy', tf.keras.metrics.Precision(), tf.keras.metrics.Recall()]
    )
    return model

# Build and train enhanced model
vocab_size = len(tokenizer.word_index)
```

```
model = build_enhanced_model(vocab_size)

import re
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.model_selection import train_test_split

# [Previous data dictionary and functions remain the same until callbacks]

# Updated callbacks with .keras extension
callbacks = [
    EarlyStopping(
        monitor='val_accuracy',
        patience=5,
        restore_best_weights=True
    ),
    ModelCheckpoint(
        'best_model.keras',  # Changed from .h5 to .keras
        monitor='val_accuracy',
        save_best_only=True
    )
]

# Train with enhanced parameters
history = model.fit(
    X_train, y_train,
    epochs=50,
    batch_size=16,
    validation_data=(X_test, y_test),
    callbacks=callbacks,
    class_weight={0: 1.0, 1: 1.2}  # Give slightly more weight to correct examples
)


# Enhanced prediction function
def check_and_correct_enhanced(text, confidence_threshold=0.7):
    cleaned_text = enhanced_clean_text(text)
    sequence = tokenizer.texts_to_sequences([cleaned_text])
    padded_sequence = pad_sequences(sequence, padding='post', maxlen=50)
    prediction = model.predict(padded_sequence)[0][0]

    if prediction > confidence_threshold:
        return {
            'status': 'correct',
            'text': cleaned_text,
            'confidence': float(prediction)
        }
    else:
        return {
            'status': 'incorrect',
            'text': cleaned_text,
            'confidence': float(prediction),
            'suggestion': 'Grammar issue detected'
        }
```

```
# Test function
def run_tests():
    test_cases = [
        "அவன் பாடலா எழுதினான்",
        "நான் வருகிறேன் விரைவா",
        "அவர் ஒரு நல்ல நண்பர்"
    ]

    print("Testing enhanced model:")
    for text in test_cases:
        result = check_and_correct_enhanced(text)
        print(f"\nInput: {text}")
        print(f"Result: {result}")

if __name__ == "__main__":
    run_tests()
```

```
Epoch 1/50
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90: U
  warnings.warn(
3/3 ━━━━━━━━━━━━━━━━━━━━ 9s 596ms/step - accuracy: 0.5982 - loss: 0.7711 - pr
Epoch 2/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 153ms/step - accuracy: 0.5748 - loss: 0.7506 - pr
Epoch 3/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 250ms/step - accuracy: 0.5435 - loss: 0.7610 - pr
Epoch 4/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 192ms/step - accuracy: 0.6060 - loss: 0.7415 - pr
Epoch 5/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 236ms/step - accuracy: 0.5982 - loss: 0.7259 - pr
Epoch 6/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 242ms/step - accuracy: 0.5592 - loss: 0.7531 - pr
Testing enhanced model:
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 778ms/step

Input: அவன் பாடலா எழுதினான்
Result: {'status': 'incorrect', 'text': 'அவன் படல எழதனன்', 'confidence': 0.
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step

Input: நான் வருகிறேன் விரைவா
Result: {'status': 'incorrect', 'text': 'நன் வரகறன் வரவ', 'confidence': 0.561
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step

Input: அவர் ஒரு நல்ல நண்பர்
Result: {'status': 'incorrect', 'text': 'அவர் ஒர நல்ல நண்பர்', 'confidence': 0
```