Smart E-Mails Assistant

Features:

1. Perform NER on emails (extracting persons, locations, times (temporal), and so on..)
2. Fill the [MASK] missing words in the text

This system was designed using `Hugging Face`

1. NER Model: https://huggingface.co/eventdata-utd/conflibert-named-entity-recognition
2. Filling Model: https://huggingface.co/google-bert/bert-base-uncased

Libraries

In [1]:
```
from transformers import pipeline
```

Loading Models

In [2]:
```
# Load the NER and Fill-Mask pipelines
ner_pipeline = pipeline("token-classification", model="eventdata-utd/conflibert-named-entity-recognition")
unmasker = pipeline('fill-mask', model='bert-base-cased')
```

```
Hardware accelerator e.g. GPU is available in the environment, but no `device` argument is passed to the `Pipeline` o
bject. Model will be on CPU.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
```

```
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
```

```
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
A parameter name that contains `beta` will be renamed internally to `bias`. Please use a different name to suppress t
his warning.
A parameter name that contains `gamma` will be renamed internally to `weight`. Please use a different name to suppres
s this warning.
Some weights of the model checkpoint at bert-base-cased were not used when initializing BertForMaskedLM: ['bert.poole
r.dense.bias', 'bert.pooler.dense.weight', 'cls.seq_relationship.bias', 'cls.seq_relationship.weight']
- This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or
with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be e
xactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).
c:\Users\Ayman\AppData\Local\Programs\Python\Python310\lib\site-packages\transformers\tokenization_utils_base.py:160
1: FutureWarning: `clean_up_tokenization_spaces` was not set. It will be set to `True` by default. This behavior will
be depracted in transformers v4.45, and will be then set to `False` by default. For more details check this issue: ht
tps://github.com/huggingface/transformers/issues/31884
  warnings.warn(
Hardware accelerator e.g. GPU is available in the environment, but no `device` argument is passed to the `Pipeline` o
bject. Model will be on CPU.
```

Functions Documentation

Overview

This document outlines three functions: `extract_entities`, `fill_missing`, and `fill_text_with_suggestions`. These functions are designed to work with Named Entity Recognition (NER) and text prediction tasks.

---

1. `extract_entities(text)`

This function extracts named entities from a given text using a Named Entity Recognition (NER) model.

Parameters

- **text** (str): The input text from which to extract entities.

Returns

- **list**: A list of dictionaries, each containing information about the identified entities, including:
  - `entity` (str): The type of the entity (e.g., PERSON, ORGANIZATION).
  - `word` (str): The actual word identified as the entity.
  - `start` (int): The start index of the entity in the text.
  - `end` (int): The end index of the entity in the text.
  - `score` (float): The confidence score of the entity recognition (filtered to scores above 0.85).

Example

```
entities = extract_entities("Apple is looking at buying U.K. startup for $1 billion.")
```

In [3]:
```
# NER function
def extract_entities(text):
    ner_results = ner_pipeline(text)
    entities = []
    for entity in ner_results:
        if entity['score'] > 0.85:  # Filter by confidence score
            entities.append({
                'entity': entity['entity'],
                'word': entity['word'],
                'start': entity['start'],
                'end': entity['end'],
                'score': entity['score']
            })
    return entities
```

Fill-Mask Function Documentation

Overview

The `fill_missing` function is designed to replace a specified word in a given text with a mask (e.g., `[MASK]`) and then generate predictions for the missing word using a language model.

Function Signature

```
def fill_missing(text: str) -> List[str]:
```

In [4]:
```
# Fill-Mask function
def fill_missing(text):
    mask_text = text.replace("Manhattan", "[MASK]")  # Example of masking a word
    predictions = unmasker(mask_text)
    return predictions[:3]  # Return top 3 suggestions
```

Fill Text with Suggestions Function Documentation

Overview

The `fill_text_with_suggestions` function replaces occurrences of a mask (e.g., `[MASK]`) in the original text with the top suggestions provided for each masked position.

Function Signature

```
def fill_text_with_suggestions(original_text: str, suggestions: List[List[Dict]]) -> str:
```

In [5]:
```
# Function to fill the text with top suggestions
def fill_text_with_suggestions(original_text, suggestions):
    filled_text = original_text
    for i, suggestion_group in enumerate(suggestions):
        # Get the top suggestion for the current mask
        top_suggestion = suggestion_group[0]['token_str']
        # Replace the [MASK] with the top suggestion
        filled_text = filled_text.replace("[MASK]", top_suggestion, 1)  # Replace only the first occurrence
    return filled_text
```

User Email Input or Example Selection

Overview

This code snippet prompts the user to choose between viewing an example email containing masked placeholders (`[MASK]`) or entering their own email text with placeholders.

In [6]:
```
# Ask the user if they want to see an example or input their own email
user_choice = input("Do you want to see an example email or input your own? (type 'example' or 'own'): ").strip().lo

if user_choice == 'example':
    # Example text
    text = """
    Hi John, please meet me at the [MASK] 10:00 AM Thursday.
    The meeting will be at 123 in front of [MASK] second street. Let me know if you have any [MASK].
    Best regards, Sarah
    """
else:
    # Input text from the user
    text = input("Please enter your email text with [MASK] where you want to fill: ")
```

Suggestions Processing and Text Filling

Overview

This code snippet retrieves suggestions for masked words in a given text, prints each suggestion with its score, and then fills the text with the top suggestions.

In [7]:
```
suggestions = fill_missing(text)

# Loop through each suggestion and print the score and token_str
for i, suggestion_group in enumerate(suggestions):
    print(f"\nSuggestions for Mask {i+1}:")
    for suggestion in suggestion_group:
        print(f"Token: {suggestion['token_str']}, Score: {suggestion['score']}")

# Get the filled text
filled_text = fill_text_with_suggestions(text, suggestions)
```

```
# Print the original and filled text
print("\nOriginal Text:")
print(text)
print("\nFilled Text:")
print(filled_text)
```

```
Suggestions for Mask 1:
Token: address, Score: 0.16885749995708466
Token: office, Score: 0.13908250629901886
Token: nearest, Score: 0.05670227110385895
Token: house, Score: 0.04021579399704933
Token: library, Score: 0.03397786617279053

Suggestions for Mask 2:
Token: the, Score: 0.9514138102531433
Token: your, Score: 0.009262710809707642
Token: my, Score: 0.008731571026146412
Token: our, Score: 0.008648613467812538
Token: a, Score: 0.005733762867748737

Suggestions for Mask 3:
Token: questions, Score: 0.6247861981391907
Token: problems, Score: 0.10022515058517456
Token: ideas, Score: 0.036073002964258194
Token: news, Score: 0.034569431096315384
Token: plans, Score: 0.02737584337592125

Original Text:

    Hi John, please meet me at the [MASK] 10:00 AM Thursday.
    The meeting will be at 123 in front of [MASK] second street. Let me know if you have any [MASK].
    Best regards, Sarah


Filled Text:

    Hi John, please meet me at the address 10:00 AM Thursday.
    The meeting will be at 123 in front of the second street. Let me know if you have any questions.
    Best regards, Sarah
```

Entity Extraction and Display

Overview

This code snippet extracts named entities from a given text and prints each entity along with its corresponding type.

In [8]:
```
# Extract entities and print them
entities = extract_entities(text)
print("Extracted Entities:")
for entity in entities:
    print(f"{entity['word']} -> {entity['entity']}")
```

```
Asking to truncate to max_length but no maximum length is provided and the model has no predefined maximum length. De
fault to no truncation.
Extracted Entities:
hi -> B-Person
john -> I-Person
me -> B-Person
10 -> B-Temporal
: -> I-Temporal
00 -> I-Temporal
am -> I-Temporal
thursday -> I-Temporal
123 -> B-Location
of -> I-Location
[MASK] -> I-Location
second -> I-Location
street -> I-Location
sarah -> B-Person
```

In [ ]:
```

```