Cahier des Charges

Projet : AI News Analyzer

Système d'analyse automatique d'articles de presse avec détection de sentiment et catégorisation intelligente

Encadré par : MANSOURI Oussema & Ghassen Khaled

Réalisé par : Jemai Abed El Mortadha - Chaima Tlatli - Omaima Jabloun

Année universitaire : 2025–2026

Étude d'existence

Il existe plusieurs plateformes d'analyse de texte et de veille médiatique telles que Google News, Feedly ou MediaCloud. Cependant, ces outils n'intègrent pas tous une analyse automatique de sentiment ou de catégorisation des articles. Le projet **AI News Analyzer** vise à combler cette lacune en exploitant le Deep Learning et le Machine Learning pour fournir une analyse intelligente du contenu médiatique (texte et images), aidant ainsi les utilisateurs à comprendre rapidement la tonalité générale des informations publiées.

Problématique

Comment concevoir une application web capable d'analyser automatiquement le contenu d'articles d'actualité (texte et images) pour en extraire le sentiment (positif, négatif ou neutre), identifier le thème principal et présenter les résultats de façon claire et interactive ?

Objectifs du projet

L'objectif principal du projet **AI News Analyzer** est de créer une application web complète utilisant le Deep Learning pour analyser des articles d'actualité. Les objectifs spécifiques sont :

- Collecter et stocker les articles dans une base de données NoSQL (MongoDB).
- Appliquer des modèles Deep Learning de pointe pour l'analyse de sentiment textuel et visuel.
- Offrir une interface web fluide et moderne pour consulter et filtrer les résultats.
- Fournir des visualisations sur les tendances médiatiques et les sentiments dominants.
- Mettre en place un pipeline DevOps pour automatiser le déploiement et la mise à jour du système.

Besoins fonctionnels

1. **Gestion des utilisateurs :**
   - Inscription, connexion et gestion des rôles (utilisateur / administrateur).
   - L'administrateur peut gérer les sources d'actualités et consulter les statistiques globales.
2. **Analyse et affichage des articles :**
   - L'utilisateur peut consulter les articles récents importés automatiquement.
   - Chaque article affiche son score de sentiment (négatif, neutre, positif) et sa catégorie prédite.
   - Analyse du sentiment des images associées aux articles.
3. **Recherche et filtrage :**
   - L'utilisateur peut filtrer les articles par date, thème ou sentiment.
4. **Visualisation :**

— Des graphiques dynamiques affichent la répartition des sentiments et des thèmes sur une période donnée.

Besoins non fonctionnels

— **Performance :** réponses rapides aux requêtes et traitement efficace des articles.
— **Sécurité :** authentification sécurisée et gestion des sessions utilisateurs.
— **Accessibilité :** interface responsive compatible avec ordinateurs et mobiles.
— **Scalabilité :** possibilité d'intégrer plus de sources et d'utilisateurs sans refonte du système.
— **Fiabilité :** robustesse et tolérance aux erreurs, notamment lors de la récupération d'articles externes.

Technologies utilisées

— **Front-end :** React.js
— **Back-end :** Django REST Framework
— **Deep Learning :** PyTorch pour l'implémentation des modèles neuronaux
— **Base de données :** MongoDB
— **DevOps :** Docker, GitHub Actions pour CI/CD, intégration continue et déploiement automatisé

Modèles Deep Learning

Le projet intègre deux modèles Deep Learning sophistiqués pour l'analyse de sentiment :

1. Modèle CNN pour l'analyse de sentiment textuel

Architecture : Réseau de neurones convolutif (CNN) personnalisé construit avec PyTorch.

Caractéristiques techniques :
— **Couche d'embedding :** Dimension 128 avec vocabulaire de 5000 mots
— **Couches convolutives :** 3 filtres parallèles de tailles 3, 4 et 5 (128 filtres chacun)
— **Pooling :** Max pooling global sur chaque filtre
— **Régularisation :** Dropout (50%) pour éviter le surapprentissage
— **Classification :** Couche fully-connected vers 3 classes (positif, neutre, négatif)

Entraînement :
— 100 époques avec batch size de 8
— Optimiseur Adam avec learning rate de 0.001
— Longueur de séquence fixe : 50 tokens
— Support GPU (CUDA) pour accélération du training

Fichiers générés :
— `sentiment_cnn.pth` : poids du modèle entraîné
— `vocab.pkl` : vocabulaire pour la tokenization

2. Modèle CNN pour l'analyse de sentiment visuel

Architecture : Transfer Learning avec ResNet18 pré-entraîné sur ImageNet.
Approche :
— Utilisation d'un modèle ResNet18 pré-entraîné (11M paramètres)
— Gel de toutes les couches convolutives (feature extraction)
— Entraînement uniquement de la couche finale (512 → 3 classes)
— Seulement 1539 paramètres entraînables (0.01% du modèle total)
Data Augmentation :
— Redimensionnement à 224×224 pixels (taille standard de ResNet)
— Flip horizontal aléatoire
— Rotation aléatoire (±10°)
— Variation de luminosité et contraste (±20%)
— Normalisation ImageNet (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
Entraînement :
— 15 époques avec batch size de 32
— Optimiseur Adam avec learning rate de 0.001
— Learning rate scheduler avec réduction automatique
— Validation continue avec sauvegarde du meilleur modèle
— Support GPU pour accélération significative
Avantages du Transfer Learning :
— Performance supérieure avec moins de données
— Temps d'entraînement réduit (15 époques vs 100+)
— Meilleure généralisation grâce aux features pré-apprises
— Précision élevée même sur dataset limité
Fichier généré :
— `image_sentiment.pth` : poids du modèle fine-tuné

Acteurs cibles

1. **Utilisateurs :** personnes souhaitant analyser le ton général des actualités et suivre les tendances médiatiques.
2. **Administrateurs :** responsables de la gestion du système, des sources et du suivi des performances globales.

Planification du projet

1. **Analyse des besoins :** étude de l'existant, identification des sources et des fonctionnalités.
2. **Conception :** réalisation des diagrammes UML et choix de l'architecture logicielle.
3. **Développement :** création de l'API Django, intégration des modèles Deep Learning PyTorch et de MongoDB.
4. **Entraînement des modèles :** préparation des datasets et entraînement des CNN.
5. **Tests et validation :** vérification des performances et du pipeline DevOps.

Diagramme de cas d'utilisation

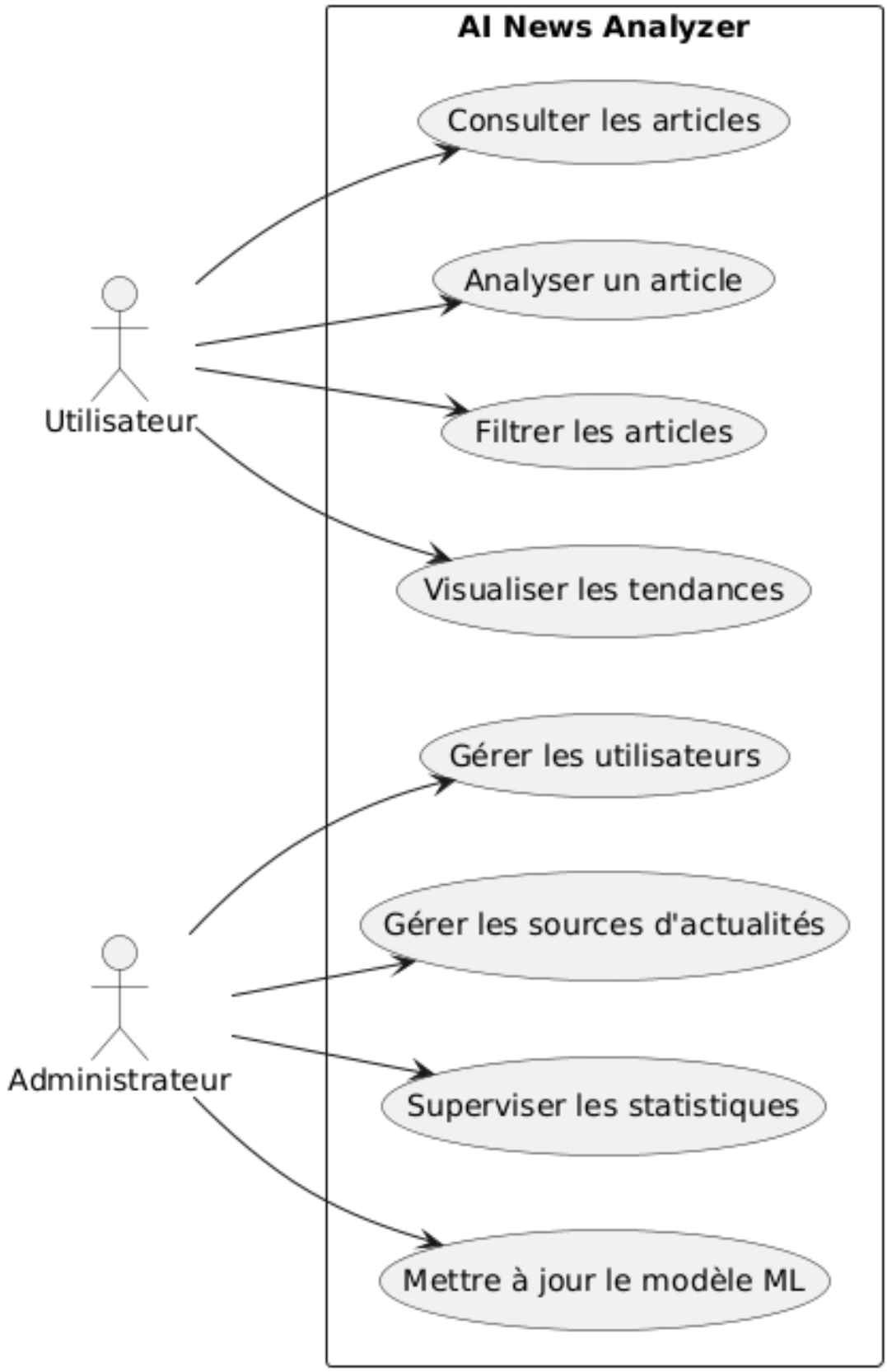

Diagramme de cas d'utilisation

Diagramme de classes

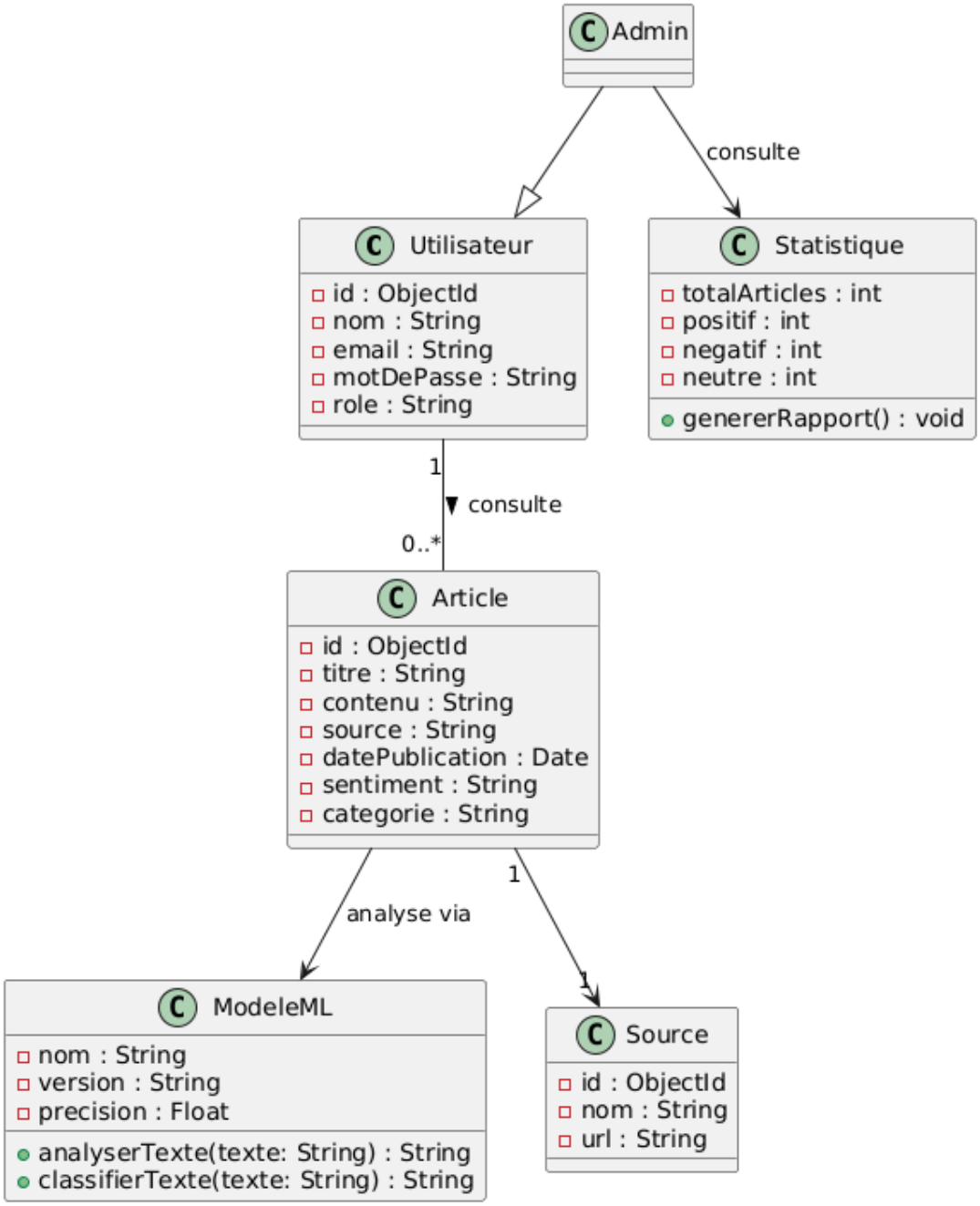

Diagramme de classes

Conclusion

Le projet **AI News Analyzer** vise à offrir une solution web innovante d'analyse des actualités reposant sur le Deep Learning de pointe. Grâce à l'utilisation de **PyTorch** et de réseaux de neurones convolutifs (CNN) pour l'analyse du sentiment textuel et visuel, ainsi que **MongoDB** pour le stockage flexible des articles, le système garantit une architecture moderne, évolutive et performante avec une précision élevée.

L'approche par Transfer Learning pour l'analyse d'images permet d'atteindre d'excellentes performances même avec des datasets limités, rendant le système pratique et

efficace pour une utilisation réelle.

Combiné à React, Django et des outils DevOps, ce projet constitue une plateforme complète, robuste et adaptée aux besoins réels d'analyse médiatique intelligente.