

```
In [1]: import feedparser
import logging
import csv
import uuid
from sqlalchemy import create_engine, Column, Integer, String, ForeignKey, DateTime
from sqlalchemy.orm import sessionmaker, relationship, declarative_base
from datetime import datetime
from celery import Celery
import spacy
```

```
In [2]: # Configure Logging
logging.basicConfig(filename='news_classification.log', level=logging.INFO, format='%(asctime)s - %s - %s' % (time.strftime('%Y-%m-%d %H:%M:%S'), time.strftime('%Y-%m-%d %H:%M:%S'), time.strftime('%Y-%m-%d %H:%M:%S')))

# Define the SQLite database engine
engine = create_engine('sqlite:///news_articles.db', echo=True)

# Define the base class for declarative class definitions
Base = declarative_base()
```

```
In [3]: # Define the Article and Source classes
class Article(Base):
    __tablename__ = 'articles'

    id = Column(Integer, primary_key=True)
    title = Column(String)
    content = Column(String)
    published_date = Column(DateTime, default=datetime.utcnow)
    source_id = Column(Integer, ForeignKey('sources.id'))
    category = Column(String, default='Others') # Added for category

    source = relationship('Source', back_populates='articles')

class Source(Base):
    __tablename__ = 'sources'

    id = Column(Integer, primary_key=True)
    name = Column(String)
    url = Column(String)

    articles = relationship('Article', back_populates='source')

# Create the tables in the database
Base.metadata.create_all(engine)

# Create a session to interact with the database
Session = sessionmaker(bind=engine)
session = Session()

# Configure Celery
app = Celery('tasks', broker='redis://:hc4vtkIYNrbIKZJwPsIXVpnGBRbi1G0K@redis-16182.c1.asia-northe
```

```
2024-01-30 17:09:44,492 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2024-01-30 17:09:44,498 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("articles")
2024-01-30 17:09:44,501 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-01-30 17:09:44,506 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("sources")
2024-01-30 17:09:44,509 INFO sqlalchemy.engine.Engine [raw sql] ()
2024-01-30 17:09:44,512 INFO sqlalchemy.engine.Engine COMMIT
```

```

In [4]: # Load spaCy English Language model
nlp = spacy.load("en_core_web_sm")

# Set up Logging
logger = logging.getLogger(__name__)

@app.task
def classify_category(article_id):
    try:
        article = session.query(Article).filter(Article.id == article_id).first()
        if article:
            doc = nlp(article.content)
            categories = []

            # Check for specific keywords and entities
            keywords = {
                "terrorism", "protest", "political unrest", "riot",
                "positive", "uplifting",
                "natural", "disaster"
            }

            entities = {ent.text.lower() for ent in doc.ents}

            # Extract relevant tokens and their POS tags
            relevant_tokens = [(token.text.lower(), token.pos_) for token in doc if token.text.lower() in keywords]

            # Classify the article based on the context of the text
            if any(entity in entities for entity in ["terrorism", "protest", "political unrest", "riot"] \
                or any(token[0] in {"terrorism", "protest", "political unrest", "riot"} for token in relevant_tokens)):
                categories.append("Terrorism/Protest/Political Unrest/Riot")
            elif any(entity in entities for entity in ["positive", "uplifting"]) \
                or any(token[0] in {"positive", "uplifting"} for token in relevant_tokens):
                categories.append("Positive/Uplifting")
            elif any(entity in entities for entity in ["natural", "disaster"]) \
                or any(token[0] in {"natural", "disaster"} for token in relevant_tokens):
                categories.append("Natural Disasters")
            else:
                categories.append("Others")

            article.category = ', '.join(categories)
            session.commit()
    except Exception as e:
        logger.error(f"Error occurred while classifying category for article ID {article_id}: {e}")

```

```

In [5]: # Define function to parse feeds and send articles to the Celery queue
def parse_feeds(feeds):
    articles = []
    for feed_url in feeds:
        try:
            feed = feedparser.parse(feed_url)
            for entry in feed.entries:
                article = {}
                article['id'] = str(uuid.uuid4())
                article['title'] = entry.title
                article['content'] = entry.get('summary', '')
                article['publish'] = entry.get('published', '')
                article['source_url'] = entry.link

                # Fetch category for the article
                doc = nlp(article['content'])
                categories = []
                for token in doc:
                    if token.text.lower() in ["terrorism", "protest", "political unrest", "riot"]:
                        categories.append("Terrorism/Protest/Political Unrest/Riot")
                    elif token.text.lower() in ["positive", "uplifting"]:
                        categories.append("Positive/Uplifting")
                    elif token.text.lower() in ["natural", "disaster"]:
                        categories.append("Natural Disasters")
                if not categories:
                    categories.append("Others")
                article['category'] = ', '.join(categories)

                if article not in articles:
                    articles.append(article)
                    # Send article to Celery queue for further processing
                    classify_category.delay(article['id']) # Pass article id
            except Exception as e:
                logger.error(f"Error occurred while parsing feed {feed_url}: {e}")
    return articles

```

```

In [6]: # Define function to export data to CSV
def export_to_csv(articles):
    try:
        with open('classified_articles.csv', 'w', newline='', encoding='utf-8') as csvfile:
            fieldnames = ['id', 'title', 'content', 'source_url', 'publish', 'category']
            writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
            writer.writeheader()
            for article in articles:
                writer.writerow(article)
    except Exception as e:
        logger.error(f"Error occurred while exporting articles to CSV: {e}")

# List of RSS feeds
rss_feeds = [
    "http://rss.cnn.com/rss/cnn_topstories.rss",
    "http://qz.com/feed",
    "http://feeds.foxnews.com/foxnews/politics",
    "http://feeds.reuters.com/reuters/businessNews",
    "http://feeds.feedburner.com/NewshourWorld",
    "https://feeds.bbci.co.uk/news/world/asia/india/rss.xml"
]

# Parse the feeds and store articles in the database
parsed_articles = parse_feeds(rss_feeds)

# Export classified articles to CSV
export_to_csv(parsed_articles)

```

In [7]: `print(parsed_articles)`

```
[{'id': '767d1aac-5e7f-4209-b60b-9d4c02c28db0', 'title': 'Some on-air claims about Dominion Voting Systems were false, Fox News acknowledges in statement after deal is announced', 'content': '', 'publish': 'Wed, 19 Apr 2023 12:44:51 GMT', 'source_url': 'https://www.cnn.com/business/live-news/fox-news-dominion-trial-04-18-23/index.html', 'category': 'Others'}, {'id': 'a5d78de7-9bcb-4922-ae94-9272820e2f29', 'title': 'Nvidia’s biggest customers are also the AI chip maker’s biggest threat', 'content': '<p>Nvidia—which roughly translates to “envy” in Latin—has been having its <a class="sc-1out364-0 dPMosf sc-145m8ut-0 lcFFec js_link" href="https://qz.com/a-40-000-nvidia-chip-has-become-the-worlds-most-sought-1850746956" rel="noopener noreferrer" target="_blank">AI moment</a>. The chip maker’s stock is up almost 30% year-to-date, closing at $624.65 on Monday.</p><p><a href="https://qz.com/nvidia-generative-ai-google-microsoft-meta-1851206854">Read more...</a></p>', 'publish': 'Tue, 30 Jan 2024 10:00:00 GMT', 'source_url': 'https://qz.com/nvidia-generative-ai-google-microsoft-meta-1851206854', 'category': 'Others'}, {'id': 'b325dfdf-ce08-413b-9b7e-f10409e2f460', 'title': 'Michigan man charged after threatening to hang Biden and Harris, bomb Washington, DC', 'content': 'Federal prosecutors charged a Michigan man who threatened to execute President Biden and Vice President Kamala Harris and called for the bombing of Washington, D.C.', 'publish': 'Tue, 30 Jan 2024 06:07:53 -0500', 'source_url': 'https://www.foxnews.com/politics/michigan-man-charged-threatening-hang-biden-harris-bomb-washington-dc', 'category': 'Others'}, {'id': '474f34be-6643-4854-9680-d39a535bf296', 'title': 'Xi Jinping celebrates China’s rising power – and his own', 'content': 'President Xi Jinping opened China’s twice-per decade Communist Party Congress on Wednesday hailing the reforms he put in place during his first five-year term and sharing his vision for where he hopes to take the nation. William Branham reports on the congress as it prepares to announce Xi’s successor and how new leadership may transform China’s role as a global economic partner.', 'publish': 'Wed, 18 Oct 2017 22:35:21 +0000', 'source_url': 'http://www.pbs.org/newshour/bb/xi-jinping-celebrates-chinas-rising-power/', 'category': 'Others'}, {'id': '0ca60fc3-b746-4a82-8621-876576d10067', 'title': 'India navy rescues Pakistani sailors from pirates', 'content': 'This was the second anti-hijacking operation in two days by India's navy off Somalia's coast.', 'publish': 'Tue, 30 Jan 2024 08:49:59 GMT', 'source_url': 'https://www.bbc.co.uk/news/world-asia-india-68091830?at_medium=RSS&at_campaign=KARANGA', 'category': 'Others'}]
```

In []: