Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

Evaluation de petits modèles de langues (SLM) sur un corpus de Sciences Humaines et Sociales (SHS) en français

Sam Vallet, Philippe Suignard
EDF R&D
7, boulevard Gaspard Monge 91120 PALAISEAU
prenom.nom@edf.fr

RÉSUMÉ

Cet article évalue une série de plusieurs petits modèles de langues (SLM) sur une tâche de classification de tweets en français. Plusieurs stratégies d'optimisation sont testées : différents prompts (*zero-shot*, *few-shot*), *fine-tuning* avec une couche de classification, présence ou non d'une couche LoRa. Les résultats obtenus avec le modèle Qwen optimisé rivalisent avec un modèle beaucoup plus gros, ce qui valide notre intérêt pour les petits modèles.

ABSTRACT

Evaluation of small language models (SLM) on a corpus of Human and Social Sciences (HSS) in French

This paper evaluates a series of several small language models (SLM) on a French tweet classification task. Several optimization strategies are tested : different prompts (zero-shot, few-shot), fine-tuning with a classification layer, presence or absence of a LoRa layer. The results obtained with the optimized Qwen model compete with a much larger model, which validates our interest in small models.

MOTS-CLÉS : LLM, SLM, IA frugale, modèle de langue, évaluation.

KEYWORDS: LLM, SLM, frugal AI, language model, evaluation.

1 Introduction

Il n'est plus besoin de présenter les LLM (« Large Language Models ») des GAFAM (Google, Microsoft, Facebook/Meta, etc.), OpenAI (ChatGPT) ou Mistral, etc. Ces LLM ont vu leur taille et performance fortement évoluer à la hausse ces dernières années. L'adoption de ces outils en entreprise est également très rapide. Dans beaucoup d'entreprises, des « Task force » ont vu le jour sur le sujet des IA génératives, en général pour aboutir au lancement de plusieurs tests, démonstrateurs ou POC. Mais souvent, les mêmes conclusions reviennent :

— Bien que les résultats de ces expérimentations soient positifs, les IA génératives nécessitent des infrastructures très importantes (en apprentissage mais également en inférence) ce qui engendre des coûts élevés ;
— Les grands modèles sont très performants mais il est difficile de connaître les données sur lesquelles ils ont été entrainés, ce qui peut également constituer un frein à leur déploiement.

Les grands LLM sont donc des outils généralistes très performants qu'il est possible d'utiliser de manière générique (en les promptant comme ChatGPT par exemple) ou en les spécialisant (affinant)

Actes de l'atelier Évaluation des modèles génératifs (LLM) et challenge 2025 (EvalLLM), pages 178–187.

sur des tâches spécifiques (« *fine-tuning* »). Mais le ratio coût/performance des approches LLM généralistes n'est pas optimal. Comme le montre (Grangier *et al.*, 2024), il y a un équilibre à trouver entre la taille du modèle utilisé (et indirectement son coût de fonctionnement) et ses performances.

Si l'utilisation des LLM est bien adaptée pour des usages généralistes, leur utilisation dans des tâches métiers très spécifiques/spécialisées et répétitives pose des questions. Pour une tâche métier de classification, les LLM ne font pas toujours mieux que des approches plus frugales, (Vautier *et al.*, 2024) ont montré qu'une approche à base de LLM pouvait être moins performante qu'une approche de type CamemBERT hyper optimisée (*fine-tunée*, c'est à dire affinée sur du vocabulaire métier).

Dans ces conditions, nous trouvons intéressant d'étudier les petits modèles (« Small Language Models » ou SLM) en faisant le pari qu'ils peuvent être mieux adaptés pour le monde de l'entreprise, moins énergivores et plus facilement déployables. Certains domaines comme celui de la santé commencent à déployer de tels modèles (Garg *et al.*, 2025). Ces modèles sont de plus en plus l'objet d'attention comme en témoignent les états de l'art faits dans (Lu *et al.*, 2024) et (Wang *et al.*, 2024), un petit modèle désignant ici un modèle d'environ un milliard de paramètres ou moins. La section 2 dresse la liste des SLM testés.

Le corpus utilisé pour nos tests est un corpus de tweets en français sur la thématique de l'énergie, annoté selon deux catégories : dénonciation ou discernement. Il est décrit dans la section 3.

La section 4 présente les différentes stratégies utilisées pour « optimiser » le modèle : prompt naïf qui explique les deux catégories, *few-shot* en indiquant des exemples dans le prompt ou *fine-tuning* du modèle.

La section 5 présente et commente les résultats obtenus. Elle est suivie par une section de conclusion qui présente des réflexions pour des travaux futurs.

2 Liste des SLM testés

La notion de petits modèles n'étant pas encore stabilisée dans la communauté scientifique, nous nous focalisons sur des modèles dont la taille est de l'ordre du milliard de paramètres et nous nous concentrons donc sur l'exploration et l'évaluation de modèles de langue de taille comprise entre entre 135M et 1,5B de paramètres. Certains modèles sont proposés en version *Instruct*, c'est-à-dire pré-entraînés ou adaptés pour répondre à des instructions formulées en langage naturel.

Tous les modèles listés ci-dessous sont disponibles en open source sur la plateforme Hugging Face :

Modèle	**Paramètres**	**Instruct**	**Langue(s)**
Falcon-RW	1B		anglais
Qwen2.5-Instruct	0,5B – 1,5B	oui	anglais, français, chinois...
SmolLM	135M – 360M	oui	anglais
OLMo	1B		anglais
CroissantLLM-Base	1B		français, anglais
TinyLlama-1.1B-Chat-v1.0	1,1B	oui	anglais
Pleias-preview	350M – 1,2B		français, anglais

TABLE 1 – Comparaison de modèles compacts (instruct, langues, taille)

3 Corpus utilisé pour l'évaluation

Le corpus utilisé pour les tests est un corpus d'environ **3000 tweets**, en français, sur le thème de l'énergie (nucléaire, éolien, solaire, etc.) (Brugidou & Suignard, 2024). Ce corpus a été constitué dans le cadre d'une analyse sociologique sur la dénonciation (Boltanski *et al.*, 1984). Dans cet article, Luc Boltanski montrait que les discours de dénonciation médiatique s'organisaient selon un schéma actanciel formé de quatre actants : un dénonciateur, une victime, un persécuteur et un juge. L'annotation du corpus reprend l'idée de l'article, mais restreint le cadre d'analyse à deux catégories seulement : **dénonciation** ou **discernement**. La catégorie discernement concerne des tweets pédagogiques qui expliquent ou présentent factuellement une position, une mesure ou une technologie, tandis que la catégorie dénonciation concerne des tweets plus polémiques, qui critiquent ou dénoncent soit une personne, une association ou un parti dont l'action est jugée injuste, soit une mesure ou une technologie dont les conséquences sont jugées néfastes. Cette distinction en deux catégories correspond aux stratégies observées sur Twitter à savoir des tweets explicatifs qui visent à démontrer, détailler des opinions (Foderaro & Lorentzen, 2023) et des tweets de mobilisation qui visent à dénigrer des personnes, associations ou technologies (ici le nucléaire ou l'éolien) (Mercier, 2015).

Le corpus est extrait d'un corpus beaucoup plus vaste (plusieurs millions de tweets) mais se veut le plus représentatif possible. Il présente les caractéristiques suivantes :

- être équilibré entre les positions pro/anti nucléaire/éolien/solaire ;
- émis par des influenceurs (personnes dont la voix compte) mais aussi par des personnes « lambda » entre janvier 2021 et mai 2023 ;
- les tweets sont émis pendant des periodes de forte politisation (élection régionale de juin 2021 et élection présidentielle d'avril 2022) mais également en dehors de ces périodes ;
- il contient autant de tweets dénonciation que de tweets discernement.

Voici quelques exemples de tweets de ce corpus :

- **Dénonciation** : « Le nucléaire n'est pas seulement dangereux, il est également extrêmement coûteux. Penser énergies renouvelables ne relève pas de l'idéologie mais du pragmatisme » ;
- **Discernement** : « La France veut se réindustrialiser, reprendre sa place dans le nucléaire, conquérir la chaîne de valeur de l'hydrogène... » ;
- **Dénonciation** : « En vérité, les éoliennes sont une plaie pour la beauté des paysages français. C'est le remède pire que le mal. » ;
- **Discernement** : « Le PR confirme à Belfort la nécessité d'une production électrique nationale et que l'on ne peut se passer d'éolien. » ;

Les tweets ont été annotés par un sociologue expert (4/5$^{\text{ème}}$ du corpus) et par une personne moins experte (1/5$^{\text{ème}}$ du corpus). Ils ont aussi été annotés à l'aide du LLM `NeuralHermes-2.5-Mistral-7B` quantifié en 4bits [1]. Le taux d'accord entre l'annotation automatique faite par le LLM et l'annotation manuelle experte est de l'ordre de **70%**, ce qui constitue une base de référence pour évaluer les résultats obtenus avec les SLM.

1. Il s'agit d'un modèle Mistral 7b fine-tuné, disponible sur le site huggingface.

4 Stratégie d'optimisation des modèles

Nous avons divisé notre jeu de données en trois sous-ensembles : 40% pour le test, 50% pour l'entraînement et 10% pour la validation. Afin de garantir des distributions de classes, dans chaque sous-ensemble, similaires à celle du jeu de données global, nous avons utilisé la fonctionnalité `stratify` de la bibliothèque `sklearn` et avons fixé la seed à une valeur spécifique (`random_state=42`).

Expérimentations sur les stratégies de prompt : Nous avons utilisé plusieurs types de prompts pour évaluer nos modèles. Le premier reprend celui employé lors de l'évaluation initiale du LLM dans la première partie de l'article. Nous avons ensuite testé des prompts avec des exemples. Ces différents formats ont également été utilisés lors des phases de *fine-tuning*. Nous avons exploré des entraînements où la donnée à classer était intégrée dans un prompt plutôt que présentée de manière brute. L'idée derrière cette approche est de fournir un contexte explicite au modèle, afin de guider sa compréhension de la tâche et potentiellement améliorer la qualité de ses prédictions.

Nous avons expérimenté quatre configurations principales :

- Prompt 1 – Explicatif : une consigne décrivant la différence entre discernement et dénonciation, sans exemple.
- Prompt 2 – Deux exemples du jeu de données : la consigne est accompagnée de deux cas annotés provenant du dataset.
- Prompt 3 – Mélange humain/IA : deux exemples réels issus du dataset et deux exemples générés par un LLM, équilibrés entre discernement et dénonciation.
- Prompt 4 – *Few-shot* étendu : vingt exemples du dataset, répartis de manière équilibrée entre les deux classes.

Les prompts complets sont fournis en Annexe A.

***fine-tuning* avec ajout d'une couche de classification :** Nous avons ajouté une couche de classification en sortie de nos modèles. Lors du *fine-tuning*, les poids du modèle pré-entraîné sont gelés, seuls les paramètres de la couche de classification sont mis à jour. Cette approche permet de bénéficier des représentations linguistiques déjà apprises tout en adaptant le modèle à notre tâche spécifique avec un coût computationnel réduit. Avant d'ajouter la couche de classification, nous appliquons une étape de *pooling*. En effet, notre modèle de langue génère une représentation vectorielle pour chaque *token* de la séquence, mais nous devons donner une seule entrée à notre couche de classification. Nos modèles reposent sur l'architecture *Transformer*, grâce au mécanisme d'attention, le dernier *token* est censé capturer des informations sur l'ensemble de la séquence. Nous utilisons donc la représentation de ce dernier *token* comme entrée pour la couche de classification. Une alternative serait de prendre la moyenne des représentations des *tokens* et de la donner au classifieur.

Les hyperparamètres utilisés pour l'entraînement sont les suivants :

- Fonction de perte : `BinaryCrossEntropyLoss`
- Optimiseur : `AdamW`
- Taux d'apprentissage (*learning rate*) : $1e{-5}$
- Taille de batch : 32
- Critère d'*early stopping* : pas d'amélioration de la *loss* de validation après 5 epochs consécutives.

***fine-tuning* avec LoRA :** Pour certains modèles ayant montré de bonnes performances avec la couche de classification, nous avons appliqué un deuxième niveau d'adaptation en utilisant la méthode

Low-Rank Adaptation (LoRA) (Hu *et al.*, 2022). LoRA consiste à insérer des matrices de faible rang parallèles aux poids existants de certaines couches du modèle. Pendant l'entraînement, seuls ces nouveaux paramètres sont mis à jour, tandis que les poids d'origine restent gelés. Cette approche permet de réduire significativement le nombre total de paramètres à entraîner. Dans notre cas nous maintenons la couche de classification et le reste des poids du modele gelée, et nous insérons des matrices LoRA sur certaines couches du modèle.

Les paramètres utilisés sont :

- Rank LoRA : 32
- Alpha : 16
- Dropout : 0,05
- Optimiseur : `AdamW`, même learning rate que précédemment
- Couche : `q_proj (query projection)`, `v_proj (value projection)`

5 Résultats obtenus

Dans cette section nous présenterons les resultats obtenus avec les SLM sur la tâche de classification, dans un premier temps sans *fine-tuning*, puis avec l'ajout d'une couche de classification et finalement avec l'ajout d'un LoRa.

Classification sans *fine-tuning* : Pour cette première évaluation, les modèles sont utilisés en génération libre, comme dans un usage classique. Afin de limiter la sortie à une réponse concise, nous imposons une contrainte de génération avec un maximum de 10 nouveaux *tokens*. L'évaluation repose ensuite sur le premier mot généré par le modèle : si ce mot correspond à l'une des deux classes attendues (« dénonciation » ou « discernement »), il est considéré comme la prédiction du modèle sinon la réponse est considérée comme incorrecte.

Modèle/Prompt	Prompt 1	Prompt 2	Prompt 3	Prompt 4
Falcon-rw-1B	0,00	0,0376	0,1665	0,205
Qwen2.5-0.5B-Instruct	**0,392**	**0,421**	0,333	0,34
Qwen2.5-1.5B-Instruct	0,254	0,406	**0,62**	0,35
SmolLM2-135M-Instruct	0,00	0,011	0,329	0,148
SmolLM-360M-Instruct	0,00	0,014	0,19	0,336
OLMo-1B-hf	0,00	0,116	0,333	0,333
CroissantLLMBase	0,00	0,08	0,222	0,33
TinyLlama-1.1B-chat-v1	0,00	0,17	0,335	**0,41**
Pleias-350m-Preview	0,00	0,166	0,333	0,276
Pleias-1.2B-Preview	0,00	0,083	0,333	0,34

TABLE 2 – Scores F1 sur le jeu de test obtenus par les différents modèles sur la tâche de classification en fonction du prompt.

Sans exemple dans les prompts, la plupart des modèles échouent à comprendre la consigne, à l'exception des modèles Qwen. L'ajout d'exemples améliore généralement les résultats. À ce stade, les modèles Qwen apparaissent comme les plus prometteurs. On note également que la taille du modèle ne garantit pas nécessairement de meilleures performances. Pour l'instant, à part pour Qwen2.5

1,5B, les résultats sont moins bon que le hasard. Cela est principalement dû au fait que les modèles prédisent souvent autre chose que dénonciation ou discernement.

***fine-tuning* sur la couche de classification :** Nous avons ensuite évalué les performances des modèles après un *fine-tuning* appliqué uniquement sur la couche de classification. Pour chaque modèle, plusieurs variantes ont été testées :

— la version **Base**, dans laquelle les données d'entraînement sont utilisées telles quelles, sans mise en forme particulière ;
— les versions **prompt 1**, **prompt 2** et **prompt 3**, où on fournit les données pour l'entraînement à travers les prompts.

Modèle\Prompt	Base	Prompt 1	Prompt 2	Prompt 3
Qwen2.5-0.5B-Instruct-prompt 1	0,59	0,33	**0,58**	**0,64**
Qwen2.5-1.5B-Instruct-Base	**0,62**	0,33	0,40	0,34
SmolLM-360M-Instruct-prompt 2	0,42	**0,53**	0,41	0,39

TABLE 3 – Scores F1 sur le jeu de test obtenus par les modèles *fine-tunés* avec la couche de classification, sur la tâche de classification en fonction du prompt. Cette table présente les modèles ayant obtenu les meilleures performances, les résultats complets sont disponibles en Annexe B.

L'ajout de la couche de classification améliore les performances des modèles, bien que certains obtiennent encore des résultats inférieurs à un modèle aléatoire. Les modèles Qwen sont généralement les plus performants. Ceux entraînés avec les prompts 2 et 3, qui contiennent des exemples, ont tendance à avoir des résultats moins bons, sauf pour Pleias 1,2B. Cela pourrait être dû à une difficulté à généraliser, probablement en raison de l'influence trop forte des exemples fournis lors de l'entrainement. Une autre observation est que de nombreux modèles obtiennent des scores autour de 0,33. Dans ces cas, le modèle prédit presque uniquement une seule classe.

***fine-tuning* avec LoRA :** Nous avons sélectionné les modèles ayant obtenu de bons résultats avec le classifieur. Ensuite, nous avons ajouté LoRA à ces modèles. Plus précisément, LoRA a été appliqué aux modules `'q_proj'` et `'v_proj'`, qui font partie du mécanisme d'attention des modèles Transformer. Le module `'q_proj'` projette les requêtes (queries), qui permettent au modèle de déterminer quelles parties de l'entrée sont importantes, tandis que `'v_proj'` projette les valeurs (values), qui sont utilisées pour la prise de décision. L'ajout de LoRA sur ces modules permet d'affiner l'attention du modèle sans alourdir son architecture. Nous avons également testé l'ajout de LoRa sur d'autres modules mais les meilleures performances sont atteintes avec `'q_proj'` et `'v_proj'`.

Les noms des modèles dans le tableau ci dessous suivent la structure suivante : **Qwen2.5-0.5B-Instruct-prompt 1 Lora32 Base :**

— **Qwen2.5-0.5B-Instruct** fait référence au modèle de base Qwen.
— prompt 1 : indique sur quel type de données est entrainé le classifieur.
— Lora32 : fait référence à l'application de LoRA d'attention avec un rang de 32.
— Base : indique sur quel type de données on a entrainé le LoRa

Avec l'ajout du LoRa on finit par obtenir de bons resultats sur notre jeu de données de test avec des résultats supérieurs a 0,7 pour le F1 score. On obtient cependant des scores similaires avec un TF-IDF suivi d'une régression logistique.

Modèle\Prompt	Base	prompt 1	prompt 2	prompt 3
Qwen2.5-0.5B-Instruct-prompt 1 Lora 32 - Base	0,63	0,33	**0,736**	0,748
Qwen2.5-1.5B-Instruct-Base Lora 32 - Base	0,738	0,33	0,35	0,35
Qwen3-1.7B-Base Lora 32 - Base	0,76	0,35	0,68	**0,75**
OLMo-1B-hf-prompt 2 Lora 32 - Base	**0,78**	0,34	0,35	0,33
Pleias-350m-Preview-prompt 1 Lora 32 - Base	0,54	**0,48**	0,57	0,61
TF-IDF Regression logistique	0,73			

TABLE 4 – Scores F1 sur le jeu de test obtenus par les modèles *fine-tunés* avec la couche de classification puis par un LoRa, sur la tâche de classification en fonction du prompt. Les performances sont comparées à un modèle utilisant une régression logistique entraînée sur des représentations TF-IDF.

6 Conclusion et travaux futurs

Dans cet article, nous avons testé le comportement de plusieurs petits modèles de langues (SLM) sur une tâche de classification de tweets en français. En ajoutant une couche de classification au modèle et en ajoutant une couche LoRa, nous obtenons un score supérieur, avec un modèle 1,5B (Qwen), à celui obtenu avec un modèle 7B, plus gros.

L'approche semble donc prometteuse et plusieurs axes d'amélioration sont identifiés :

- Tester d'autres SLM comme celui que va prochainement sortir la société Linagora (Lucie-1B) ;
- Distillation : une autre approche à tester consiste à partir d'un grand modèle afin de réaliser une distillation des connaissances vers un modèle plus petit. Les grands modèles ont tendance à mieux apprendre lors du *fine-tuning*. L'idée serait donc de fine-tuner un LLM sur notre problématique, puis de distiller les connaissances vers un modèle plus léger ;
- Tester d'autres approches ou méthodes de fine-tunning ;
- Les tokeniseurs utilisés par les modèles sont généralement entrainés sur des corpus anglophones, il pourrait être intéressant de creuser cet aspect pour produire des tokeniseurs plus adaptés au français et à des corpus métier spécifiques ;
- Tester l'utilisation de SLM sur d'autres tâches métier.

Remerciements

Nous remercions Mathieu Brugidou pour sa contribution à l'annotation des tweets du corpus dénonciation/discernement.

Références

BOLTANSKI L., DARRÉ Y. & SCHILTZ M.-A. (1984). La dénonciation. *Actes de la recherche en sciences sociales*, **51**(1), 3–40.

BRUGIDOU M. & SUIGNARD P. (2024). Des communautés numériques et des énoncés dans tous leurs états. In *Les scènes de la dénonciation publique Médias, langages et sociétés. Médias, langages et sociétés*.

FODERARO A. & LORENTZEN D. G. (2023). Argumentative practices and patterns in debating climate change on twitter. *Aslib Journal of Information Management*, **75**(1), 131–148.

GARG M., RAZA S., RAYANA S., LIU X. & SOHN S. (2025). The rise of small language models in healthcare : A comprehensive survey. *arXiv preprint arXiv :2504.17119*. arXiv : 2504.17119.

GRANGIER D., KATHAROPOULOS A., ABLIN P. & HANNUN A. (2024). Specialized language models with cheap inference from limited domain data.

HU E. J., SHEN Y., WALLIS P., ALLEN-ZHU Z., LI Y., WANG S., WANG L., CHEN W. *et al.* (2022). Lora : Low-rank adaptation of large language models. *ICLR*, **1**(2), 3.

LU Z., LI X., CAI D., YI R., LIU F., ZHANG X., LANE N. D. & XU M. (2024). Small language models : Survey, measurements, and insights. *arXiv preprint arXiv :2409.15790*.

MERCIER A. (2015). Twitter, espace politique, espace polémique : L'exemple des tweet-campagnes municipales en france (janvier-mars 2014).

VAUTIER N., HÉRY M., MILED M., TRUCHE I., BULLIER F., GUÉNET A.-L., DUPLESSIS G. D., CAMPANO S. & SUIGNARD P. (2024). Utilisation de llms pour la classification d'avis client et comparaison avec une approche classique basée sur camembert. In *APIA 2024*.

WANG F., ZHANG Z., ZHANG X., WU Z., MO T., LU Q., WANG W., LI R., XU J., TANG X. *et al.* (2024). A comprehensive survey of small language models in the era of large language models : Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *arXiv preprint arXiv :2411.03350*. arXiv : 2411.03350.

Annexe A : Détails des prompts utilisés

— Prompt 1 :

```
Tu dois classer un tweet selon deux catégories : dénonciation
ou discernement. La catégorie discernement concerne
des textes pédagogiques qui expliquent ou présentent
factuellement une position, une mesure ou une technologie.
La catégorie dénonciation concerne des textes plus
polémiques, qui critiquent ou dénoncent soit une personne,
une association ou un parti dont l'action est jugée injuste,
soit une mesure ou une technologie dont les conséquences sont
jugées néfastes. Voici le tweet à classer : « Patrick Cohen
démontre, s'il en était besoin, que les lubies écolos sur
le solaire et l'éolien se heurtent au principe de réalité
de l'hiver et que le nucléaire est une chance que la France
doit préserver. ». Quelle est la catégorie de ce tweet :
```

dénonciation ou discernement ? Un seul choix est autorisé. Réponds avec un seul mot.

— Prompt 2 :

Voici quelques exemples de tweets classés dans deux catégories : dénonciation et discernement. Je veux que tu me donnes uniquement la catégorie du tweet suivant, dénonciation ou discernement : 1. Tweet : PARTIS PRIS ... Catégorie : Dénonciation 2. Tweet : En 1938... Catégorie : Discernement 3 Tweet : Patrick Cohen... catégorie :

— Prompt 3 :

Voici quelques exemples de tweets classés dans deux catégories : dénonciation et discernement. Je veux que tu me donnes uniquement la catégorie du tweet suivant, dénonciation ou discernement : 1. Tweet : PARTIS PRIS ... Catégorie : Dénonciation 2. Tweet : En 1938... Catégorie : Discernement 3. Tweet : Les technologies d'énergie solaire... Catégorie : Discernement 4. Tweet : Pourquoi les politiques... Catégorie : Dénonciation 5. Tweet : Patrick Cohen... Catégorie :

Annexe B : Résultats exhaustifs

Modèle/Prompt	Base	Prompt 1	Prompt 2	Prompt 3
Qwen2.5-0.5B-Instruct-Base	0,51	0,33	0,33	0,33
Qwen2.5-0.5B-Instruct-prompt 1	0,59	0,33	**0,58**	**0,64**
Qwen2.5-0.5B-Instruct-prompt 2	0,40	0,33	0,33	0,33
Qwen2.5-0.5B-Instruct-prompt 3	0,34	0,33	0,33	0,33
Qwen2.5-0.5B-Instruct-prompt mix	0,57	0,38	0,33	0,33
Qwen2.5-1.5B-Instruct-Base	**0,62**	0,33	0,40	0,34
Qwen2.5-1.5B-Instruct-prompt 1	0,51	0,33	0,33	0,33
Qwen2.5-1.5B-Instruct-prompt 2	0,46	0,33	0,33	0,33
Qwen2.5-1.5B-Instruct-prompt 3	0.33	0,33	0,33	0,33
SmolLM2-135M-Instruct-Base	0,34	0,33	0,33	0,33
SmolLM2-135M-Instruct-prompt 1	0,47	0,35	0,33	0,33
SmolLM2-135M-Instruct-prompt 2	0,38	0,33	0,33	0,33
SmolLM2-135M-Instruct-prompt 3	0,46	0,33	0,41	0,49
SmolLM-360M-Instruct-Base	0,37	0,33	0,33	0,33
SmolLM-360M-Instruct-prompt 1	0,43	0,33	0,33	0,33
SmolLM-360M-Instruct-prompt 2	0,42	**0,53**	0,41	0,39
SmolLM-360M-Instruct-prompt 3	0,36	0,33	0,33	0,33
OLMo-1B-hf-Base	0,38	0,33	0,38	0,33
OLMo-1B-hf-prompt 1	0,44	0,33	0,33	0,33
OLMo-1B-hf-prompt 2	0,52	0,33	0,33	0,33
OLMo-1B-hf-prompt 3	0,38	0,33	0,33	0,33
TinyLlama-1.1B-chat-v1-Base	0,48	0,33	0,33	0,33
TinyLlama-1.1B-chat-v1-prompt 1	0,39	0,33	0,34	0,33
TinyLlama-1.1B-chat-v1-prompt 2	0,45	0,33	0,33	0,33
TinyLlama-1.1B-chat-v1-prompt 3	0,42	0,33	0,33	0,33
Pleias-350m-Preview-Base	0,39	0,33	0,35	0,33
Pleias-350m-Preview-prompt 1	0,47	0,33	0,34	0,33
Pleias-350m-Preview-prompt 2	0,35	0,33	0,33	0,33
Pleias-350m-Preview-prompt 3	0,42	0,33	0,33	0,33
Pleias-1.2B-Preview-Base	0,43	0,33	0,41	0,48
Pleias-1.2B-Preview-prompt 1	0,48	0,33	0,33	0,33
Pleias-1.2B-Preview-prompt 2	0,48	0,42	0,33	0,33
Pleias-1.2B-Preview-prompt 3	Nan	Nan	Nan	Nan

TABLE 5 – Scores F1 obtenus par les modèles *fine-tunés* avec la couche de classification, sur la tâche de classification en fonction du prompt.