

Cahier des charges - Extension Chrome "PosiText"

1. Contexte

Les interactions sur les réseaux sociaux peuvent parfois être empreintes de négativité, ce qui nuit à la qualité des échanges en ligne. Face à ce constat, l'objectif est de développer une extension Chrome interagissant avec une API d'analyse de sentiment. Cette API analysera en temps réel le contenu textuel saisi par les utilisateurs, détectant les propos négatifs et encourageant leur reformulation. Cette solution vise à promouvoir des interactions plus positives et constructives sur les réseaux sociaux tels que Facebook, Twitter, et autres.

2. Problématique

Les utilisateurs des réseaux sociaux peuvent, parfois inconsciemment, publier des messages négatifs, contribuant à un climat toxique en ligne. Une solution est nécessaire pour offrir un retour instantané sur la tonalité des messages sans perturber l'expérience utilisateur ni compromettre la confidentialité des données. Le défi réside dans la création d'une solution rapide, efficace et confidentielle, tout en étant légère et réactive.

3. Solution

L'extension "PosiText" fonctionnera en lien avec une API déployée sur un serveur AWS EC2. Cette extension capturera le texte saisi par l'utilisateur dans les champs de texte des réseaux sociaux, l'enverra à l'API pour analyse, et renverra une notification lorsque des propos négatifs seront détectés. L'API d'analyse utilisera la bibliothèque **VADER** pour analyser la polarité du texte et fournir un score de sentiment, garantissant ainsi une analyse rapide et efficace.

4. Choix technique

Langages et technologies utilisés :

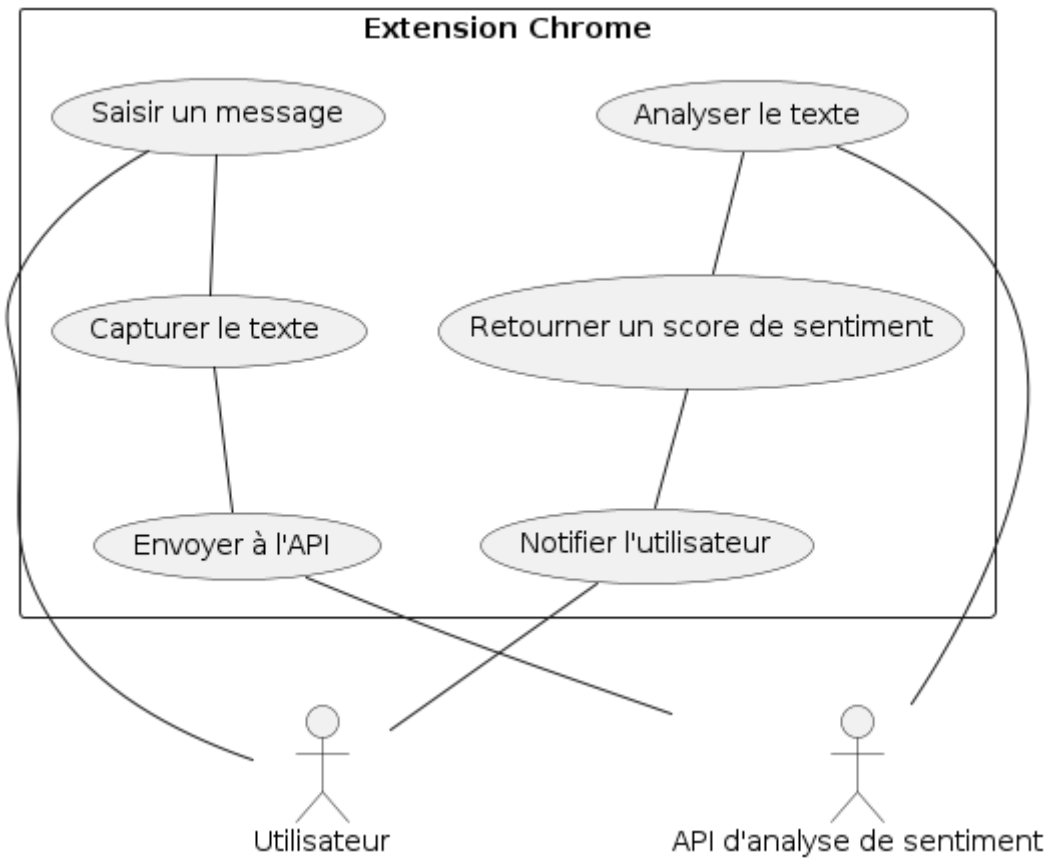
- **HTML/CSS/JavaScript** : Langages principaux pour la création de l'extension, l'injection de scripts, et l'interface utilisateur.
- **Chrome extensions API** : Interface permettant d'intégrer l'extension dans le navigateur, d'interagir avec les pages web et de capturer les données des champs de texte.
- **FastAPI** et **VADER** : FastAPI sera utilisé pour créer l'API d'analyse de sentiment. VADER, un modèle NLP léger, sera utilisé pour effectuer l'analyse de sentiment sur les textes envoyés par l'extension.
- **AWS EC2** : Serveur Ubuntu hébergeant l'API, garantissant une disponibilité et des performances optimales.

5. Fonctionnalités (sous forme de tableau)

Fonctionnalité	Description	Technologie utilisée
Capture du texte	Capturer en temps réel le texte saisi dans les champs de texte sur les réseaux sociaux	Chrome extensions API
Analyse de la polarité du texte	Envoyer le texte à l'API et analyser le sentiment (positif, neutre, négatif)	FastAPI, VADER
Notification d'alerte	Alerter l'utilisateur en cas de texte jugé négatif et proposer une reformulation	JavaScript notification API
Interface utilisateur minimaliste	Icône dans la barre d'outils ouvrant un pop-up pour gérer l'état de l'extension (ON/OFF)	HTML/CSS/JavaScript

Diagramme de cas d'utilisation :

Le diagramme illustre l'interaction entre l'utilisateur, l'extension, et l'API d'analyse de sentiment. Il décrit le flux de données, depuis la saisie du texte jusqu'à la notification de reformulation.



6. Documentation technique

a) Capture du texte

L'extension injecte un script dans les pages visitées pour détecter les champs de texte (`textarea`, `input[type="text"]`, etc.). Le script capture le contenu saisi en temps réel dès que l'utilisateur commence à taper.

b) Envoi du texte à l'API et analyse de sentiment

Une fois le texte capturé, il est envoyé à une API déployée sur **AWS EC2** via une requête HTTP POST. L'API, utilisant **VADER**, analyse immédiatement le texte pour produire un score de polarité (de -1 pour très négatif à +1 pour très positif).

c) Notification et suggestion

Si le score de polarité est inférieur à un certain seuil (exemple : -0,3), une notification est envoyée à l'utilisateur via l'**API de notification Chrome**, l'invitant à reformuler son message pour le rendre plus positif.

d) Interface utilisateur minimaliste

Un bouton d'extension situé dans la barre d'outils Chrome permet d'ouvrir un pop-up simple qui permet d'activer ou de désactiver l'extension (ON/OFF). Ce pop-up est conçu avec **HTML/CSS** pour la structure et le style, et utilise **JavaScript** pour interagir avec l'état de l'extension.

e) Manifest et structure de fichiers

- **manifest.json** : Ce fichier définit les permissions de l'extension (accès aux pages web, notifications, etc.) et précise quels scripts doivent être injectés dans les pages web.
- **background.js** : Gère l'état général de l'extension et surveille les événements importants.
- **content.js** : Injecté dans les pages web pour capturer le texte des champs de saisie, envoyer les données à l'API et traiter les réponses.
- **popup.html** : Interface utilisateur minimaliste pour activer/désactiver l'extension.
- **popup.js** : Contrôle le changement d'état de l'extension et stocke cet état dans **chrome.storage**.

f) Respect de la vie privée

Aucune donnée personnelle n'est stockée ou transmise à des serveurs tiers. L'API ne traite que le texte brut envoyé par l'extension et ne garde aucune trace de ces données une fois l'analyse effectuée.

7. À savoir

L'extension "Filtre Anti-négativité" améliore les interactions sociales en ligne en fournissant une analyse en temps réel des propos négatifs, tout en garantissant une confidentialité maximale des données. Ce projet, qui doit être réalisé en 3 jours, proposera une extension légère et réactive, soutenue par une API d'analyse de sentiment robuste et rapide, hébergée sur un serveur AWS EC2.