**Rapport de Projet n°9**

**Titre :** Analyseur de tendances sur les commentaires des utilisateurs (lefaso.net)  
**Objectif général :** Développer un système d'analyse de tendances et d'opinions sur les commentaires des utilisateurs, combinant scraping, preprocessing, analyse de sentiment et visualisation interactive.

**1. Contexte et enjeux**

Les réseaux sociaux et forums comme **lefaso.net** génèrent un volume considérable de commentaires, contenant des opinions diverses, allant du neutre au positif ou négatif. Ces données représentent un potentiel d’information stratégique pour :

* Suivre l’opinion publique.
* Détecter des tendances émergentes.
* Mesurer l’engagement des utilisateurs.

L’objectif est de créer un **tableau de bord analytique**, capable de :

* Collecter les commentaires de manière automatisée.
* Analyser les sentiments et opinions.
* Visualiser les tendances temporelles et les mots-clés.
* Détecter automatiquement des pics d’activité ou de mentions.

**2. Architecture générale du projet**

Le projet combine plusieurs modules intégrés :

1. **Scraping et collecte de données**
2. **Nettoyage et prétraitement des données**
3. **Analyse de tendances et d’opinions**
4. **Visualisation et interface interactive**
5. **Alertes automatiques**

Chaque module est interconnecté pour produire un pipeline complet, depuis la collecte jusqu’à la visualisation.

**3. Modèles Django utilisés**

**3.1. Modèle Article**

* **Champs principaux :**
  + article\_id : identifiant unique.
  + titre : titre de l’article.
  + url : lien vers l’article.
  + categorie : catégorie ou thématique.
  + date\_publication : date de publication.
  + nombre\_commentaires : nombre total de commentaires.
  + taux\_engagement : calcul basé sur le nombre de commentaires et de réponses.
* **Fonctions associées :**
  + get\_comments() : récupération de tous les commentaires liés.
  + analyze() : analyse globale des commentaires (sentiments, mots-clés).

**3.2. Modèle Commentaire**

* **Champs principaux :**
  + commentaire\_id : identifiant unique.
  + article : lien vers l’article (ForeignKey).
  + auteur : nom ou identifiant de l’auteur.
  + contenu : texte du commentaire.
  + date\_publication : date du commentaire.
* **Fonctions :**
  + sentiment\_analysis() : retourne le sentiment du commentaire (positif, neutre, négatif).
  + extract\_keywords() : extraction des mots-clés à partir du texte.

**3.3. Modèle Author (optionnel)**

* Permet de centraliser les auteurs pour calculer les statistiques globales.

**4. Pipeline de traitement**

**4.1. Collecte de données**

* **Technologie utilisée :** requests et BeautifulSoup pour le scraping.
* **Procédé :**
  1. Charger la page de l’article.
  2. Identifier les blocs de commentaires via sélecteurs CSS.
  3. Extraire le texte des commentaires.
  4. Stocker en base Django.
* **Gestion des erreurs :**
  1. Vérification des doublons.
  2. Gestion des pages non disponibles.

**4.2. Nettoyage spécialisé**

* Suppression des balises HTML, emojis, mentions et hashtags.
* Normalisation du texte pour gérer le langage informel.
* Tokenisation et suppression des stopwords.

**4.3. Détection de tendances**

* Analyse de **fréquence temporelle des mots-clés et n-grammes** :
  + Comptage des occurrences de mots par jour.
  + Identification des termes les plus utilisés.
  + Création de séries temporelles pour visualisation.

**4.4. Analyse de sentiment**

* **Méthode :**
  + Modèles NLP (TextBlob, NLTK ou transformers adaptés au français).
  + Classification des commentaires en :
    - Positif
    - Neutre
    - Négatif
  + Agrégation par article et globalement pour le site.

**5. Visualisations et interface**

**5.1. Tableau de bord interactif**

* **Technologies :** Django Templates, Bootstrap, Chart.js, WordCloud2.js.
* **Visualisations incluses :**
  + Graphiques temporels (activité par jour)
  + Graphiques d’engagement par auteur
  + Nuage de mots pour les mots-clés fréquents
  + Diagrammes de sentiment (positif, neutre, négatif)

**5.2. Nuage de mots**

* Créé dynamiquement sur <canvas> via WordCloud2.js.
* Taille et couleur des mots proportionnelles à leur fréquence.
* Tooltip au survol pour afficher le nombre d’occurrences.

**5.3. Modal d’analyse approfondie**

* Pour chaque article, un modal affiche :
  + Statistiques clés (nombre de commentaires, réponses, taux d’engagement)
  + Analyse de sentiment détaillée
  + Liste des mots-clés
  + Timestamp de l’analyse

**5.4. Alertes automatiques**

* Détection des **pics de mentions** ou de commentaires sur un mot-clé.
* Notifications dans l’interface ou via emails selon seuils configurables.

**6. Méthodologie et workflow**

1. **Scraping des articles et commentaires**
2. **Prétraitement** : nettoyage, tokenisation, normalisation
3. **Analyse quantitative** : fréquence des mots, nombre de commentaires par auteur et date
4. **Analyse qualitative** : sentiment et extraction de mots-clés
5. **Visualisation** : graphiques interactifs et nuages de mots
6. **Alertes** : détection de pics de mentions
7. **Interface utilisateur** : dashboards, modals et exports

**7. Points clés et innovations**

* **Automatisation complète** de la collecte et de l’analyse.
* **Visualisation interactive** : Dashboard clair et responsive.
* **Analyse individuelle et globale** : possibilité de consulter les tendances par article et pour l’ensemble des commentaires.
* **Fiabilité et robustesse** : gestion des cas vides, erreurs de scraping et validation des données.
* **Extensible** : facile d’ajouter de nouvelles sources ou d’autres indicateurs.

**8. Conclusion**

Le projet 9 fournit un **outil complet de suivi et d’analyse des commentaires des utilisateurs** sur lefaso.net. Il combine des techniques de **scraping, prétraitement linguistique, analyse de sentiment et visualisation interactive**. L’outil permet :

* D’identifier rapidement les tendances.
* D’évaluer l’opinion générale.
* D’observer l’engagement des auteurs.
* De générer des alertes pour les pics d’activité.

Ce système constitue un **dashboard décisionnel pour la veille de contenu et la détection d’opinions publiques**.

Le **taux d’engagement** est un indicateur clé pour mesurer **l’interaction des utilisateurs avec un contenu**. Dans le contexte de ton projet (analyse des commentaires sur lefaso.net), il permet de savoir **à quel point les articles suscitent des réactions**.

**1. Définition générale**

Le taux d’engagement se calcule généralement comme :

Taux d’engagement=Interactions totalesNombre total de vues ou articles×100\text{Taux d'engagement} = \frac{\text{Interactions totales}}{\text{Nombre total de vues ou articles}} \times 100

Dans ton projet, tu l’as défini ainsi :

Taux d’engagement=Nombre total de commentaires sur l’articleNombre d’articles×100\text{Taux d'engagement} = \frac{\text{Nombre total de commentaires sur l'article}}{\text{Nombre d'articles}} \times 100

ou parfois en incluant d’autres interactions : réponses, likes, partages (si disponibles).

**2. Comment l’interpréter**

* **Taux élevé** → Les articles suscitent beaucoup d’intérêt ou de réactions.  
  Exemple : un article qui reçoit 100 commentaires sur 10 vues (ou sur une base normalisée) montre un engagement très fort.
* **Taux moyen** → L’article ou le contenu attire un certain intérêt, mais pas exceptionnel.
* **Taux faible** → Peu de commentaires ou d’interactions. Cela peut indiquer :
  + Le contenu n’est pas pertinent pour le public.
  + Le public ne se sent pas concerné ou motivé pour interagir.

**3. Utilisation pratique**

1. **Prioriser les articles**  
   Les articles avec un taux d’engagement élevé peuvent être analysés plus en profondeur pour comprendre **les sujets qui intéressent le plus les utilisateurs**.
2. **Corrélation avec le sentiment**
   * Taux d’engagement élevé + sentiment majoritairement positif → Contenu apprécié.
   * Taux d’engagement élevé + sentiment majoritairement négatif → Sujet polémique ou sensible.
3. **Ajustement des visualisations**  
   Dans ton dashboard, le taux d’engagement peut être affiché :
   * Par article
   * Par auteur
   * Globalement sur une période
4. **Détection de pics d’activité**  
   Un pic soudain dans le taux d’engagement peut déclencher une **alerte automatique** dans le système.

Voici une formulation claire et professionnelle que tu peux utiliser pour expliquer **comment tu as mesuré et analysé les sentiments** dans ton projet :

**1. Collecte des données**

Les commentaires des utilisateurs ont été récupérés via **scraping** sur le site **lefaso.net**. Chaque commentaire constitue une unité d’analyse pour le sentiment.

**2. Prétraitement des textes**

Avant l’analyse, chaque commentaire a été nettoyé pour améliorer la qualité des résultats :

* Suppression des **caractères spéciaux**, HTML, et ponctuation inutile.
* Gestion des **émoticônes et emojis** pour conserver leur valeur émotionnelle.
* Nettoyage des **hashtags, mentions, liens**.
* Conversion du texte en **minuscules** pour uniformiser l’analyse.
* Tokenisation et éventuellement suppression des **stopwords** (mots très fréquents mais peu informatifs comme “le”, “de”, “et”).

**3. Analyse des sentiments**

Pour chaque commentaire, une **classification automatique du sentiment** a été effectuée :

* **Positif** : le commentaire exprime une opinion favorable ou encourageante.
* **Négatif** : le commentaire exprime une opinion défavorable ou critique.
* **Neutre** : le commentaire ne montre pas d’émotion forte, ou est informatif.

Méthodes possibles utilisées :

1. **Lexique de sentiment**
   * Attribution d’une note de sentiment à chaque mot (positif/négatif) et calcul d’un score global pour le commentaire.
   * Exemple : TextBlob, VADER.
2. **Modèle de Machine Learning**
   * Entraînement ou utilisation d’un modèle pré-entraîné (comme **transformers BERT ou CamemBERT**) pour classer le texte en positif/négatif/neutre.
3. **Score global par article**
   * Les scores individuels des commentaires sont agrégés pour obtenir un **sentiment moyen par article** ou par période.
   * Ces scores alimentent ensuite les **visualisations** (graphique en doughnut, tendances temporelles, etc.).

**4. Validation**

* Vérification manuelle d’un échantillon de commentaires pour **valider la précision** du modèle ou du lexique.
* Ajustement des seuils si nécessaire pour améliorer la **fiabilité des classifications**.

**5. Visualisation**

* **Diagrammes circulaires** pour la répartition global Positif / Neutre / Négatif.
* **Tendances temporelles** pour voir comment le sentiment évolue au fil du temps.
* Possibilité de croiser le **sentiment avec le taux d’engagement** pour identifier les contenus polémiques ou populaires.