Université Paris Nanterre

Licence MIASHS deuxième année

Rapport de projet informatique

Détecteur de cyberharcèlement avancé en C assisté par IA

Projet réalisé du 1/01/2025 au 6/01/2025

Membres du groupe

DJEMATENE Dilan (44002835)
LACOMBE Ariane (42006077)

Dépôt GitHub :

`https://github.com/amazing-source/algo-preventioncyberharcelement`

Remerciements

Nous tenons à remercier nos enseignants pour cette proposition de projet stimulante, ainsi que la communauté open-source pour la documentation sur libcurl et json-c. Un remerciement particulier à Mistral AI pour l'accès gratuit à leur API, et à Claude (Anthropic) pour l'assistance au développement.

Table des matières

1 Introduction

1.1 Le cyberharcèlement : un fléau moderne aux conséquences dramatiques

Le cyberharcèlement est devenu l'un des problèmes les plus graves de notre ère numérique. Contrairement au harcèlement traditionnel, il ne connaît ni frontières, ni horaires : la victime peut être atteinte 24 heures sur 24, 7 jours sur 7, dans l'intimité de son foyer.

Ces dernières années, nous avons assisté à l'émergence de phénomènes particulièrement alarmants. Des groupes organisés sur des plateformes comme Telegram orchestrent des campagnes de harcèlement de masse, ciblant des individus avec une violence inouïe. Ces "raids" coordonnés peuvent détruire la réputation d'une personne en quelques heures, diffuser des contenus intimes sans consentement, ou pousser des victimes dans leurs derniers retranchements.

Les conséquences du cyberharcèlement sont bien plus graves que ce que l'on imagine généralement. Au-delà de l'anxiété et de la dépression, le cyberharcèlement peut mener à l'irréparable. Chaque année, des victimes, souvent jeunes, mettent fin à leurs jours après avoir subi des campagnes de haine en ligne. Ces drames nous rappellent que derrière chaque écran se trouve un être humain, et que les mots peuvent tuer.

1.2 Objectif du projet

Ce projet est dédié à la lutte contre le cyberharcèlement. Notre objectif est de développer un outil capable de détecter automatiquement les commentaires toxiques, insultants ou menaçants, afin d'aider à la modération des espaces en ligne.

1.3 Limites et considérations éthiques

Il est important de souligner une réalité fondamentale : **il est extrêmement difficile de quantifier la gravité d'un propos par un simple nombre**. La perception d'un commentaire varie considérablement d'une personne à l'autre. Ce qui peut sembler anodin pour l'un peut être profondément blessant pour l'autre, en fonction de son vécu, de sa sensibilité, ou du contexte.

Notre système de scoring (0-100) ne prétend pas être une vérité absolue. Il s'agit d'une estimation basée sur des critères linguistiques et contextuels, inspirés des approches utilisées par les grandes plateformes (Google Perspective API, systèmes de modération de TikTok et Reddit). Nous avons fait de notre mieux pour créer un outil aussi précis et nuancé que possible, tout en reconnaissant ses limites inhérentes.

2 Environnement de travail

2.1 Système d'exploitation

Le développement s'est effectué sous Windows 11, avec MSYS2 MinGW-w64 pour disposer d'un environnement de compilation Unix-like compatible avec les bibliothèques nécessaires.

2.2 Outils de développement

— **IDE :** Code : :Blocks 20.03 configuré avec MinGW-w64
— **Compilateur :** GCC 13.2.0 (MSYS2 MinGW-w64)
— **Terminal :** MSYS2 MinGW64 pour l'installation des dépendances
— **Assistant IA :** Claude (Anthropic) pour le développement assisté
— **Gestionnaire de paquets :** pacman (MSYS2) pour libcurl et json-c

2.3 Configuration de Code : :Blocks

Configuration spécifique pour l'utilisation de libcurl et json-c :
— **Compiler search directories :** C :\msys64\mingw64\include
— **Linker search directories :** C :\msys64\mingw64\lib
— **Link libraries :** curl, json-c

3 Description du projet et objectifs

3.1 Fonctionnalités principales (v2.0)

1. **Menu interactif complet** avec 6 modes d'utilisation
2. **Génération automatique** de commentaires réalistes par IA
3. **Analyse de commentaires reçus** : pour vérifier si un message reçu est problématique
4. **Vérification avant publication** : pour tester son propre message avant de l'envoyer
5. **Mode conversation** : analyse multiple avec statistiques
6. **Ressources d'aide** : numéros d'urgence et sites de soutien (3018, 3114, etc.)
7. **Analyse contextuelle avancée** inspirée de Google Perspective API
8. **Conseils automatiques** : suggestions de reformulation si message problématique

3.2 Architecture à deux agents IA

Le système repose sur deux agents distincts :
— **Agent Générateur :** Produit des commentaires ultra-réalistes avec fautes d'orthographe, abréviations (mdr, tkt, jsp...), absence de majuscules, style authentique des réseaux sociaux
— **Agent Analyseur :** Évalue le niveau de toxicité selon des critères contextuels multiples

4 Bibliothèques et technologies

4.1 Bibliothèque libcurl

Utilisée pour les requêtes HTTP vers l'API Mistral. Permet la gestion des en-têtes d'authentification et l'envoi de requêtes POST contenant les prompts au format JSON.

4.2 Bibliothèque json-c

Essentielle pour le parsing des réponses de l'API. Permet d'extraire les champs score, catégorie et explication des réponses structurées de l'IA.

4.3 API Mistral AI

— **Authentification :** Clé API via header Authorization Bearer
— **Endpoint :** /v1/chat/completions
— **Modèle utilisé :** mistral-small-latest (gratuit)
— **Format :** Requêtes et réponses en JSON

5 Utilisation de l'Intelligence Artificielle pour le développement

5.1 Méthodologie de développement assisté par IA

Ce projet a été réalisé avec l'assistance de Claude (Anthropic), démontrant l'efficacité du pair programming humain-IA :

— **Debugging et résolution d'erreurs :** Identification rapide des problèmes de compilation et de logique
— **Architecture et design patterns :** Structure modulaire du code
— **Génération de code :** Modules de parsing JSON et communication API
— **Optimisation des prompts :** Affinement des instructions pour l'IA d'analyse
— **Documentation :** Aide à la rédaction du README et de ce rapport

5.2 Bénéfices mesurables

— **Réduction du temps de développement :** Environ 70%
— **Qualité du code :** Meilleures pratiques appliquées systématiquement
— **Apprentissage accéléré :** Compréhension approfondie de libcurl et des callbacks

6 Fonctionnement de l'IA intégrée (API Mistral)

6.1 Présentation de Mistral AI

Mistral AI est une entreprise française proposant des modèles de langage performants. L'API est accessible gratuitement avec le modèle `mistral-small-latest`.

6.2 Agent Générateur : Commentaires ultra-réalistes

Le prompt de génération a été conçu pour produire des commentaires authentiques :

```
Tu es un generateur de commentaires de reseaux sociaux
ULTRA REALISTES.
REGLES STRICTES pour etre realiste :
- Certains commentaires ont des fautes d'orthographe
  (sa au lieu de ca, ses au lieu de c'est)
- Certains n'ont pas de majuscules du tout
- Certains utilisent des abreviations
  (tkt, mdr, ptdr, jsp, chui, pr, tt)
```

```
- Certains sont tres courts (1-5 mots)
  genre 'nul' ou 'trop bien'
- Certains ont des emojis ou des ... ou ??? ou !!!
- Style VRAI Twitter/TikTok/YouTube francais
```

6.3 Agent Analyseur : Analyse contextuelle avancée

L'analyse ne se base jamais sur les mots isolés, mais sur le contexte global :

```
Tu es un detecteur de cyberharcelement base sur des
criteres contextuels (similaires a TikTok, Reddit
et Google Perspective API).

NE TE BASE JAMAIS sur les mots isoles.
Le mot "gros" peut etre amical ("ca va gros ?")
ou insultant ("sale gros").

Evalue les elements suivants :
- Ton general : amical / neutre / moqueur / agressif
- Intention : humoristique, affective, insultante
- Structure : attaque directe ("t'es...")
- Cible : une personne specifique ou non
- Intensite : legere, moderee, forte
- Contexte linguistique : modificateurs aggravants
```

6.4 Grille de scoring contextuelle

- **0-20 :** Sain / amical / neutre
- **20-40 :** Rude / désagréable mais non insultant
- **40-60 :** Insulte légère / moquerie isolée
- **60-80 :** Attaque personnelle claire / humiliation
- **80-100 :** Haine, body-shaming agressif, menace, discrimination

7 Travail réalisé

7.1 Fonctionnalités implémentées

7.1.1 Menu interactif principal

```
----------------------------------------
              MENU PRINCIPAL
----------------------------------------
  [1] Generation automatique (IA)
  [2] Analyser un commentaire recu
  [3] Verifier avant de publier
  [4] Mode conversation (multi-analyse)
  [5] Ressources d'aide
  [6] Quitter
----------------------------------------
```

7.1.2 Mode 1 : Génération automatique

L'utilisateur choisit un thème et un nombre de commentaires. L'IA génère des commentaires ultra-réalistes (avec fautes, abréviations, style internet) puis les analyse un

par un avec statistiques finales.

7.1.3 Mode 2 : Analyser un commentaire reçu

Destiné aux **victimes potentielles**. L'utilisateur colle un message qu'il a reçu pour vérifier s'il est problématique. Si le score est élevé, le programme affiche automatiquement les ressources d'aide (3018, netecoute.fr).

7.1.4 Mode 3 : Vérifier avant de publier

Destiné aux **auteurs**. Permet de tester son propre message avant de l'envoyer. Le programme affiche des avertissements graduels selon la gravité et propose des conseils de reformulation si le message est problématique.

7.1.5 Mode 4 : Conversation

Permet d'analyser plusieurs commentaires à la suite. Affiche des statistiques globales à la fin (score moyen, répartition par catégorie, commentaire le plus toxique).

7.1.6 Mode 5 : Ressources d'aide

Affiche les numéros et sites d'aide :
— 3018 : Numéro national contre le cyberharcèlement
— 3114 : Prévention du suicide (24h/24)
— netecoute.fr, e-enfance.org, internet-signalement.gouv.fr

7.2 Modules du code source

— **main.c :** Menu interactif et logique principale
— **api_client.c/.h :** Communication HTTP avec l'API Mistral
— **comment_generator.c/.h :** Agent 1 - Génération réaliste
— **harassment_detector.c/.h :** Agent 2 - Analyse contextuelle
— **stats.c/.h :** Calcul et affichage des statistiques
— **config.h :** Configuration de la clé API

8 Répartition du travail

— **DJEMATENE Dilan :** Architecture générale, développement du code C, intégration de l'API Mistral.
— **LACOMBE Ariane :** Recherches sur le cyberharcèlement, rédaction du rapport, tests utilisateur, documentation, relecture, tests et debugging, conception des prompts IA

9 Difficultés rencontrées

9.1 Gestion des accents sous Windows

L'affichage des caractères accentués dans la console Windows a nécessité une configuration spéciale avec `SetConsoleOutputCP(65001)` pour passer en UTF-8.

9.2 Calibrage du scoring

Trouver le bon équilibre entre faux positifs et faux négatifs a demandé de nombreuses itérations sur les prompts. L'approche contextuelle (ne pas se baser sur les mots isolés) a grandement amélioré la précision.

9.3 Réalisme des commentaires générés

Les premiers commentaires générés étaient trop "propres" et ne ressemblaient pas à de vrais commentaires internet. L'ajout d'instructions spécifiques (fautes, abréviations, style oral) a résolu ce problème.

10 Bilan

10.1 Résultats obtenus

Le détecteur v2.0 offre une expérience interactive complète avec trois modes d'utilisation. L'analyse contextuelle permet une évaluation nuancée qui distingue les usages amicaux des usages insultants d'un même mot.

10.2 Conclusion

Ce projet démontre qu'il est possible de créer un outil de détection de cyberharcèlement en C, exploitant l'intelligence artificielle pour une analyse sémantique fine. Bien que le système ne soit pas parfait (la perception de la gravité reste subjective), il constitue une base solide pour aider à la modération des espaces en ligne.

La lutte contre le cyberharcèlement est l'affaire de tous. Nous espérons que ce projet contribuera, à son échelle, à sensibiliser et à protéger les victimes potentielles.

10.3 Perspectives

— Intégration avec des APIs de réseaux sociaux (Discord, Reddit)
— Interface graphique pour une utilisation grand public
— Système d'apprentissage pour améliorer la précision au fil du temps
— Extension multilingue

11 Bibliographie

[1] Aucun ouvrage papier consulté.

12 Webographie

[CURL] Documentation libcurl : https://curl.se/libcurl/c/
[JSONC] Documentation JSON-C : https://github.com/json-c/json-c
[MISTRAL] Documentation API Mistral : https://docs.mistral.ai/
[PERSPECTIVE] Google Perspective API : https://perspectiveapi.com/
[MSYS2] MSYS2 guide d'installation : https://www.msys2.org/
[CLAUDE] Assistant IA Claude : https://claude.ai/

13 Annexes

Annexe A : Cahier des charges

Annexe A.1 : Objectifs fonctionnels

- Menu interactif avec plusieurs modes
- Générer des commentaires réalistes via l'Agent 1
- Permettre l'analyse de commentaires personnalisés
- Analyser chaque commentaire via l'Agent 2 (analyse contextuelle)
- Attribuer un score de toxicité (0-100)
- Classifier selon les catégories définies
- Fournir une explication pour chaque classification
- Générer des statistiques globales

Annexe A.2 : Contraintes techniques

- Langage C uniquement
- Compilation avec GCC sans warnings
- Exécution en ligne de commande avec menu interactif
- Communication API via libcurl
- Parsing JSON via json-c

Annexe B : Exemple d'exécution du projet

```
$ ./cyberharcelement_detector

================================================
     DETECTEUR DE CYBERHARCELEMENT v2.0
       Analyse & Prevention par Intelligence Artificielle
================================================

----------------------------------------
              MENU PRINCIPAL
----------------------------------------
  [1] Generation automatique (IA)
  [2] Analyser un commentaire recu
  [3] Verifier avant de publier
  [4] Mode conversation (multi-analyse)
  [5] Ressources d'aide
  [6] Quitter
----------------------------------------

Votre choix : 5

================================================
          RESSOURCES D'AIDE ET DE PREVENTION
================================================

Si vous etes VICTIME de cyberharcelement :
----------------------------------------
  - 3018 : Numero national contre le cyberharcelement (gratuit)
  - https://www.netecoute.fr : Ecoute et conseils
```

```
  - https://www.e-enfance.org : Protection des mineurs

Si vous avez des PENSEES SOMBRES :
-----------------------------------------
  - 3114 : Numero national de prevention du suicide (24h/24)
  - https://www.sos-amitie.com : Ecoute 24h/24
```

```
Votre choix : 2

=== ANALYSER UN COMMENTAIRE RECU ===
(Pour verifier si un message que vous avez recu est problematique)

Collez le commentaire a analyser :
> ca va gros ? t'as passe un bon weekend ?

[Analyse en cours...]

--- RESULTAT DE L'ANALYSE ---
| Score     : 8/100
| Categorie : sain
|----------------------------
| Explication :
| Le terme "gros" est utilise de maniere amicale
| et familiere. Le ton est bienveillant.
----------------------------
[OK] Ce commentaire semble sain.
```

```
Votre choix : 3

=== VERIFIER AVANT DE PUBLIER ===
(Testez votre message AVANT de l'envoyer)

Ecrivez votre commentaire :
> t'es vraiment qu'un gros debile

[Verification en cours...]

##################################################
#           VERIFICATION AVANT PUBLICATION       #
##################################################

[XXX] STOP ! Ce commentaire est une ATTAQUE.
      Categorie : humiliation (score: 72/100)

      -> Ceci peut constituer du HARCELEMENT.
      -> La victime pourrait PORTER PLAINTE.
      -> Vous risquez des POURSUITES JUDICIAIRES.

      NE PUBLIEZ PAS CE MESSAGE.
      Respirez. Fermez l'ecran. Revenez plus tard.

##################################################

Voulez-vous des suggestions pour reformuler ? (o/n) : o

Conseils pour reformuler :
- Exprimez votre desaccord sans attaquer la personne
- Utilisez "je pense que..." plutot que "tu es..."
- Critiquez les idees, pas les individus
```

Annexe C : Manuel utilisateur

Annexe C.1 : Installation

```
# Ouvrir MSYS2 MinGW 64-bit

# Installation des dependances
pacman -S mingw-w64-x86_64-curl
pacman -S mingw-w64-x86_64-json-c

# Compilation
make
```

Annexe C.2 : Configuration

Modifier le fichier `config.h` avec votre clé API Mistral (la mienne est mise à prédisposition) :

```
#define API_KEY "votre-cle-mistral-ici"
```

Annexe C.3 : Utilisation

```
./cyberharcelement_detector
```

Puis suivre le menu interactif.

Annexe D : Structure du code source

```
projet/
  |-- main.c                  # Menu interactif
  |-- api_client.c            # Communication API
  |-- api_client.h
  |-- comment_generator.c     # Agent 1
  |-- comment_generator.h
  |-- harassment_detector.c   # Agent 2
  |-- harassment_detector.h
  |-- stats.c                 # Statistiques
  |-- stats.h
  |-- config.h                # Configuration
  |-- Makefile
  |-- README.md
  |-- rapport/                # Sources LaTeX
      |-- main.tex
      |-- pagedegarde.sty
      |-- bordure.png
      |-- logo_Paris_Nanterre_couleur_RVB.png
```