GENAI

NAME – ANKUR SHARMA

SRN – PES2UG23CS077

Project 38 : Stock Ticker Finder

```python
from transformers import pipeline

def extract_companies(text):
    """
    Uses a Hugging Face NER pipeline to find Organizations in text.
    """
    # 1. Initialize the NER pipeline
    # We use 'simple' aggregation to group B-ORG and I-ORG tags together
    ner_pipe = pipeline(
        "ner",
        model="dslim/bert-base-NER",
        aggregation_strategy="simple"
    )

    # 2. Run the model on your news text
    results = ner_pipe(text)

    # 3. Filter for 'ORG' (Organizations) and clean the output
    found_companies = []
    for entity in results:
        if entity['entity_group'] == 'ORG':
            found_companies.append({
                "company": entity['word'],
                "confidence": round(float(entity['score']), 4),
                "start": entity['start'],
                "end": entity['end']
            })
```

```python
    return found_companies

# --- Example Usage ---
news_headline = """
Microsoft and Alphabet are seeing massive gains in the AI sector,
while Tesla faces production hurdles in Berlin.
Meanwhile, JPMorgan is advising caution on tech stocks.
"""

extracted = extract_companies(news_headline)

print(f"--- Extracted Companies for Portfolio Linking ---")
for item in extracted:
    print(f"Entity: {item['company']:<15} | Confidence: {item['confidence']}")
```

```
WARNING:torchao.kernel.intmm:Warning: Detected no triton, on systems without Triton certain kernels will not work
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart you
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100% 829/829 [00:00<00:00, 20.0kB/s]
model.safetensors: 100% 433M/433M [00:03<00:00, 223MB/s]
Some weights of the model checkpoint at dslim/bert-base-NER were not used when initializing BertForTokenClassification: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight']
- This IS expected if you are initializing BertForTokenClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a
- This IS NOT expected if you are initializing BertForTokenClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequen
tokenizer_config.json: 100% 59.0/59.0 [00:00<00:00, 5.80kB/s]
vocab.txt: 213k/? [00:00<00:00, 9.91MB/s]
added_tokens.json: 100% 2.00/2.00 [00:00<00:00, 165B/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 9.57kB/s]
Device set to use cpu
--- Extracted Companies for Portfolio Linking ---
Entity: Microsoft       | Confidence: 0.9989
Entity: Alphabet        | Confidence: 0.999
Entity: Tesla           | Confidence: 0.9949
Entity: JPMorgan        | Confidence: 0.9988
```

1. Problem Statement

Manual tracking of stock-related news is inefficient and prone to human error. In a financial portfolio management system, it is critical to quickly identify which specific companies are being discussed in a news headline to trigger relevant buy/sell alerts or risk assessments.

2. Technical Architecture

The core of this system is a Named Entity Recognition (NER) pipeline built on the BERT (Bidirectional Encoder Representations from Transformers) architecture.

2.1 The NER Logic

1. Input: Raw text (e.g., *"Microsoft is investing in AI"*).
2. Tokenization: The system breaks the sentence into smaller units called tokens. BERT uses a "Subword" tokenizer, which means it can handle unknown words by breaking them into smaller pieces.
3. Contextual Understanding: Unlike older models that read text only from left to right, BERT reads the entire sentence in both directions simultaneously. This helps it understand that "Apple" refers to the company in a financial context, not the fruit.
4. Classification: Each token is assigned a label. In our model, we filter for the ORG (Organization) label.
5. Aggregation: Since tokens can be subwords (e.g., "J", "P", "Morgan"), we use an Aggregation Strategy to re-stitch them into a single, readable entity: "JPMorgan."

3. Implementation Details

- Model Used: dslim/bert-base-NER. This is a fine-tuned version of BERT trained specifically on the CoNLL-2003 dataset, which is the gold standard for recognizing names and organizations.
- Environment: Developed in Python using the Hugging Face transformers library.

- Output Format: Structured data containing the Entity Name, Start/End character positions, and a Confidence Score (0.0 to 1.0).

4. Understanding the Results

In the generated report, the Confidence Score represents the model's mathematical certainty.

- High Confidence (>95%): These are safe to link directly to your portfolio database without human review.
- Lower Confidence (<95%): These represent ambiguous cases (e.g., a new startup or a generic name) that may require manual verification by a portfolio manager.