PROJET UE-4

GUNDUZ Maxime

AIT SAID Hicham

Master 2 Informatique Biomédicale

2024-2025

Identification des interlocuteurs par intelligence artificielle dans les appels du Samu

Parti 1 : Évaluer et comparaison de performances

Notre objectif dans cette partie était d'évaluer et de comparer deux méthodes d'apprentissage à partir du jeu de données Iris de la librairie sklearn, qui contient des informations sur trois espèces de fleurs. Nous avons choisi une méthode supervisée, le Random Forest, qui utilise les catégories connues pour apprendre à les reconnaître, et une méthode non supervisée, le K-Means, qui regroupe les données en fonction de leurs ressemblances sans connaître les catégories à l'avance.

Pour visualiser leurs performances, nous avons réalisé 20 itérations sur chacun des deux modèles afin d'obtenir les taux de précision, tout en modifiant progressivement le taux sélectionné du jeu d'entraînement (on réalise un mélange à chaque itération) pour obtenir un nuage de points représentant les différences de performance entre ces deux modèles.

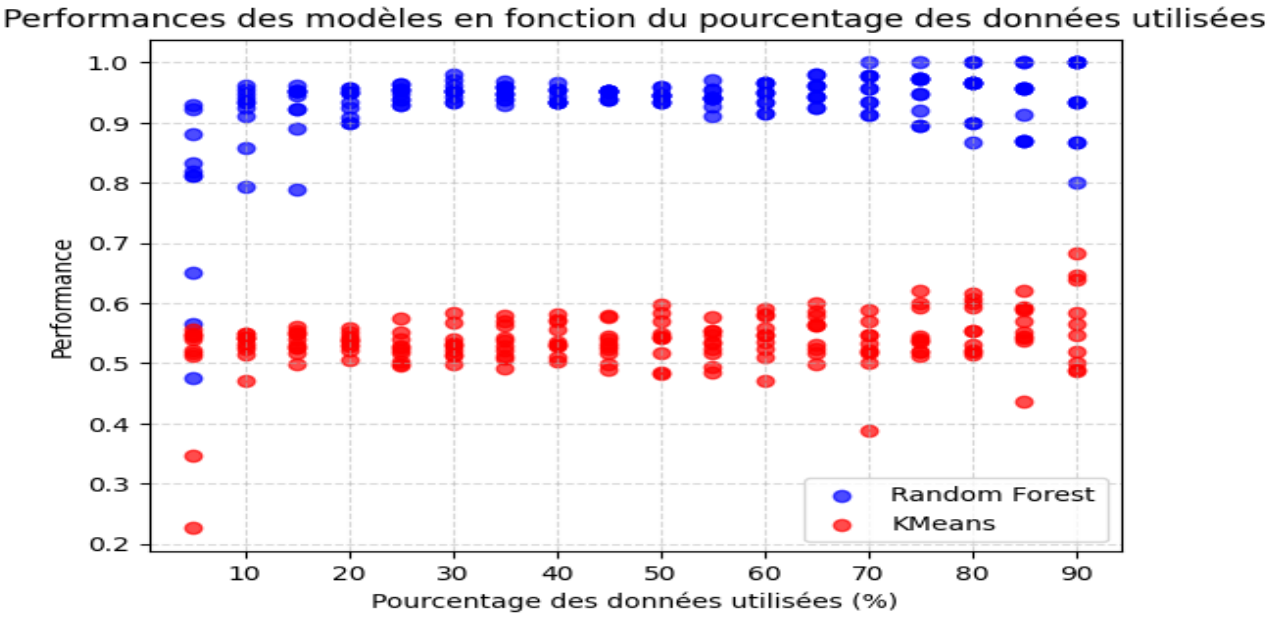

Figure représentant un nuage de points comparative des 2 modèles

L'analyse des résultats montre une supériorité de «Random Forest» par rapport au «K-Means». Le «Random Forest» maintient des performances élevées avec des taux de précision oscillant entre 80% et 100% généralement, même avec un faible pourcentage de données d'entraînement. Ceci démontre la robustesse de cette méthode supervisée. En revanche, le K-Means affiche des performances plus modestes avec des scores compris entre 45% et 65%. Cette différence s'explique principalement par le fait que le Random Forest bénéficie de l'information des étiquettes pendant l'apprentissage, tandis que le K-Means doit découvrir la structure des données sans cette information préalable. De plus, on observe que l'augmentation du volume de données d'entraînement n'améliore pas les performances du K-Means, suggérant que ce modèle atteint rapidement ses limites sur ce jeu de données.

En guise de conclusion, cette comparaison montre que les méthodes supervisées comme «Random Forest» sont plus performantes en termes de performance car elles apprennent à partir de données étiquetées, tandis que les méthodes non supervisées comme K-Means sont moins précises dans ce cas-là, mais restent pertinentes lorsque l'on ne dispose pas de données préalablement classifiées.

Parti 2 : Traitement Automatique des Langues

Introduction

Le SAMU (Service d'Aide Médicale d'Urgence) est un service public français dédié à l'assistance médicale d'urgence et à la coordination des secours en cas de situations médicales critiques. Le SAMU reçoit de nombreuses sollicitations téléphoniques tout au long de l'année, impliquant 2 interlocuteurs principaux, un appelant qu'on qualifie de «patient», et un appelé qualifié de «médecin».

Dans ce contexte, notre objectif est de concevoir et d'entraîner 2 modèles d'intelligence artificielle, l'un basé sur une méthode classique et l'autre sur une méthode de deep learning. Ces modèles visent à identifier automatiquement les interlocuteurs dans les conversations téléphoniques enregistrées. Pour ce faire, nous devons extraire et structurer les échanges, puis attribuer à chaque segment l'une des étiquettes «Médecin» ou «Patient» en suivant notre jeu de données. Le jeu de données ainsi obtenu servira à entraîner et évaluer nos modèles, avec pour objectif un taux minimal de précision de 80%.

Matériel et méthode

Traitement du jeu de données :

Afin de développer nos modèles d'intelligence artificielle répondant à nos objectifs, nous avons commencé par identifier un jeu de données adapté. Nous avons choisi le jeu de données SimSamu, disponible sur la plateforme HuggingFace, et qui contient les dialogues textuels entre les interlocuteurs identifiés comme «médecin» et «patient». Ceci est nécessaire pour entraîner nos modèles. Il offre des échanges fictifs simulant des appels au SAMU, avec des cas définis. Nous avons importé un repo Github contenant uniquement et exactement le jeu de données «SimSamu».

Les fichiers fournis par SimSamu incluent des fichiers «srt» contenant les transcriptions textuelles des échanges avec des repères temporels, mais sans l'identification des interlocuteurs. Parallèlement, les fichiers «rttm» indiquent l'identité des interlocuteurs ainsi que leurs intervalles temporels respectifs. Notre objectif était de fusionner ces 2 sources pour construire une structure de données claire et exploitable pour l'entraînement de nos modèles.

Nous avons conçu un code Python permettant d'extraire et de regrouper les dialogues par interlocuteur. Chaque échange est intégré dans une structure sous forme de dictionnaires où la clé correspond à l'interlocuteur («médecin» ou «patient») et la valeur est une liste contenant les phrases prononcées par l'interlocuteur. Pour chaque conversation, nous avons synchronisé les segments

temporels des fichiers «rttm» avec les lignes de dialogue des fichiers «srt». Cette étape de prétraitement nous a permis de constituer un dataset structuré et prêt à être utilisé pour entraîner nos modèles. L'extraction des données a été réalisée en parcourant récursivement les fichiers et les différents cas de SimSamu. Nous disposons désormais d'une base permettant de construire nos modèles d'IA, l'un basé sur une méthode classique et l'autre sur une approche de deep learning.

```
root_directory = "/content/simsamu"
get_all_discussions(root_directory)

print(f"Nombre de discussions : {len(discussions_samu)}")
print(discussions_samu[0])
print(discussions_samu)
```

```
Nombre de discussions : 61
{'patient': ['allo?', 'oui allo, allo?', "oui c'est ça c'e
[{'patient': ['allo?', 'oui allo, allo?', "oui c'est ça c'
```

Figure illustrant l'affichage d'un extrait de notre jeu de données

Traitement des données textuelles, approche TF-IDF

Nous avons utilisé la méthode TF-IDF (Term Frequency - Inverse Document Frequency) pour convertir les phrases du jeu de données SimSamu en vecteurs. Cette technique, réduit l'influence des mots fréquents peu informatifs tout en valorisant les termes distinctifs. Les textes ont été prétraités pour supprimer les caractères non alphabétiques et les mots inutiles, tels que « heu » ou « bah », afin de focaliser l'analyse sur les éléments pertinents pour différencier les interlocuteurs, On a mis tout en minuscule.
La vectorisation TF-IDF a produit une matrice 3061 x 2625, où chaque phrase est représentée par un vecteur de poids des termes les plus informatifs. Cette méthode a permis d'identifier les mots-clés propres à chaque classe : par exemple, « madame » et « médecin » pour les médecins, ou « non » et « hein » pour les patients. En transformant les données textuelles en un format numérique adapté aux algorithmes d'apprentissage, la TF-IDF met en évidence les différences contextuelles tout en réduisant l'impact des mots communs, facilitant ainsi la classification.

```
Forme de la matrice TF-IDF : (3061, 2625)
Nombre total de phrases : 3061
Distribution des classes : [1594 1467]

Mots les plus importants par classe :

MEDECIN :
- accord
- madame
- ça
- ok
- revoir
- là
- oui
- médecin
- si
- donc
- va
- bien
- bonjour
- faut
- allo
```

Figure illustrant les mots les plus utilisé par « Médecin »

Conception d'un modèle en méthode classique : la régression logistique

Nous avons conçu un modèle basé sur la régression logistique pour classifier les phrases en fonction des interlocuteurs («médecin» ou «patient»). La régression logistique est un algorithme classique en apprentissage supervisé, particulièrement adapté pour les tâches de classification binaire comme la nôtre.

Pour intégrer ce modèle, nous avons utilisé la matrice TF-IDF que nous avons obtenue.

Pour améliorer les performances, nous avons créé un pipeline avec 2 étapes, une sélection des caractéristiques importantes et l'application de la régression logistique. La sélection des caractéristiques a été faite en utilisant la méthode de pénalisation L1, qui garde uniquement les termes les plus pertinents, ce qui rend le modèle plus robuste et réduit les risques de surapprentissage.

Enfin, nous avons utilisé une recherche par validation croisée (GridSearchCV) pour ajuster les paramètres du pipeline, comme le type de régularisation, le seuil de sélection et le poids des classes.

Cela assure que notre modèle est bien adapté pour reconnaître les différences entre les interlocuteurs et s'adapte aux variations des données issues des échanges du SAMU.

Conception d'un modèle en méthode de Deep Learning, modèle préentrainé BERT

Pour classifier les interlocuteurs («médecin» ou «patient») dans les dialogues SAMU, nous avons conçu un modèle basé sur CamemBERT, une variante francophone de BERT (Bidirectional Encoder Representations from Transformers). CamemBERT est particulièrement adapté à notre cas d'étude, car il est spécifiquement préentrainé sur des textes en français, permettant ainsi de capturer les nuances linguistiques et contextuelles des conversations.

Inspirés des approches utilisées dans l'analyse des sentiments, nous avons intégré des prompts contextuels, tels que « Est-ce que cette phrase vient d'un médecin ou d'un patient ? », afin de guider le modèle dans la différenciation des interlocuteurs. Les phrases ont été prétraitées et encodées à l'aide du tokenizer CamemBERT, avec une longueur maximale de 128 tokens.

Nous avons utilisé le modèle CamemBERTForSequenceClassification, configuré pour une classification binaire, en ajustant les poids via un entraînement supervisé sur notre jeu de données spécifique. L'ajout de techniques telles que la régulation des logits par température, l'utilisation d'un optimiseur avancé (AdamW), et une fonction de perte adaptée (CrossEntropyLoss) a permis d'optimiser les performances. Cette méthode capitalise sur les capacités de Deep Learning et sur les connaissances préexistantes du modèle pour offrir une classification précise et adaptée aux besoins du SAMU.

Résultats

Résultat de notre modèle sur la régression logistique

L'évaluation de notre modèle de régression logistique sur l'ensemble de test (613 phrases) révèle les données suivantes :

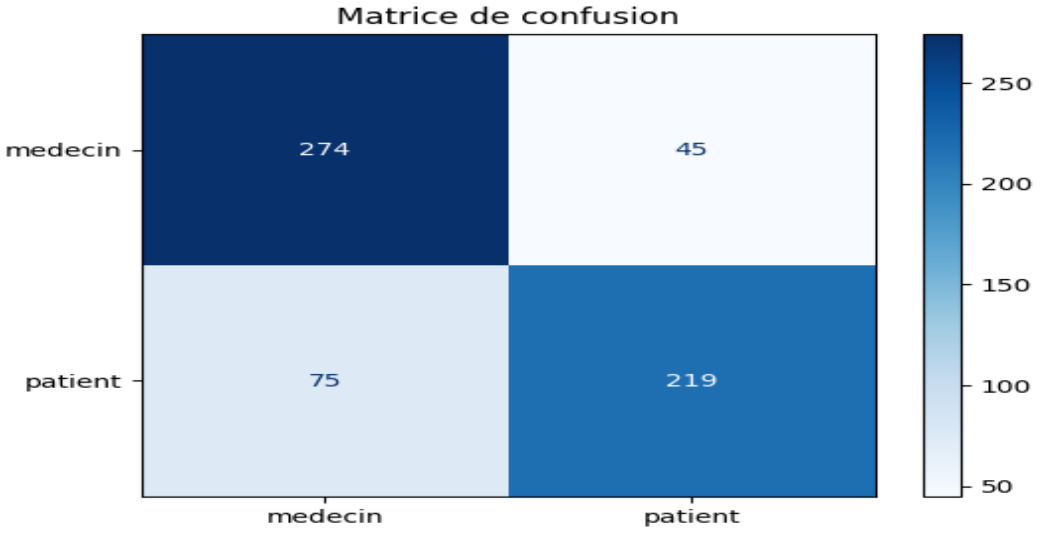

Figure illustrant la matrice de confusion de notre modèle de régression logistique

L'évaluation de la matrice de confusion sur l'ensemble de test montre que, parmi les 319 phrases (274+45) étiquetées comme provenant de médecins, 274 ont été correctement classifiées, tandis que 45 ont été incorrectement attribuées à la classe "patient". Pour les patients, sur un total de 294 (75+219) phrases, 219 ont été correctement identifiées, et 75 ont été mal classées comme provenant de médecins.

```
Meilleur score CV : 0.7762

Résultats sur l'ensemble de test:
Accuracy: 0.8042

Rapport de classification:
              precision    recall  f1-score   support

     medecin       0.79      0.86      0.82       319
     patient       0.83      0.74      0.78       294

    accuracy                           0.80       613
   macro avg       0.81      0.80      0.80       613
weighted avg       0.81      0.80      0.80       613
```

Figure illustrant la métrique de performance de notre modèle de régression logistique

En termes de métriques globales, le modèle atteint une précision de 80,42%, dépassant ainsi le seuil fixé de 80%. La précision et le rappel pour la classe "médecin" s'élèvent respectivement à 0,79 et 0,86, avec un F1-score de 0,82. Pour la classe "patient", ces valeurs sont de 0,83 pour la précision, 0,74 pour le rappel, et 0,78 pour le F1-score. Ces résultats traduisent une performance équilibrée entre les 2 groupes.

Évaluation du modèle BERT (CamemBERT)

Dans cette analyse, on évalue les performances du modèle à classer des phrases en fonction du rôle du locuteur (médecin ou patient). Pour cela, différents types de prompts sont testés :

- ***"default"*** : la phrase est donnée telle quelle.
- ***"question"*** : la phrase est présentée sous forme de question, comme "Est-ce que cette phrase vient d'un médecin ou d'un patient ?"
- ***"contexte"*** : la phrase est introduite avec un contexte, comme "Dans le contexte d'un appel au SAMU, analyser qui parle."
- ***"rôle"*** : la tâche est explicitement décrite, par exemple "Identifier le rôle du locuteur (médecin/patient) : [texte]."

Chaque prompt est testé avec trois températures (0.5, 1.0, et 2.0), influençant la variabilité des réponses du modèle. Cela génère 12 combinaisons (4 prompts × 3 températures). Après évaluation, les résultats sont comparés pour identifier la meilleure configuration (prompt + température offrant la meilleure précision) et la pire, afin d'expliquer l'écart entre leurs performances.
Ensuite, le meilleur résultat a été obtenu avec le prompt "default" à une température de 2.0 (précision = 0.8271),

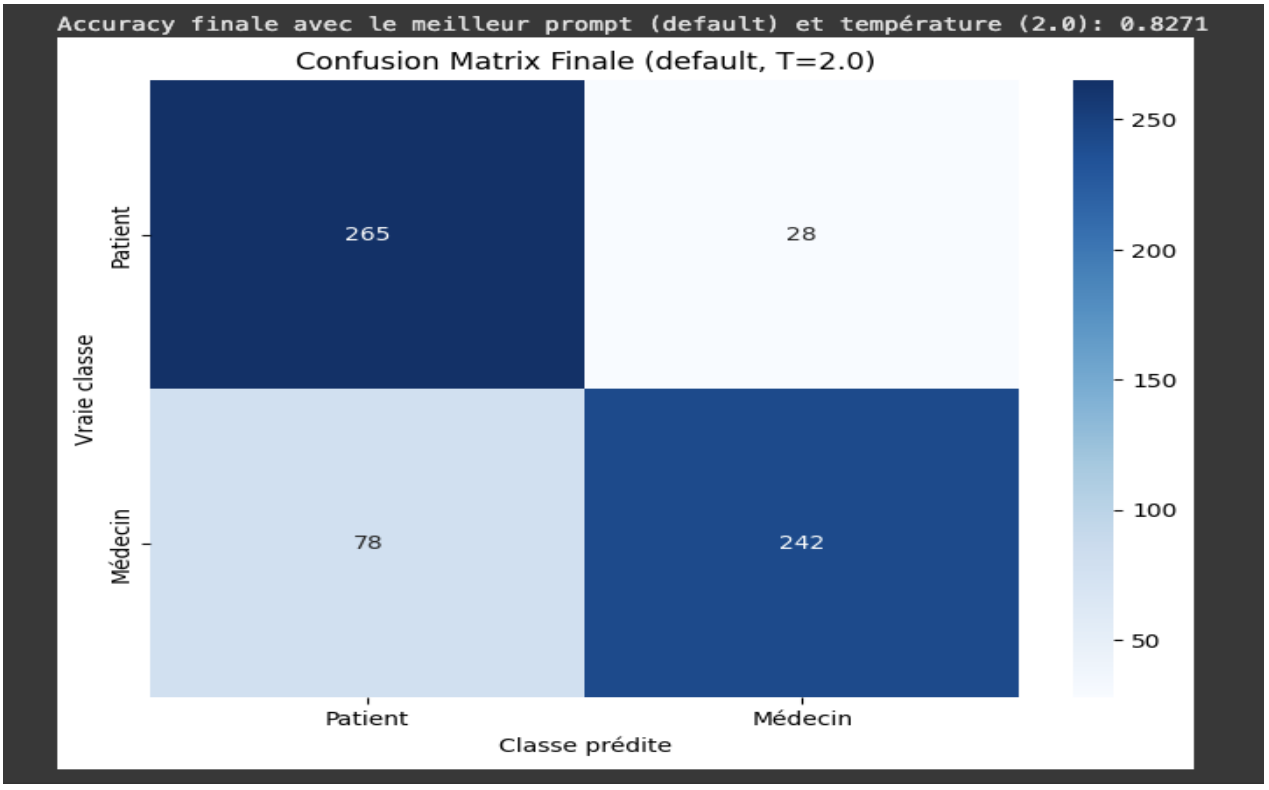

Figure illustrant la matrice de confusion de notre modèle BERT pour le meilleur cas

L'évaluation de la matrice de confusion sur l'ensemble de test montre que, parmi les 320 phrases (242+78) étiquetées comme provenant de médecins, 242 ont été correctement classifiées, tandis que 78 ont été incorrectement attribuées à la classe "patient". Pour les patients, sur un total de 293 (265+28) phrases, 265 ont été correctement identifiées, et 28 ont été mal classées comme provenant de médecins.

Le pire avec le prompt "question" à une température de 0.5 (précision : 0.5905) :

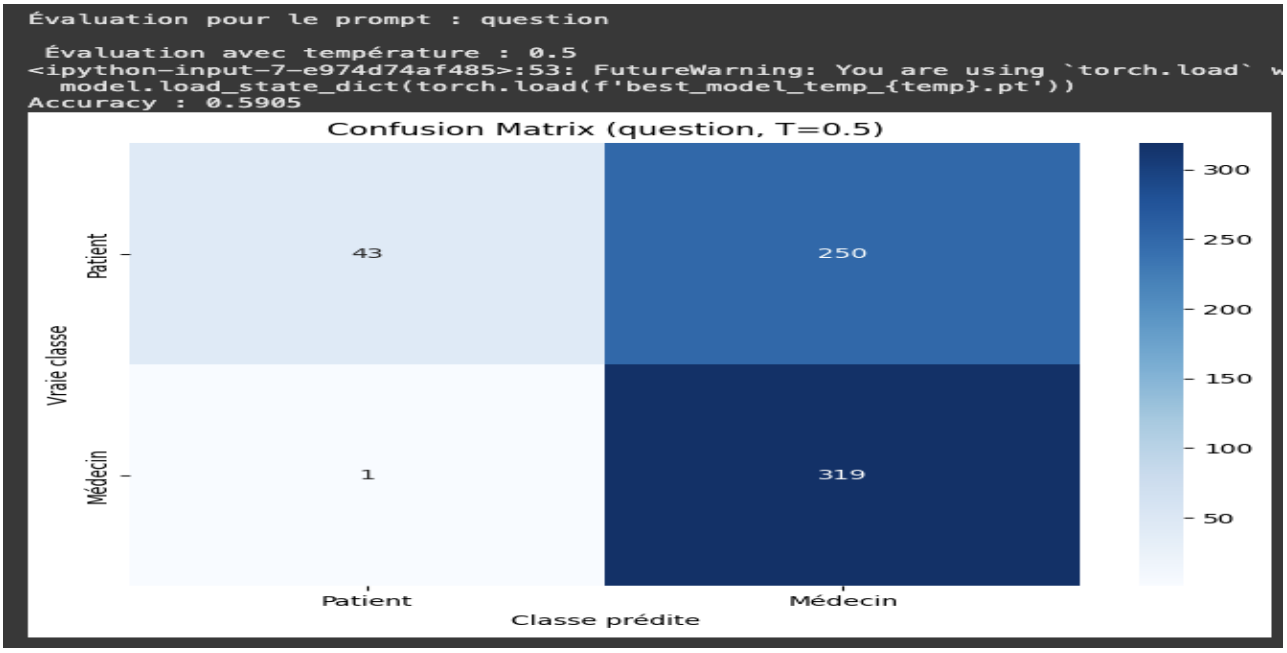

Figure illustrant la matrice de confusion de notre modèle BERT pour le pire cas

L'évaluation de la matrice de confusion sur l'ensemble de test montre que, parmi les 320 phrases (319+1) étiquetées comme provenant de médecins, 319 ont été correctement classifiées, tandis que 1 seule a été incorrectement attribuée à la classe "patient". Pour les patients, sur un total de 293 phrases (43+250), 43 ont été correctement identifiées, et 250 ont été mal classées comme provenant de médecins.

Discussion

Discussion sur les résultats de la régression logistique:

Les résultats présentés précédemment confirment l'efficacité de notre approche qui combinant la régression logistique et la vectorisation TF-IDF. La matrice de confusion met en évidence une performance légèrement supérieure pour la classification des médecins, ce qui peut s'expliquer par le caractère distinctif et formalisé du langage médical. En revanche, le taux de faux négatifs plus élevé pour la classe "patient" (75 contre 45 pour les médecins) révèle une tendance du modèle à sur-catégoriser certaines phrases en tant que "médecin".

Malgré cette disparité, les performances globales (80,42% de précision) valide notre modèle dans cette tâche de classification d'interlocuteur.

Discussion sur les résultats de BERT (CamemBERT) :

Les résultats présentés précédemment confirment l’efficacité du modèle BERT pour la classification des phrases en fonction du rôle du locuteur (médecin ou patient). La meilleure performance (82,71 % de précision) a été obtenue avec le prompt "default" à une température de 2.0. Cette configuration montre que le modèle exploite efficacement les représentations sémantiques sans être perturbé par des reformulations ou des ajouts contextuels.

La matrice de confusion met en évidence une performance équilibrée entre les 2 classes. Le modèle a correctement classé 265 phrases de patients et 242 phrases de médecins. Cependant, on note un taux de faux négatifs plus élevé pour la classe "médecin" (78 contre 28 pour les patients), révélant une légère tendance à classer des phrases médicales formelles comme provenant de patients.

En revanche, la pire performance (59,05 % de précision) a été obtenue avec le prompt "question" à une température de 0.5. Cette configuration a montré un fort déséquilibre, avec 250 phrases de patients mal classées comme provenant de médecins. Cela s'explique probablement par une surcharge cognitive introduite par la reformulation explicite en question, ainsi qu'une température basse limitant la diversité des prédictions.

Malgré ces variations, les performances globales du modèle BERT dans cette tâche (avec une précision optimale de 82,71 %) valident son efficacité, tout en soulignant l'importance du choix des prompts et des paramètres pour maximiser sa performance.

Comparaison des modèles de régression logistique (TF-IDF) et BERT

Les résultats des 2 modèles présentent des performances globales relativement proches, mais avec des comportements différents. Le modèle de régression logistique utilisant une représentation TF-IDF des textes, atteint une précision de 80,42%, tandis que le modèle BERT dans sa meilleure configuration (prompt "default" à température 2.0) obtient une précision de 82,71%.

Cette supériorité de BERT (CamemBERT) s'explique par sa capacité à saisir des nuances contextuelles plus subtiles, grâce à son architecture transformer et son pré-entraînement sur des textes en français. En revanche, la régression logistique semble plus sensible aux variations de la structure du texte, ce qui peut expliquer ses performances légèrement inférieures.

Une autre distinction notable entre les 2 modèles réside dans la gestion des faux négatifs, en effet, la régression logistique a tendance à surcatégoriser les phrases comme provenant de médecins (75 faux négatifs pour les patients), tandis que BERT présente un biais inverse avec 78 faux négatifs dans la classe "médecin". Ces divergences illustrent les différences fondamentales dans les approches des 2 modèles, la régression logistique se base sur une sélection de caractéristiques, alors que BERT bénéficie de sa compréhension contextuelle avancée et de ses mécanismes d'attention, ce qui lui permet de mieux s'adapter aux variations sémantiques présentes dans les dialogues du SAMU.

Conclusion

En conclusion, nous avons pu explorer l'identification d'interlocuteur avec l'exploitation d'un jeu de données et la conception de 2 modèles d'IA avec des approches différentes, une approche classique basée sur la régression logistique combinée à la vectorisation TF-IDF, et l'autre avec une approche utilisant un modèle de deep learning préentraîné BERT (CamemBERT).

Les 2 approches ont démontré leur efficacité en dépassant le seuil de performance fixé à 80%, la régression logistique a atteint une précision de 80,42% tandis que le modèle BERT (CamemBERT) a obtenu une précision de 82,71% dans sa meilleure configuration. Le modèle CamemBERT a des performances légèrement plus performantes (82,71% > 80,42%), ce qui s'explique par sa capacité à mieux comprendre le contexte des phrases et les subtilités de la langue française grâce à son pré-entraînement sur des textes français.

Le projet a également mis en évidence des différences dans les erreurs de classification des 2 modèles. La régression logistique avait tendance à trop classifier les phrases comme venant des médecins (75 faux négatifs pour les patients), tandis que notre BERT (CamemBERT) faisait plus d'erreurs sur les phrases des médecins (78 faux négatifs). Ceci montre que chaque modèle a ses forces et ses faiblesses. En effet, la régression logistique est rapide, mais peine à comprendre les contextes complexes, tandis que CamemBERT analyse mieux les nuances, mais consomme plus de ressources.

Pour la réalisation technique, nous avons travaillé sur Google Colab avec l'accès aux GPU gratuits limité. Cette contrainte a nécessité une attention particulière lors de leur utilisation, car le système peut s'interrompre en cas d'inactivité (pendant les entraînements) ou quand le temps d'utilisation du GPU est dépassé. Si cela arrivait, il fallait attendre un temps limité pour réavoir du GPU gratuit.

Pour l'avenir du projet, il serait intéressant d'augmenter la taille du jeu de données, avec, si possible, un nombre égal de phrases en "médecin" et "patient", il serait également intéressant d'améliorer les modèles, notamment pour notre modèle BERT (CamemBERT), il faudrait l'entraîner sur plus de textes avec plus d'exemples de langage médical formel pour mieux identifier les phrases des médecins, tandis que pour la régression logistique, on pourrait ajuster l'entraînement pour donner plus d'attention aux phrases des patients, ce qui limiterait la surclassification des phrases en "Médecin", tout ceci devrait permettre d'améliorer le taux de précision pour les 2 modèles.

Il serait également intéressant de créer des sous-classes pour l'interlocuteur "patient" afin d'identifier l'appelant ("patient", "conjoint", "mère", "père", "enfant", "patient qui est médecin"). Ce qui enrichit l'analyse, mais le jeu de données actuel ne permet pas de faire cette analyse.

Ce projet a visé à concevoir 2 modèles d'identification d'interlocuteur dans le cadre d'échanges fictifs avec le SAMU. Après avoir atteint un seuil de précision plus important, ces modèles pourraient être utilisés pour automatiser l'analyse des appels d'urgence, permettant ainsi de mieux orienter les équipes médicales en fonction des profils des interlocuteurs.

Références

Lien vers le Dataset : ***https://huggingface.co/datasets/medkit/simsamu***

Lien vers un répo github qui contient notre jeu de donnée :
https://github.com/manalland/simsamu

Lien vers la documentation de CememBERT :
https://huggingface.co/docs/transformers/en/model_doc/camembert

Lien qui nous a aidé pour TF-IDF :
https://medium.com/@claude.feldges/text-classification-with-tf-idf-lstm-bert-a-quantitative-comparison-b8409b556cb3