Documentation scientifique du groupe MoodMatrix du projet Hackathon

par : Mohammed, Reda, Alexandre, Saeed et Sandy.

Sommaire :

API d'analyse de sentiment/émotion

veille :

Analyse de sentiment et Classification des émotions :

L'analyse de sentiment et la classification des émotions sont deux approches différentes, mais liées dans le domaine du traitement automatique du langage naturel (TALN).

Analyse de sentiment : Cela consiste à évaluer le sentiment global exprimé dans un morceau de texte, généralement en le classant comme positif, négatif ou neutre. L'objectif est de déterminer le ton général du texte ou de comprendre comment les gens se sentent à propos d'un sujet spécifique. Par exemple, dans un tweet disant "J'adore ce nouvel album!", l'analyse de sentiment identifierait le sentiment comme positif.

Classification des émotions : Contrairement à l'analyse de sentiment qui se concentre habituellement sur des catégories générales comme positif, négatif ou neutre, la classification des émotions vise à identifier des émotions spécifiques telles que la joie, la tristesse, la colère, la peur, etc. Elle vise à comprendre les nuances des émotions exprimées dans un texte. Par exemple, dans la phrase "Je suis tellement triste que tu ne puisses pas venir", la classification des émotions pourrait identifier la tristesse.

Comparaison des outils NLP d'analyse de sentiment/émotion :

VADER (Valence Aware Dictionary and sEntiment Reasoner) : VADER est un outil d'analyse de sentiment basé sur des règles et conçu pour analyser les sentiments exprimés dans du texte social. Il est souvent utilisé pour l'analyse de sentiment dans les médias sociaux. Python : nltk.sentiment.vader.

TextBlob : TextBlob est une bibliothèque Python pour le traitement du langage naturel qui inclut des fonctionnalités d'analyse de sentiment. Il utilise une approche basée sur des lexiques pour attribuer des polarités (positives, négatives ou neutres) aux mots et calcule ensuite la polarité globale du texte. Python : textblob.sentiments.

Stanford CoreNLP : Il s'agit d'une suite d'outils de traitement du langage naturel développée par Stanford University. Elle comprend un module d'analyse de sentiment capable de classifier les phrases en positif, négatif ou neutre. Java : edu.stanford.nlp.

FastText : FastText est une bibliothèque open source de Facebook pour le traitement du langage naturel et l'apprentissage de vecteurs de mots. Il inclut des fonctionnalités d'analyse de sentiment par classification de texte. Python : fasttext.

Scikit-learn : Scikit-learn est une bibliothèque Python pour l'apprentissage automatique. Bien qu'elle ne fournisse pas de modèles pré-entraînés pour l'analyse de sentiment, elle propose des outils et des algorithmes pour construire des modèles

personnalisés d'analyse de sentiment en utilisant des techniques telles que la classification naïve bayésienne, les machines à vecteurs de support (SVM), etc.

RoBERTa: BERT (Bidirectional Encoder Representations from Transformers) : BERT est un modèle de langage profond développé par Google. Bien qu'il soit principalement utilisé pour des tâches de compréhension de texte, il peut également être adapté à l'analyse de sentiment en peaufinant le modèle sur des données d'analyse de sentiment. Implémentations disponibles en Python : Huggingface Transformers, TensorFlow, PyTorch.

Expérimentation :

Nous avons d'abord cherché un modèle de reconnaissance d'émotions en français, les seuls modèles disponibles étaient soit des modèles d'analyse de sentiment ou des modèles de classification d'émotions peu performant/pas à jour. Nous nous sommes donc tournés vers une architecture avec une première étape de traduction vers l'anglais (GoogleTranslator), puis nous avons utilisé un modèle performant en anglais (EmoRoBERTa) pour détecter les émotions.

résultats :

Choix : EmoRoBERTa
RoBERTa est une amélioration de la stratégie de masquage de langage de BERT et modifie certains hyperparamètres clés de BERT, notamment en supprimant l'objectif de pré-entraînement de la phrase suivante de BERT, et en s'entraînant avec des mini-lots et des taux d'apprentissage beaucoup plus grands. RoBERTa a également été entraîné sur un ordre de grandeur de données plus important que BERT, pendant une période plus longue. Cela permet aux représentations de RoBERTa de généraliser encore mieux aux tâches aval que BERT.
Nous utilisons la variante EmoRoBERTa adaptée à la classification d'émotions.

API de chatbot

veille :

Comparaison des outils de chatBot :

Crisp chatbot : Cette extension inclut le live chat, les notifications en temps réel, le chatbot, les applications pour ordinateur de bureau et portable et le paramétrage des heures de disponibilité du chat. Le chatbot est payant.

Rasa : Chatbot opensource et très rapide à prendre en main. Il est entièrement personnalisable et donc très adaptable. Mais il ne fonctionne pas avec les versions python plus récentes que 3.10 et il a besoins d'être entraîner de A à Z ce qui peut prendre beaucoup de temps

Mistral 8x7b : Modèle gratuit, rapide et performant, un concurrent de GPT 3.5.

neo GPT : Basé sur chatGPT 3.5, chatbot disponible gratuitement et open-source, cependant a besoins d'entraînement pour être efficace

FrenchAlpaca : Modèle basé sur le modèle Mistral 7b, entraîné sur un dataset en français pour le spécialiser dans la compréhension de la langue. Semble efficace mais requiert un pc puissant sous Nvidia puisqu'il utilise la technologie CUDA

DialoGPT : Modèle de microsoft open-source et basé sur gpt2, spécialisé dans les conversation, entrainé sur des datasets de forum sur reddit ou autre. Cependant Manque de compréhension en français

Blender bot : Modèle de meta(facebook), modèle gratuit mais non-entraîné

GPT2 : Modèle d'OPENAI, modèle gratuit mais non-entraîné

expérimentation :

Mistral 8x7b : Modèle efficace, adapte son texte en fonction de la langue et de l'émotion facilement. Cependant, peut générer des entrées à la place de l'utilisateur

DialoGPT : Modèle moyennement efficace, comprend bien et répond bien de manière interactive. Cependant, il ne comprend pas totalement le français et finis par répondre en anglais dans certains cas

résultats :

Choix : Mistral 8x7b s'est révélé rapidement être le modèle de prédilection pour notre projet. Même s'il peut arriver qu'il puisse agir de manière autonome, sa compréhension des émotions au point même qu'il puisse répondre avec des emojis et son adaptation en fonction du langage montre bien que le modèle est puissant. Je précise que la version qu'on utilise est une version démo qui n'a pas été entrainé par la suite. Cela prouve que ce modèle est puissant.

Détail technique de l'application

Overview Appli :

Voici notre application Flask ‘Emobot’. C'est une application web simple construite avec Flask qui permet aux utilisateurs d'interagir avec un chatbot. Les utilisateurs peuvent saisir des messages, et le chatbot répond en conséquence. L'application garde une trace de l'historique des conversations, y compris les horodatages, les messages des utilisateurs, les réponses du bot et les émotions associées à chaque interaction.

Composants clés :

1. Routes Flask :

- Trois routes principales sont définies dans l'application Flask :
- /: Renders l'interface de chat principale où les utilisateurs peuvent interagir avec le chatbot.
- /refresh : Efface l'historique de chat et redirige vers l'interface de chat principale.
- /monitoring : Rend une page de surveillance, affichant certaines données récupérées d'une base de données.

Intégration des fonctions :

- La fonctionnalité principale de génération de réponses de chatbot est intégrée avec la fonction **text_analyse_et_generation.**
- La fonction **get_df_bd_monitoring** est intégrée pour récupérer des données pour la page de surveillance.

Méthodologie :

- La route principale (/) gère les saisies des utilisateurs en capturant les requêtes POST. Elle traite l'entrée, génère des réponses de chatbot et met à jour l'historique de chat en conséquence.
- La route /refresh efface l'historique de chat sur demande.
- La route monitoring récupère des données de la base de données et les rend dans un tableau HTML sur la page de surveillance.