Utilisation de LLMs pour la classification d'avis client et comparaison avec une approche classique basée sur CamemBERT

N. Vautier[1], M. Héry[1], M. Miled[3], I. Truche[4], F. Bullier[3], A.L. Guénet[2]
G. Dubuisson Duplessis[2], S. Campano[1], P. Suignard[1]
[1] EDF Lab Paris Saclay, SEQUOIA
[2] EDF Commerce, Direction des Systèmes d'Information et du Numérique (DSIN)
[3] AI&Data
[4] EY

nicolas.vautier, marc.hery, guillaume.dubuisson-duplessis, anne-laure.guenet, sabrina.campano, philippe.suignard@edf.fr

Résumé

Les cas d'usage courants de la relation client chez EDF comme le routage d'un document à la bonne personne, la catégorisation de documents, font actuellement appel à une tâche de classification automatique supervisée. Dans ce contexte, cet article compare deux approches pour la classification de commentaires de satisfaction et e-mails de clients : une approche classique mise en production à EDF basée sur CamemBERT fine-tuné sur des données EDF et une approche plus récente basée sur des LLMs (Large Language Models). Pour cette 2ème approche, 3 stratégies de prompting sont testées (zero-shot, few-shot, et keyword prompting) avec plusieurs LLMs "open-weights" : Mistral, Mixtral, NeuralHermès et Phi3 sur des tâches de classification de référence de 2 types différents : classification binaire et multilabel. En plus de ces stratégies, l'impact du pré-traitement sur les textes en entrée des LLMs a été évalué, ainsi que l'apport de leur fine-tuning spécifiquement pour les tâches. S'il ressort de ces tests que les performances des LLMs non fine-tunés sont en deçà de celles des approches CamemBERT, l'étude apporte des enseignements sur l'impact du prompting, du pré-traitement des textes et l'apport du fine-tuning dans l'utilisation de ces modèles.

Mots-clés

Grands modèles de langue, ingénierie du prompt, classification de texte.

Abstract

Common customer relationship use cases at EDF, such as routing a document to the right person and document categorization, currently require a supervised automatic classification task. In this context, this article compares two approaches for the classification of satisfaction comments and customer emails : a classic approach put into production at EDF based on CamemBERT fine-tuned on EDF data and a more recent approach based on LLMs (Large Language Models). For this 2nd approach, 3 prompting strategies are tested (zero-shot, few-shot, et keyword prompting) with several open-weights LLMs : Mistral, Mixtral, NeuralHermès and Phi3 on reference classification tasks of 2 different types : binary classification and multilabel. In addition to these strategies, the impact of pre-processing on the LLM input texts was evaluated, as well as the contribution of their fine-tuning specifically for the tasks. If it appears from these tests that the performances of non-fine-tuned LLMs are below those of CamemBERT approaches, the study provides lessons on the impact of prompting, text preprocessing and the contribution of fine-tuning in the use of these models.

Keywords

Large language models, prompt engineering, text classification.

1 Introduction

Chaque mois, la gestion de la relation client chez EDF Commerce produit un volume important de données textuelles, issues tant des interactions des clients (par exemple, courriels et réponses ouvertes dans des enquêtes de satisfaction) que des observations formulées par les conseillers (telles que les commentaires de contacts). Ces informations, principalement rédigées en français, se caractérisent par leur richesse et diversité. Elles englobent une variété de formats, depuis des expressions spontanées jusqu'à des réponses plus structurées, comme celles enregistrées via des formulaires. De plus, elles révèlent une hétérogénéité notable en termes d'orthographe, de syntaxe et de niveau de langue qui soulève de véritables défis pour les explorer efficacement [7]. Ces données sont utilisées pour répondre au mieux aux attentes de nos clients en suivant le cadre réglementaire du « règlement général sur la protection des données » (RGPD) [6]. En outre, elles sont exploitées dans de nombreux cas d'usage visant à optimiser la relation client (e.g., sur le canal e-mail [8] ou sur le canal téléphonique [9]).

Les cas d'usage courants de la relation client comme le pilotage d'une activité basée sur des données texte (e.g., emails, réponses ouvertes à des enquêtes), le routage d'un document à la bonne personne, la catégorisation de documents font appel à une tâche de classification classique de l'apprentissage supervisé. La tâche de classification consiste à attribuer à chaque entrée de données une catégorie spécifique parmi un ensemble prédéfini. Une tâche de classification a le bénéfice d'être conceptuellement simple tout en offrant un cadre d'évaluation robuste via des métriques bien établies (comparativement moins subjectif que des tâches dites « génératives »).

L'avènement récent des "grands modèles de langue" (LLMs) comme GPT3, GPT4 [1], puis des modèles "open-weights", comme Mistral [14, 15], est en train de bouleverser le monde du traitement automatique du langage. Outre les nouvelles possibilités offertes par les LLMs (e.g., résumé de texte, chatbots, ...), ils sont également applicables sur des tâches de classification plus traditionnelles. A ce sujet, des résultats contradictoires ont été rapportés sur l'apport des LLMs pour les tâches de classification. D'un côté, les capacités d'apprentissage dans des scénarios en zéro/few-shot sont soulignées [16] tandis que de l'autre ces capacités sont remises en cause en raison de problèmes de contamination [18, 2, 5].

Les travaux présentés dans cet article visent à mieux cerner l'impact des LLMs sur des tâches de classification pour des cas d'usage industriels. Tout d'abord, nous nous intéressons à des tâches de classification sur des données privées qui ne sont pas sujettes à des problèmes de contamination. Cela nous permet d'évaluer de manière plus robuste le réel apport des LLMs sur des tâches de classification. Ensuite, nous comparons les LLMs à des modèles de référence actuellement employés industriellement, dont un modèle fondation similaire à CamemBERT [20] spécialisé sur les données de la relation client d'EDF Commerce. Nous présentons les principaux enseignements de cette comparaison, et nous discutons également de l'impact sur la méthodologie des projets impliquant des tâches de classification.

Le plan de l'article est structuré par les parties décrites ci-après. La Section 2 pointe les travaux connexes les plus saillants. Les données sont présentées en Section 3. La Section 4 forme le cœur de l'article, elle décrit les expérimentations réalisées et les résultats obtenus. Ces résultats sont discutés en Section 5, puis la Section 6 clôt cet article en soulignant les principales conclusions et en identifiant quelques perspectives prometteuses.

2 Travaux connexes

De manière très schématique, une approche classique pour classer des textes consiste à les représenter sous une forme vectorielle, que ce soit avec des sacs de mots (*bags of words*) ou avec des plongements de mots (*word embeddings*), puis à entraîner un classifieur de type régression logistique, SVM ou autre.

Les manières de réaliser ces plongements ont beaucoup évolué ces dernières années : dans un premier temps non contextuels, comme Word2Vec [21] ou GloVe [23], ils sont devenus contextuels, en s'appuyant sur l'architecture *transformer*, comme BERT [4], CamemBERT [20] ou FlauBERT [17]. De manière encore plus récentes, les LLMs offrent également une possibilité de produire des plongements [12], possibilité qui n'a pas été investiguée dans cet article.

L'arrivée des LLMs permet d'envisager de nouvelles tâches, notamment en sciences sociales [26], mais également de réaliser des tâches historiques de classification, ce que nous avons testé dans notre étude. Une récente approche consiste à demander directement au LLM via une instruction en langue naturelle dans quelle catégorie il classerait les documents fournis en entrée, avec une approche en *zero shot* (sans exemple) ou en *few shot* (avec des exemples) :

— Chae et Davidson [3] montrent que les résultats obtenus avec des LLMs (ici GPT 3) surpassent les approches traditionnelles de machine learning utilisant des *bags of words* et des *word embeddings*, mais qu'un BERT fine-tuné sur le corpus complet reste compétitif ;

— Sun et al. [24] obtiennent des résultats largement supérieurs avec des LLMs en utilisant une stratégie de prompt plus sophistiquée consistant à demander au LLM d'extraire les informations importantes du texte à classer (mots clés, phrases, informations contextuelles et sémantiques, etc.) avant d'en déduire la catégorie du texte.

Fields et al. proposent un état de l'art sur les LLMs utilisés pour la classification de texte [11]. Ils comparent les performances de différents modèles sur 15 tâches, et notent que les LLMs récents sont plus efficaces sur la tâche de question / réponse, tandis que sur des tâches de classification de texte les modèles les plus performants incluent des modèles BERT mais également des modèles qui ne sont pas basés sur une architecture *transformer*. Dans tous les cas, une des difficultés consiste à bien formuler l'instruction ou *prompt* à adresser au LLM, comme le rapportent Liu et al. [19] dans leur revue des différentes méthodes de *prompting*.

Contrairement aux travaux précédents qui portent sur des corpus en anglais, notre étude se focalise sur des données privées composées d'avis clients au sens large, écrits en français, ce qui constitue un cas d'usage réel, avec toutes les particularités que cela peut représenter (fautes d'orthographe, écrits parfois de manière télégraphique, etc.).

3 Données

Nous présentons dans cette section les deux grands types de données utilisées pour cette expérimentation : des données issues de réponses à des enquêtes de satisfaction, et des données issues d'e-mails clients à destination d'un conseiller. Nous présentons également la procédure de désidentification appliquée à ces données.

3.1 Données de satisfaction

Le corpus est composé d'un échantillon d'un peu plus de 3000 réponses ouvertes à une enquête de satisfaction. En

effet, après clôture d'une demande ou réclamation, certains clients sont invités à répondre à plusieurs questions. Dans ce questionnaire, le client doit, entre autres, indiquer son niveau de satisfaction puis répondre à une question en fonction de celui ci. Si le client est "très satisfait" ou "pas satisfait", la question posée sera : "Quelles sont les raisons de votre satisfaction/insatisfaction ?". Si le client est "assez satisfait" ou "peu satisfait", la question posée sera : "Qu'aurait pu faire votre conseiller pour améliorer votre satisfaction ?". C'est lors de cette réponse que le client aborde potentiellement les irritants (thématiques) que l'on veut détecter. Afin de préparer l'entraînement de modèles pour la détection de ces irritants clients, une double annotation avec ré-annotation en cas de désaccord a été réalisée. La procédure d'annotation était multilabel, impliquant ainsi l'attribution d'une ou plusieurs des 13 thématiques existantes à chaque réponse. Cette classification nous aide, in fine, à mieux comprendre les irritants exprimés et permet donc l'amélioration continue de l'expérience client.

Pour cet article, 4 thématiques ont été retenues, présentant des niveaux de difficulté variable sur la tâche de classification. Des classifications binaires (présence ou non du label correspondant à la thématique) et multilabels sont à la fois réalisées. Les jeux de données binaires sont indiqués par le nom suivant : "Satisfaction-b2b-[thématique]", le jeu de données multilabel par le nom "Satisfaction-b2c-MULTI". Les thématiques retenues pour l'étude et les caractéristiques des jeux de données associés sont décrits dans le tableau 1. Ces thématiques correspondent à des plaintes quant au traitement de la demande par le service client. COUPURE DE COMMUNICATION : Le client se plaint d'avoir subi une coupure téléphonique pendant l'appel. PROBLEME NON RESOLU : le client se plaint du fait que le problème n'est toujours pas résolu malgré la clôture de la demande. REPONSE NON DESIREE : la réponse est légitime du point de vue de EDF, mais celle-ci ne satisfait pas le client. SUIVI DE LA DEMANDE : manque de communication à propos de l'évolution de la demande client.

3.2 Données d'e-mails

Un corpus d'e-mails est considéré. Il s'agit d'e-mails entrants écrits par un client à destination d'un conseiller. Ce jeu de données est utilisé pour améliorer le routage et le pilotage de l'activité. Les sollicitations clients dans ces e-mails sont très variées de par leur nature, leur vocabulaire ou encore leur taille. Le corpus est composé d'un peu moins de 6000 e-mails annotés et va servir à modéliser le ressenti dans un e-mail. Le ressenti est représenté par trois catégories. La catégorie URGENCE dans laquelle le client demande une action rapide du conseiller, la catégorie RELANCE dans laquelle le client relance suite à une première demande sans réponse, la catégorie MÉCONTENTEMENT dans laquelle le client exprime son mécontentement quant à une situation donnée. Une double annotation a été réalisée pour parvenir à ce premier corpus avec réannotation des désaccords. La procédure d'annotation était multilabel permettant d'attribuer plusieurs ou aucun ressenti à un même e-mail.

Les jeux de données sont traités comme des problèmes de classification binaire (leur nom suit la forme "Ressenti-b2b-[thématique]") et multilabel ("Ressenti-b2b-MULTI"). Les caractéristiques du jeu de données "Ressenti-b2b" sont présentées dans le tableau 1.

3.3 Désidentification des données texte

En vertu du règlement général sur la protection des données (RGPD), l'ensemble des données texte a été désidentifié et ne comportent aucune donnée à caractère personnel telles que des noms, prénoms, adresses, numéros [6]. Les données à caractère personnel sont substituées par le type de l'entité (e.g., "Je suis Monsieur *[person]* (numéro de client : *[num]*), et je vous écris pour manifester mon mécontentement au sujet de ma dernière facture pour mon logement situé à *[localisation]*").

4 Méthodes et expérimentations

4.1 Méthodes

4.1.1 Modèles

Pour évaluer la qualité des classifications obtenues par les approches LLM, nous cherchons à les comparer avec l'approche de référence : un modèle CamemBERT spécifiquement entraîné pour la classification sur chaque jeu de données.

Modèle baseline CamemBERT Le modèle CamemBERT EDF Commerce est un modèle pré-entraîné sur la base du modèle CamemBERT [20] et spécialisé pour la gestion de la relation client de la direction Commerce d'EDF. Le modèle a été pré-entraîné sur un ensemble de données textes désidentifées en français provenant de différentes sources internes (e.g., emails, réponses libres à des enquêtes de satisfaction, commentaires conseiller) reflétant les nuances et la complexité de la communication dans notre secteur d'activité. Il est important de noter que les données à caractère personnel directement identifiantes ont été exclues du pré-entraînement (e.g., nom, prénom, localisation, numéro client). Le pré-entraînement a été réalisé en utilisant la technique de masquage aléatoire (MLM) qui permet au modèle d'apprendre les relations sémantiques entre les mots et les phrases.

LLMs considérés Quatre LLMs ont été utilisés pour chaque expérimentation :

- **Mistral-7B-Instruct-v0.1** [14], le premier LLM produit par la société Mistral, spécialisé sur un jeu de données d'instructions conversationnelles, sorti en septembre 2023, avec une taille de 7 milliards de paramètres ;
- **NeuralHermes-2.5** [1], un modèle fine-tuné à partir de Mistral-7B, réalisé par le français Maxime Labonne ;
- **Mixtral-8x7B-Instruct-v0.1** [15], de la société Mistral, un modèle composé d'une mixture de 8 experts, sorti en décembre 2023 ;

1. https://huggingface.co/mlabonne/NeuralHermes-2.5-Mistral-7B

Jeu de données	Type	Nb. classes	Nb. documents	Nb. mots (moy.)	Ratio de déséquilibre
Ressenti-b2b-MECONTENTEMENT	binaire	2	4494	82 +/- 93	0.04
Ressenti-b2b-RELANCE	binaire	2	4494	82 +/- 93	0.07
Ressenti-b2b-URGENCE	binaire	2	4494	91 +/- 100	0.07
Satisfaction-b2c-COUPURE_DE_COMMUNICATION	binaire	2	2553	27 +/- 35	0.03
Satisfaction-b2c-PROBLEME_NON_RESOLU	binaire	2	2610	27 +/- 33	0.18
Satisfaction-b2c-REPONSE_NON_DESIREE	binaire	2	2678	29 +/- 37	0.14
Satisfaction-b2c-SUIVI_DE_LA_DEMANDE	binaire	2	2611	28 +/- 36	0.13
Ressenti-b2b-MULTI	multilabel	4	4494	91 +/- 100	0.04
Satisfaction-b2c-MULTI	multilabel	5	3810	28 +/- 37	0.02

TABLE 1 – Résumé des jeux de données utilisés pour les expérimentations. Le nombre moyen de mots correspond au nombre moyen de mots calculés sur l'ensemble des documents de chaque corpus (+/- l'écart type). Le ratio de déséquilibre est calculé comme le rapport entre le nombre de documents appartenant à la plus petite classe et le nombre total de documents du corpus.

— **Phi-3-mini-128k-instruct** [10], modèle léger à l'état de l'art publié par Microsoft ;

Nous avons sélectionné ces modèles en raison de leurs bonnes performances sur des cas d'usage métier EDF, notamment sur des tâches de question réponse et de résumé. Ces derniers sont sortis en *open-weights* à la fin de l'année 2023 et au début de l'année 2024, cela signifie que les poids de ces modèles sont accessibles, contrairement aux données qui ont servi à leur entraînement. Cela permet de les utiliser sur infrastructure de calcul interne, et de préserver la confidentialité des données. Par ailleurs, utiliser les versions les plus "légères" de ces modèles à 3.8 et 7 Md de paramètres, permet de se rapprocher d'une utilisation en contexte industriel, où les moyens de calcul et les temps requis pour exécuter les traitements sont plus contraints.

4.1.2 Découpage entraînement/validation/test

Pour pouvoir comparer les métriques entre la baseline CamemBERT et les LLMs, les découpages déjà utilisés pour l'entraînement et l'évaluation des modèles CamemBERT ont été repris dans la mesure du possible. Pour les expérimentations binaires, les plis ont pu être repris tels quels, avec une distribution stratifiée : avec un ratio apprentissage/test de 85%/15% pour les jeux de données d'emails et 80%/20% pour les jeux de données de satisfaction. Pour les expérimentations multilabel, les plis utilisés pour CamemBERT n'étant pas disponibles, une stratification itérative a été appliquée dans les mêmes proportions afin de conserver la même distribution de labels unitaires et des associations de labels entre jeux de données d'apprentissage et de test. On notera qu'il n'est pas pertinent d'avoir un ensemble de validation pour les évaluations des LLMs non fine-tunés, puisque seule l'inférence sur le jeu de test nous permet de se comparer à la baseline Camem. Pour les expérimentations de fine-tuning des LLMs en revanche, il a été nécessaire de redécouper l'ensemble d'apprentissage pour obtenir un ensemble de validation : le ratio du pli est alors fixé à 85%/15% et le découpage se fait par stratification simple (cas binaire) ou itérative (cas multilabel).

4.1.3 Métriques d'évaluation considérées

Chaque jeu de données étant déséquilibré, seules des métriques adaptées au déséquilibre des classes ont été utilisées. Les métriques choisies sont identiques à celles utilisées pour l'évaluation des modèles CamemBERT industrialisés. Pour les classifications binaires le coefficient de corrélation de Matthews (MCC) et le F1-score ont été utilisés. Pour les classifications multilabel, seul le F1-score a été utilisé, le MCC n'étant pas défini. Pour le cadre multilabel, le F1-Score a été calculé à l'aide de ScikitLearn [22] dans sa version "macro" calculant un F1-score global obtenu comme une moyenne des F1-score des différentes classes.

4.1.4 Cadre d'évaluation

De nombreuses évaluations ayant été nécessaires, nous avons choisi les librairies python Luigi [2] pour la chaîne de traitement, plis, entraînement et évaluation, ainsi que de la librairie MLFlow [25] pour l'enregistrement des paramètres et des métriques résultant de chaque expérimentation. Concernant l'environnement de calcul, nous avons utilisé 3 GPUs d'une plateforme Nvidia DGX H100 80Go.

4.2 Expérimentations

Nous présentons dans cette section les 3 stratégies explorées lors des expérimentations menées. Elles sont évaluées sur chaque modèle décrit Section 4.1.1, et chaque jeu de données décrit Section 3. Ces 3 stratégies sont les suivantes : quelle méthode de *prompting* utiliser, quel prétraitement appliquer sur les textes et enfin, quel gain peut-on espérer en spécialisant ces modèles avec du *fine-tuning*.

4.2.1 Stratégies de prompting

Le *prompt* désigne l'ensemble des instructions formulées en langage naturel en entrée d'un LLM afin d'obtenir le résultat désiré. Nous cherchons à concevoir un *prompt* qui contienne le maximum d'informations tout en étant le plus clair possible. Le *prompt* doit présenter le problème de manière concise, énoncer les instructions d'entrée et de sortie

2. https://github.com/spotify/luigi

au LLM et présenter ou non des exemples en fonction de la stratégie de *prompting* adoptée.
Dans le cadre de notre étude, 3 stratégies ont été retenues : *zero-shot*, *few-shot*, et *keyword prompting*. Nous souhaitons mesurer l'impact de celles-ci sur les performances de la classification.

Zero shot prompting Le *zero-shot prompting* consiste à demander au LLM de classifier un texte sans lui donner d'exemple. Seulement une définition de la catégorie est utilisée. Après plusieurs essais, la stratégie *zero-shot* que nous avons retenue est la suivante :
"Tu dois répondre par oui ou par non afin de savoir si un client a fait l'objet de [thématique] ou non. Par [thématique, on veut dire : [définition]. Voici ce qu'a dit le client : [Verbatim du client]. Ce client a-t-il fait l'objet de [thématique] ?"

Keywords prompting La stratégie de *keyword prompting* diffère du *zero-shot prompting* par l'ajout de mots clés :
"Tu dois répondre par oui ou par non afin de savoir si un client a fait l'objet de [thématique] ou non. Par [thématique, on veut dire : [définition]. Si ce que le client dit est en rapport avec un des termes de la liste alors il y a [thématique] : [liste de mots clés]. Si ce que le client dit n'est pas en rapport avec un des termes de la liste précédente alors tu dois répondre non. Voici ce qu'a dit le client : [Verbatim du client]. Ce client a-t-il fait l'objet de [thématique] ?"
Voici un exemple de prompt pour la catégorie "Coupure de communication" :
"Tu dois répondre par oui ou par non afin de savoir si un client a fait l'objet d'une coupure de la communication ou non. Par coupure de la communication, on veut dire que le conseiller a raccroché au nez du client ou que la conversation a été interrompue à cause d'un problème technique. Si ce que le client dit est en rapport avec un des termes de la liste alors il y a coupure de communication : "communication coupée", "coupure téléphonique", "coupure communication", "raccroche tout seul", "raccroche au nez", "téléphone coupe", "conseiller raccroche". Si ce que le client dit n'est pas en rapport avec un des termes de la liste précédente alors tu dois répondre non. Voici ce qu'a dit le client : 'Verbatim du client'. Ce client a-t-il fait l'objet d'une coupure de la communication ?"

Few shot prompting La dernière stratégie de *prompting* vient en complément de la précédente. En plus d'ajouter des mots clés, on insère également dans le prompt des exemples concrets de commentaires client issus de notre plan d'annotation.
Voici un exemple pour la catégorie "Mécontentement" :
"Tu dois répondre par oui ou par non afin de savoir si un mail client évoque un mécontentement. Le mécontentement est une situation exprimée comme désagréable par le client et pour laquelle il n'attend pas forcément une explication, une solution ou toute autre forme de reconnaissance. Si ce que le client dit est en rapport avec un des termes de la liste alors il y a mécontentement : "mécontent", "résiliation", "concurrence", "ironie", "jugement négatif", "menaces", "Ce n'est pas normal de votre part !", "Je vous demande donc de me rembourser les sommes que votre entreprise m'a indûment prélevées sur mon compte bancaire.", "Aujourd'hui nous recevons ENCORE une facture avec toujours l'ancienne adresse e faire le nécessaire AU PLUS VITE", "Pourriez vous faire ces modifications dans les plus brefs délais afin que je garde de bonne relation avec edf entreprises". Si ce que le client dit n'est pas en rapport avec un des termes de la liste précédente alors tu dois répondre non. Voici ce que dit le client : 'Verbatim du client'. Ce client évoque-t-il un mécontentement ?"

Prompting en multilabel Dans le cadre d'une classification multilabel, nous réutilisons les 3 stratégies précédentes pour la description des classes à détecter, en précisant que tout texte n'appartenant pas aux classes définies doit être catégorisé "AUTRE". Ainsi, pour le cas multilabel, le modèle n'est pas tenu de choisir un label quand aucun ne correspond aux données. Dans le cas multilabel, l'instruction passée en prompt est : "Chaque texte peut appartenir à plusieurs classes. Donne le résultat sous la forme d'une liste, par exemple : [CLASS_X, CLASS_Y].".

4.2.2 Pré-traitement des textes

Les textes de nos jeux de données étant déjà anonymisés, les entités nommées telles que les personnes physiques et morales, les montants, les numéros de téléphone etc. ont au préalable été remplacées par des balises d'anonymisation. Ainsi, la phrase : "une baisse d'un montant de 10 euros pour le contrat du site de Bordeaux" est présente dans nos corpus sous la forme : "une baisse d'un montant de __MONTANT__ pour le contrat du site de __LOCALISATION__". Ce pré-traitement pouvant impacter la compréhension du LLM, nous avons expérimenté un pré-traitement effectuant l'opération inverse, en remplaçant ces balises par des entités toujours factices mais vraisemblables pour le modèle. Ainsi, si le pré-traitement est appliqué, la phrase précédente sera sous la forme suivante : "une baisse d'un montant de 4 euros pour le contrat du site de Paris".

4.2.3 Fine-tuning

Pour évaluer le potentiel d'amélioration des performances des LLMs sur ces expérimentations, une spécialisation des modèles (*fine-tuning*) a été réalisée sur chaque jeu de données. Pour cela, nous avons constitué nos jeux d'apprentissage en encapsulant les exemples des ensembles d'apprentissages initiaux dans les prompts des 3 stratégies de *prompting* décrites précédemment. Le résultat attendu est ensuite concaténé au prompt afin de former la base d'apprentissage. Le jeu de test est formé en prenant le prompt indiquant la tâche de catégorisation et le texte à catégoriser, sans ajouter le résultat, que le modèle devra prédire correctement. Notre jeu de données est divisé en deux pour obtenir les jeux d'apprentissage (85%) et de validation (15%) qui serviront à *fine-tuner* nos modèles. Pour ces expérimentations de *fine-tuning*, la même méthode a été appliquée avec le même paramétrage. La méthode LoRa a été ici utilisée sur les modules d'attention des modèles pour accélérer l'apprentissage avec le moins de perte de performance possible, comme préconisé par les auteurs de [13]. Lors de l'apprentissage, un même taux d'apprentissage ($10e{-}4$) et un nombre variable d'époques (entre 3 et 8) ont été utili-

		Binaire				Multilabel			
	Fine-tuning	Zero-shot	Keyword	Few-shot	Baseline	Zero-shot	Keyword	Few-shot	Baseline
Mistral-7B	Non	0.38	0.35	0.35	/	0.22	0.21	0.24	/
	Oui	0.51	0.67	<u>0.72</u>	/	0.58	0.58	**0.64**	/
NeuralHermes-2.5	Non	0.38	0.48	0.51	/	0.29	0.29	0.39	/
	Oui	0.41	0.44	0.42	/	0.52	<u>0.60</u>	0.57	/
Mixtral-8X7B	Non	0.44	0.54	0.51	/	0.40	0.50	0.42	/
	Oui	-	-	-	/	-	-	-	/
Phi-3-mini-128k	Non	0.28	0.33	0.40	/	0.23	0.28	0.29	/
	Oui	0.53	0.60	0.65	/	0.53	0.58	0.55	/
CamemBERT	/	/	/	/	**0.79**	/	/	/	0.50

TABLE 2 – Résultats des différentes expérimentations : binaire = tâches de classification binaire, multilabel = tâches de classification multilabel. Chaque cellule correspond à la moyenne des scores obtenus sur tous les jeux de données pour une configuration (modèle fine-tuné ou non + tâche + stratégie de prompting). La métrique utilisée est le F1-score. En gras /souligné : meilleur score /second meilleur score par type de classification. CamemBERT représente le modèle baseline CamemBERT EDF Commerce.

		Ressenti-b2b-BINAIRE				Ressenti-b2b-MULTI				Satisfaction-b2c-BINAIRE				Satisfaction-b2c-MULTI			
	Fine-tuning	Zero-shot	Keyword	Few-shot	Baseline	Zero-shot	Keyword	Few-shot	Baseline	Zero-shot	Keyword	Few-shot	Baseline	Zero-shot	Keyword	Few-shot	Baseline
Mistral-7B	Non	0.40	0.34	0.42	/	0.28	0.27	0.28	/	0.37	0.36	0.30	/	0.17	0.15	0.20	/
	Oui	0.78	0.65	<u>0.79</u>	/	<u>0.80</u>	0.77	**0.83**	/	0.31	<u>0.69</u>	0.67	/	0.36	0.38	**0.44**	/
NeuralHermes-2.5	Non	0.39	0.41	0.52	/	0.40	0.37	0.51	/	0.37	0.52	0.49	/	0.18	0.22	0.24	/
	Oui	0.57	0.64	0.58	/	0.77	**0.83**	0.70	/	0.29	0.29	0.30	/	0.28	0.38	**0.44**	/
Mixtral-8X7B	Non	0.49	0.59	0.55	/	0.35	0.45	0.31	/	0.40	0.51	0.49	/	0.24	0.27	0.26	/
	Oui	-	-	-	/	-	-	-	/	-	-	-	/	-	-	-	/
Phi-3-mini-128k	Non	0.29	0.32	0.39	/	0.25	0.33	0.31	/	0.28	0.34	0.41	/	0.21	0.22	0.27	/
	Oui	0.63	0.70	0.67	/	0.69	0.76	0.69	/	0.46	0.52	0.64	/	0.37	0.40	<u>0.41</u>	/
CamemBERT	/	/	/	/	**0.81**	/	/	/	0.79	/	/	/	**0.77**	/	/	/	0.22

TABLE 3 – Résultats détaillés : binaire = tâches de classification binaire, multi = tâches de classification multilabel. Chaque cellule correspond à la moyenne des scores obtenus pour une configuration donnée. Par exemple, Ressenti-b2b-BINAIRE correspond à la moyenne des scores obtenus sur les jeux de données "mécontentement", "relance" et "urgence". La métrique utilisée est le F1-score. En gras /souligné : meilleur score /second meilleur score par type de classification. CamemBERT représente le modèle baseline CamemBERT EDF Commerce.

sés. Le modèle ayant obtenu le meilleur score sur le jeu de validation est conservé.

4.3 Résultats

Analyse globale Les résultats de la classification binaire et multilabel sont présentés dans le Tableau 2 qui contient la moyenne des résultats sur tous les jeux de données par stratégie de prompt.
Les meilleurs résultats sont mis en évidence pour chaque type de classification (binaire et multilabel) et sont à comparer avec le modèle baseline CamemBERT représentant le modèle CamemBERT EDF Commerce. Pour la tâche de classification binaire, la baseline est nettement supérieure avec un F1-score de 0.79 contre 0.72 pour la version fine-tunée de Mistral-7B sur la stratégie *few-shot*. En classification multilabel, Mistral-7B fine-tuné en *keyword* obtient un F1-score de 0.64, soit 14 points de plus que la baseline CamemBERT. A noter que Phi-3-mini-128k fine-tuné maintient un bon niveau sur les prompts keyword et few-shot par rapport à sa taille.

Impacts de la stratégie de prompting Les stratégies de *keyword prompting* et *few-shot prompting* ont affiché des performances supérieures par rapport au *zero-shot prompting* avec en moyenne un gain de 3.2 points pour la partie non fine-tunée, et un gain de 5.3 points pour la partie fine-tunée. L'ajout de mots clés et/ou d'exemples a donc un impact sur la capacité des LLM à classifier correctement les verbatims client.

Impacts du fine-tuning En moyenne, le fine-tuning a augmenté les performances de nos tâches de classification, toutes configurations confondues, de 18 points. Il est important de noter que le fine-tuning du modèle NeuralHermes-2.5 n'a pas fonctionné pour toutes les configurations de la classification binaire. Le fine-tuning du modèle est difficile et le processus de fine-tuning fait apparaître une divergence au niveau de la fonction objectif pour certains cas. Nous ne sommes pas parvenus à en identifier la cause.

Impacts du pre-processing L'utilisation de notre prétraitement des textes n'a pas eu d'impact sur la moyenne des résultats obtenus. Il faudra réaliser une étude plus approfondie afin de préciser ces résultats.

Analyse détaillée Les résultats détaillés sont présentés dans le Tableau 3. Pour les 2 jeux de données (Ressenti et Satisfaction), le fine-tuning améliore les performances des modèles, sauf pour NeuralHermes-2.5 sur le corpus de Satisfaction. Le jeu de données Satisfaction-b2c-MULTI donne les moins bon résultats avec 0.22 de f1-score pour la baseline et 0.44 pour les meilleurs LLM. Les meilleures performances sont atteintes par Mistral-7B et NeuralHermes-2.5 (0.83 de F1-Score) sur Ressenti-b2b-MULTI. Plusieurs hypothèses peuvent expliquer ces différences : la taille des jeux de données, la longueur des documents qui les constitue, l'intention derrière le texte rédigé par le client (mail libre ou réponse à une question) mais aussi par la nature même des labels. En effet, les 3 labels du corpus de Ressenti sont très corrélés entre eux de manière positive. Un mail comportant un caractère de relance aura de forte chance d'apparaître avec un mail à caractère d'urgence etc. Ce qui n'est pas le cas pour les 4 labels étudiés sur le corpus de Satisfaction.
En résumé, nous avons montré la capacité des LLMs à se rapprocher, voire dépasser, les scores obtenus avec la baseline. Cependant la volatilité des résultats reflète la difficulté à désigner une stratégie de prompting comme étant la stratégie générique à appliquer pour chaque jeu de données. On retiendra la capacité des LLMs à être plus performants sur des tâches jugées plus complexes avec plusieurs labels/classes comparés à CamemBERT.

5 Discussion

Les résultats obtenus dans la section précédente montrent des performances pour les LLMs inférieures au modèle CamemBERT Commerce pour les tâches de classification binaire. Cela peut trouver plusieurs explications.

Tout d'abord, dans le cadre de l'utilisation d'un LLM non fine-tuné, concevoir un "bon" prompt pour une tâche donnée n'est pas intuitif. Il s'agit d'un processus itératif ayant demandé plusieurs essais. Nous avons constaté par expérimentation que tout doit être explicité dans le prompt pour obtenir de bons résultats, et que des détails importants pour la classification peuvent facilement être omis. Cette utilisation se différencie d'une tâche de classification avec un modèle fine-tuné sur des données de référence, où des connaissances sont implicitement contenues dans les classes attribuées aux données d'entrainement, et n'ont donc pas besoin d'être explicitées. Cette observation est étayée par le fait qu'ajouter des exemples (*few-shot*) ou des mots-clés (*keyword*) dans le prompt conduit à de meilleurs résultats qu'une approche en *zero-shot*.

Les résultats obtenus pourraient également avoir été impactés par la fragilité des LLMs face à la forme du prompt. De légères variations dans l'instruction (ex : minuscules vs majuscules) induisent des résultats différents. De même, le format de la séquence de texte prédite en sortie des modèles présente des variabilités, même lorsque le format de sortie souhaité est explicité dans l'instruction fournie en entrée au modèle. Par exemple, il a été observé plusieurs noms de classes prédites contenant des caractères indésirables qui compliquent le parsing du résultat : "SOUSCRIPTION_CONTRAT" (Mixtral) ou "SOUSCRPITION_CONTRACT" (NeuralHermes) au lieu du nom précisé dans l'instruction "SOUSCRIPTION_CONTRAT". De même, l'instruction "Donne le résultat sous la forme d'une liste, par exemple : [CLASS_X, CLASS_Y]." peut aboutir dans de très rares cas au résultat "CLASS_X" au lieu d'un nom de classe attendu.

Concernant la comparaison des performances des LLMs à celles du modèle métier CamemBERT EDF Commerce, des différences dans l'entrainement des modèles pourraient avoir avantagé le modèle métier. Le modèle CamemBERT EDF Commerce a été pré-entrainé sur un grand volume de données métier EDF, ce qui pourrait l'avoir favorisé au détriment des LLMs malgré leur taille, y compris dans la configuration de fine-tuning. Les LLMs n'ont en effet pas été pré-entrainés sur données métier avant d'être fine-tunés sur la tâche. En bref : spécialiser des modèles plus petits semble offrir de meilleurs résultats que les LLMs non fine-tunés testés dans ces travaux.

Enfin, sur l'aspect multilabel, il est important de préciser que les modèles CamemBERT multilabels ne sont pas utilisés de façon opérationnelle à cause de leur plus faible performance (pouvant s'expliquer par de possibles corrélations entre les classes, ou encore par le problème de stratification multilabel sur des classes déjà fortement déséquilibrées). Au lieu de cela, un système composé de plusieurs classifieurs binaires leur est préféré, présentant de meilleures performances mais ayant également l'inconvénient d'être plus difficile à maintenir en production. Les résultats des LLM finetunés sur les tâches multilabel montrent qu'il est possible d'obtenir des performances presque équivalentes avec un seul modèle, un avantage de taille pour une utilisation en production.

6 Conclusion et perspectives

Cet article nous a permis de tester des LLMs et de comparer les résultats obtenus sur une tâche de classification avec une chaîne de traitement en production basée sur un modèle CamemBERT finetuné sur des données Commerce. De ces différentes expérimentations, nous pouvons conclure que :

— CamemBERT EDF obtient les meilleurs résultats en classification binaire, même s'il existe presque toujours une configuration avec LLM qui obtient un résultat équivalent (notamment en fine-tunant le LLM). En classification multilabel, ce sont les LLMs fine-tunés qui se montrent les plus performants, illustrant peut-être la plus grande capacité de généralisation de ces modèles.

— Les scores obtenus par les LLMs sont proches du modèle CamemBERT Commerce, surtout quand ils sont

fine-tunés. Par contre, nous observons un impact faible des stratégies utilisées (pré-traitement et prompting *zero shot*, *keyword* et *few shot*), l'avantage des LLMs étant leur performance correcte quelles que soient ces stratégies. Leur inconvénient est la difficulté à trouver une stratégie pour améliorer ces résultats.

— Trouver le meilleur *prompt* possible est une tâche complexe qui nécessite parfois plusieurs essais.

Pour améliorer les premiers résultats obtenus, plusieurs perspectives s'offrent à nous. Tout d'abord nous pourrions fine-tuner le modèle Mixtral-8x7B, ce que nous n'avons pas fait par manque de temps. Ensuite, il serait possible de poursuivre le pré-entraînement des LLMs sur les données Commerce, comme EDF Commerce l'a fait pour spécialiser CamemBERT et produire le CamemBERT EDF. Des améliorations pourraient être apportées sur la construction des prompts, notamment au niveau du formatage de la sortie du LLM afin de mieux exploiter ses réponses. Plusieurs options de fine-tuning peuvent également être envisagées pour encore augmenter les scores. Nous pourrions mesurer l'impact des différents hyper-paramètres comme la température. Au vu de l'instabilité des modèles, nous pourrions lancer plusieurs runs par configuration pour avoir une estimation plus fiable des performances. Enfin, pour généraliser les résultats obtenus dans l'article, il serait souhaitable de tester ces approche sur des benchmarks plus larges d'avis clients.

Remerciements

Nous remercions chaleureusement toutes les personnes qui sont intervenues de près ou de loin sur ce projet : Sofiane Kerroua, Mathilde Poulain, Mathilde Jeuland, Aurore Hamimi, Oualid Akhsass, Marwen Touzi, Laura Rouhier, Sonia Audheon, Dominique Manzoni-Quantin, François Raynaud.

Références

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*, 2023.

[2] Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. Leak, cheat, repeat : Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv :2402.03927*, 2024.

[3] Youngjin Chae and Thomas Davidson. Large language models for text classification : From zero-shot learning to fine-tuning. *Open Science Foundation*, 2023.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 4171–4186, 2019.

[5] Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, and Ge Li. Generalization or memorization : Data contamination and trustworthy evaluation for large language models. *arXiv preprint arXiv :2402.15938*, 2024.

[6] Guillaume Dubuisson Duplessis, Elliot Bartholme, Sofiane Kerroua, Mathilde Poulain, Ahès Roulier, and Anne-Laure Guénet. Désidentification de données texte produites dans un cadre de relation client. In *Actes de la 27eme conférence Traitement Automatique des Langues Naturelles (TALN) – démonstrations*, pages 10–13, 2020.

[7] Guillaume Dubuisson Duplessis, François Bullier, and Anne-Laure Guénet. Démonstration : exploration sémantique de données texte de la relation client. In *9ème Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle APIA@ PFIA2023*, pages 103–106, 2023.

[8] Guillaume Dubuisson Duplessis, Sofiane Kerroua, Ludivine Kuznik, and Anne-Laure Guénet. Cameli@ : analyses automatiques d'e-mails pour améliorer la relation client. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume IV : Démonstrations*, pages 623–626, 2019.

[9] Guillaume Dubuisson Duplessis, Manon Richard, and Anne-Laure Guénet. Segmentation de phases de dialogue dans des retranscriptions de conversations de centres d'appels. In *9ème Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle APIA@ PFIA2023*, 2023.

[10] Marah Abdin et al. Phi-3 technical report : A highly capable language model locally on your phone, 2024.

[11] John Fields, Kevin Chovanec, and Praveen Madiraju. A survey of text classification with transformers : How wide ? how large ? how long ? how accurate ? how expensive ? how safe ? *IEEE Access*, 12 :6518–6531, 2024.

[12] Matthew Freestone and Shubhra Kanti Karmaker Santu. Word embeddings revisited : Do llms offer something new ? *arXiv preprint arXiv :2402.11094*, 2024.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora : Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

[14] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv :2310.06825*, 2023.

[15] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas,

Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv :2401.04088*, 2024.

[16] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35 :22199–22213, 2022.

[17] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert : Unsupervised language model pre-training for french. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, 2020.

[18] Changmao Li and Jeffrey Flanigan. Task contamination : Language models may not be few-shot anymore, 2023.

[19] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict : A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9) :1–35, 2023.

[20] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[24] Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models. *arXiv preprint arXiv :2305.08377*, 2023.

[25] Matei A. Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41 :39–45, 2018.

[26] Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science ? *Computational Linguistics*, pages 1–53, 2023.