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

Évaluation de la Robustesse des LLM : Proposition d'un Cadre Méthodologique et Développement d'un Benchmark

Fares Grina, Natalia Kalashnikova
Centre de Recherche et d'Innovation de Talan, 14-20 rue Pergolèse, 75116 Paris, France
fares.grina@talan.com, natalia.kalashnikova@talan.com

RÉSUMÉ

L'évaluation fiable des grands modèles de langage (LLM) demeure un défi. Nous présentons un framework systématique, basé sur des tests de robustesse et une évaluation hybride. Il génère des variantes de benchmarks pour tester la stabilité des LLM. Les réponses sont évaluées par une double approche automatique (LLM-as-a-judge et une évaluation quantitative). Comme contexte applicatif pour ce type d'évaluation, nous présentons la création et l'annotation d'un benchmark pour l'extraction d'information à partir d'appels d'offres. Un mécanisme de détection compare ensuite les évaluations automatiques ; les désaccords importants déclenchent une expertise humaine ciblée. En agrégeant les scores automatiques cohérents et les jugements humains, notre approche vise à fournir une mesure de performance et de robustesse plus fiable. Ce processus quantifie également le taux de désaccord entre méthodes automatiques, offrant le potentiel pour une comparaison plus transparente et équitable des LLM.

ABSTRACT

Hybrid and Robust LLM Evaluation Framework with Industry Benchmark Creation

Reliable and robust evaluation of Large Language Models (LLMs) remains a challenge. We present a systematic framework based on robustness testing and hybrid evaluation. It generates benchmark variants to test LLM stability. Responses are evaluated via a dual automatic approach (LLM-as-a-judge and quantitative evaluation). As an application context for this type of evaluation, we present the creation and annotation of a benchmark for information extraction from calls for tenders. A detection mechanism then compares the automatic evaluations ; significant disagreements trigger targeted human expertise. By aggregating coherent automatic scores and human judgments, our approach aims to provide a more reliable measure of performance and robustness. This process also quantifies the disagreement rate between automatic methods, offering the potential for more transparent and fairer LLM comparisons.

MOTS-CLÉS : Évaluation de LLM, Robustesse, Création de Benchmark, Annotation de Données, Human-in-the-loop, LLM-as-a-judge, Extraction d'informations.

KEYWORDS: LLM Evaluation, Robustness, Benchmark Creation, Data Annotation, Human-in-the-loop, LLM-as-a-judge, Information Extraction.

ARTICLE : **Accepté à EvalLLM 2025.**

Actes de l'atelier Évaluation des modèles génératifs (LLM) et challenge 2025 (EvalLLM), pages 151–163.

1 Introduction

Les grands modèles de langage (LLM) ont connu une progression fulgurante, démontrant des capacités impressionnantes dans une multitude de tâches liées au traitement du langage naturel (Brown *et al.*, 2020). De GPT (OpenAI, 2023) à Llama (Touvron *et al.*, 2023) en passant par Claude (Bai *et al.*, 2022) et Gemini (Team *et al.*, 2023), ces modèles sont de plus en plus intégrés dans des applications critiques et des processus décisionnels. Cependant, cette adoption rapide soulève une question cruciale : comment évaluer de manière fiable et pertinente leurs performances réelles ? Les méthodes d'évaluation actuelles reposent souvent sur des benchmarks statiques (Wang *et al.*, 2019) qui ne capturent qu'une fraction des défis rencontrés en pratique. Elles peinent notamment à évaluer la robustesse des modèles face à de légères variations dans les instructions (prompts) ou à la présence de bruit (Ribeiro *et al.*, 2016), caractéristiques pourtant fréquentes dans les interactions réelles. De plus, l'émergence de l'évaluation par LLM (*LLM-as-a-judge*) (Zheng *et al.*, 2023) offre une alternative prometteuse mais soulève des questions de biais (Li *et al.*, 2025) et de fiabilité (Wataoka *et al.*, 2024), tandis que les métriques automatiques traditionnelles (BLEU (Papineni *et al.*, 2002), ROUGE (Lin, 2004), etc.) ne corrèlent pas toujours parfaitement avec le jugement humain (Zhang *et al.*, 2019; Wu *et al.*, 2021).

Pour répondre à ces limitations, nous proposons dans cet article un framework d'évaluation systématique des LLM, conçu pour intégrer nativement les tests de robustesse et une validation croisée des méthodes d'évaluation. Notre approche, illustrée schématiquement en Figure 1, repose sur trois piliers : (1) la génération de multiples variantes d'un benchmark de référence pour simuler des scénarios réalistes et tester la robustesse ; (2) une double évaluation automatique combinant la flexibilité d'un LLM-juge et la rigueur de métriques quantitatives adaptées à la tâche ; (3) une boucle d'intervention humaine experte, déclenchée uniquement en cas de conflit entre les évaluateurs automatiques, pour garantir la fiabilité des résultats finaux.

En plus de présenter ce framework méthodologique général, nous détaillons la création d'un benchmark spécifique dans le cadre d'une étude de cas industrielle portant sur l'extraction d'informations clés à partir d'appels d'offres (Section 4). Cette contribution vise à fournir un exemple concret de mise en place d'un benchmark pertinent pour ce type d'évaluation robuste.

Cet article est structuré comme suit : la Section 2 discute des travaux connexes. La Section 3 détaille les composants de notre framework d'évaluation. La Section 4 présente l'étude de cas et la création du benchmark sur les appels d'offres.

2 État de l'art

L'évaluation des grands modèles de langage (LLM) est un domaine de recherche en pleine effervescence, confronté à des défis uniques liés à la nature générative et aux capacités étendues de ces modèles (Chang *et al.*, 2023).

Défis de l'évaluation et Benchmarks Standards L'évaluation des LLM a progressé depuis les benchmarks focalisés sur des tâches de traitement de langage spécifiques (ex : GLUE (Wang *et al.*, 2018), SuperGLUE (Wang *et al.*, 2019)) vers des évaluations plus massives et multi-tâches (ex : MMLU (Hendrycks *et al.*, 2021)). Des efforts considérables ont été déployés pour créer des suites d'évaluation holistiques, comme HELM (Liang *et al.*, 2022), qui visent à couvrir un large spectre de

scénarios et de métriques pour une évaluation plus complète.

Cependant, ces benchmarks standards présentent plusieurs limitations. Leur nature souvent statique ne reflète pas la dynamique des interactions réelles et rend difficile l'évaluation de la robustesse des modèles aux variations d'entrée (Dodge *et al.*, 2021). De plus, le risque de contamination des données d'entraînement des LLM par les données des benchmarks d'évaluation est une préoccupation croissante, pouvant biaiser les résultats (Deng *et al.*, 2024). Enfin, se concentrer sur un score agrégé unique peut masquer des faiblesses spécifiques ou des comportements indésirables que des évaluations plus ciblées ou comportementales cherchent à révéler (Ribeiro *et al.*, 2020).

Évaluation de la Robustesse en TAL La robustesse des modèles de TAL face à des entrées bruitées ou légèrement modifiées est essentielle pour leur déploiement fiable en conditions réelles (Siska *et al.*, 2024). Plusieurs approches ont été explorées pour évaluer et améliorer cette robustesse. Les attaques adverses visent à générer des perturbations minimales (souvent imperceptibles pour un humain) conçues spécifiquement pour tromper le modèle (Jin *et al.*, 2020; Alzantot *et al.*, 2018). Bien qu'utiles pour sonder les vulnérabilités, ces perturbations ne sont pas toujours représentatives des variations naturelles rencontrées en pratique. D'autres travaux se concentrent sur des perturbations plus naturelles, simulant des erreurs de frappe (Belinkov & Bisk, 2017), des reformulations ou paraphrases (Kang *et al.*, 2024), ou l'ajout de bruit lexical. Des techniques comme l'augmentation des données (Wei & Zou, 2019) ou la traduction aller-retour (back-translation) (Rajaei *et al.*, 2024) sont souvent utilisées pour générer ces variations. Des librairies comme `nlpaug` (Ma, 2019) facilitent l'application de ces différentes stratégies de bruitage. L'évaluation sur des données traduites (Wang *et al.*, 2024), que nous incluons également, permet d'aborder la dimension cross-lingue de la robustesse.

Méthodes d'Évaluation Automatique L'évaluation automatique est indispensable pour traiter de grands volumes de réponses LLM.

Métriques quantitatives. Pour la génération de texte, les métriques historiques comme BLEU (Papineni *et al.*, 2002), ROUGE (Lin, 2004) et METEOR (Banerjee & Lavie, 2005), basées sur le recouvrement de n-grammes, sont encore utilisées mais largement critiquées pour leur faible corrélation avec la qualité sémantique ou le jugement humain (Wu *et al.*, 2021; Reiter, 2018; Kocmi & Federmann, 2021). Des métriques plus récentes exploitant les plongements contextuels, comme BERTScore (Zhang *et al.*, 2019), MoverScore (Zhao *et al.*, 2019), ou des métriques entraînées comme BLEURT (Sellam *et al.*, 2020), montrent de meilleures corrélations mais restent imparfaites et peuvent échouer à capturer la factualité ou la cohérence (Mathur *et al.*, 2020). Pour l'Extraction d'Information (IE), les métriques standards sont la Précision, le Rappel et le F1-score au niveau des entités extraites, complétées par l'Exact Match (EM) pour les valeurs (Manning *et al.*, 2008). Ces métriques sont bien définies mais l'EM peut être trop stricte et le F1 ne capture pas toutes les nuances d'une extraction partiellement correcte.

LLM-as-a-Judge. Face aux limites des métriques précédentes, l'utilisation d'un LLM puissant pour juger les sorties d'un autre LLM (*LLM-as-a-judge*) a gagné en popularité (Zheng *et al.*, 2023). Cette approche offre une flexibilité remarquable, permettant d'évaluer des critères complexes (style, cohérence, créativité, etc.) via des prompts en langage naturel (Gu *et al.*, 2025). Cependant, des études récentes ont mis en évidence plusieurs biais potentiels : biais de position, de verbosité, préférence pour son propre style (*self-preference*) ou pour des modèles similaires (Wataoka *et al.*, 2024), ainsi qu'une sensibilité au prompt d'évaluation utilisé. De plus, l'héritage des biais présents dans les données d'entraînement du LLM-juge reste une préoccupation (Li *et al.*, 2025).

Notre framework reconnaît les forces et faiblesses de ces deux approches automatiques. Il les utilise

donc en parallèle comme deux signaux indépendants, dont la confrontation permet d'identifier les cas nécessitant une vérification plus approfondie.

Évaluation Humaine et Approches Hybrides L'évaluation humaine reste considérée comme la référence ("gold standard") pour juger de la qualité des systèmes de TAL (Belz & Kow, 2008). Cependant, elle est coûteuse en temps et en ressources, difficile à standardiser, et sujette à la variabilité inter-annotateurs, nécessitant des guidelines claires et des mesures d'accord (ex : Kappa, Alpha de Krippendorff) (Artstein & Poesio, 2008).

Pour pallier ces inconvénients, les approches hybrides ou *Human-in-the-Loop* (HITL) cherchent à combiner l'efficacité de l'automatisation avec la précision du jugement humain sur les cas difficiles ou ambigus (Wu *et al.*, 2022; Drori & Te'eni, 2024). Notre framework implémente une boucle HITL spécifiquement pour la validation de l'évaluation : l'intervention humaine n'est pas systématique mais déclenchée par un signal de conflit entre les évaluateurs automatiques, visant ainsi à maximiser la fiabilité de l'évaluation finale tout en maîtrisant l'effort humain.

Création de Benchmarks Spécifiques et Industriels La communauté reconnaît de plus en plus le besoin de benchmarks qui reflètent des tâches et des données du monde réel (Raji *et al.*, 2021). La création de benchmarks pour des domaines industriels spécifiques présente des défis particuliers liés à la confidentialité des données, à la complexité du domaine métier, et au coût de l'annotation experte. Les principes du micro-benchmarking (Babonnaud, 2024) peuvent guider la définition de tâches ciblées et pertinentes. La méthodologie d'annotation doit être particulièrement rigoureuse, incluant la définition collaborative du schéma, l'élaboration de guidelines détaillées, la formation des annotateurs, l'utilisation d'outils adaptés (Kim *et al.*, 2024), et des mécanismes de contrôle qualité. La Section 4 de cet article détaille notre application de ces principes pour créer le benchmark AO, une contribution visant à fournir une ressource pertinente pour l'évaluation de l'IE dans ce contexte spécifique.

En conclusion, nous proposons un framework combinant robustesse, double évaluation automatique et validation humaine, illustré par un benchmark industriel.

3 Framework d'Évaluation Proposé

Nous proposons un framework d'évaluation systématique des modèles de langage (LLM) fondé sur la robustesse, la diversité des scénarios, et une validation croisée des résultats. Comme illustré dans la Figure 1, ce cadre repose sur la génération de multiples versions d'un benchmark de référence (B_{ref}), incluant des variantes perturbées (B_{pert}) et traduites (B_{trad}), afin d'évaluer la stabilité des performances face à des modifications réalistes du prompt.

Plusieurs LLM candidats $m \in M$ (ex : OpenAI, Llama, Claude, Gemini) sont ensuite évalués sur chacune de ces versions $B_i \in B$ selon une double approche automatique : (1) une évaluation par un autre LLM jouant le rôle de juge (*LLM-as-a-judge*), produisant un score S_{juge}, et (2) une évaluation quantitative basée sur des métriques $S_{\text{metriques}}$ (ex : BERTScore, BLEU, ROUGE, F1) adaptées à la tâche et comparant la réponse $R_{i,m}$ à une référence Ref_i.

En cas de cohérence détectée entre S_{juge} et $S_{\text{metriques}}$ (après normalisation et comparaison à un seuil δ), une agrégation des scores est effectuée pour produire une évaluation automatique fiable. En revanche, si un conflit significatif est détecté, une intervention humaine experte est déclenchée pour obtenir une

évaluation S_{humain} qui primera.

Ce processus permet non seulement de quantifier les performances moyennes des modèles, mais aussi de mesurer leur robustesse (via la performance sur les B_{pert} et B_{trad}) et le taux de désaccord automatique, offrant ainsi un cadre plus fiable et transparent pour comparer des LLMs dans des contextes réalistes. Les étapes sont détaillées ci-après.

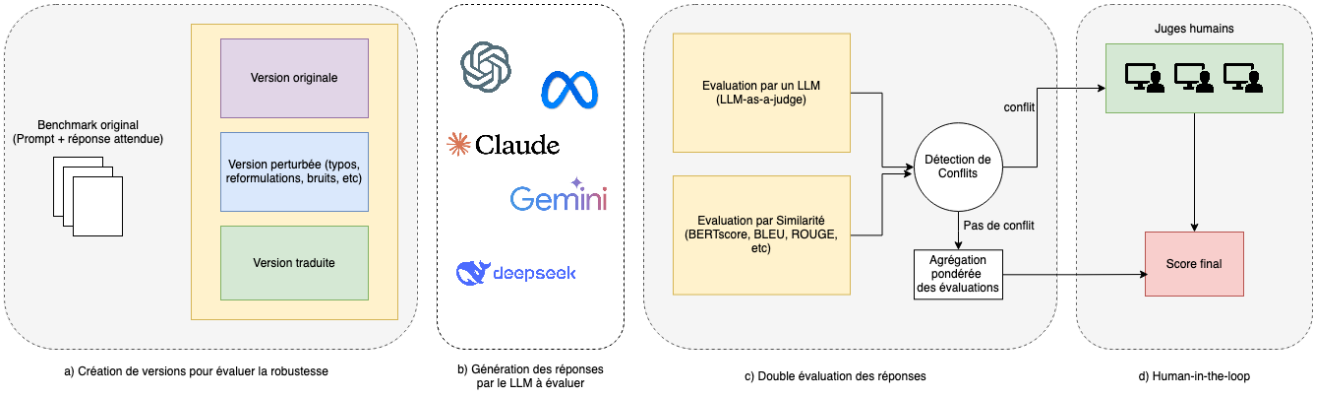

FIGURE 1 – Cadre proposé pour l'évaluation robuste des LLM. Chaque modèle est testé sur plusieurs versions d'un benchmark (originale, perturbée, traduite). Les réponses générées sont évaluées à la fois par un LLM (LLM-as-a-judge) et par des métriques de similarité/tâche. En cas de désaccord, une intervention humaine est déclenchée.

3.1 Préparation du benchmark et génération de variantes

Une limitation majeure des évaluations standard de LLM est leur dépendance à des benchmarks statiques, qui ne reflètent souvent pas la variabilité et le bruit inhérents aux interactions réelles ou aux prompts formulés par divers utilisateurs. Pour évaluer de manière plus approfondie la robustesse et la stabilité des modèles face à des modifications réalistes des entrées, notre framework, tel qu'illustré dans la Figure 1, commence par la génération systématique de multiples variantes d'un benchmark de référence.

Le point de départ est un benchmark de référence soigneusement sélectionné (ou créé), noté B_{ref}. Ce benchmark est choisi pour être représentatif de la ou des tâches cibles pour lesquelles les LLM doivent être évalués. Dans le cadre de notre étude de cas, B_{ref} correspond au benchmark d'extraction d'informations sur les appels d'offres détaillé en Section 4.

À partir de B_{ref}, nous appliquons plusieurs stratégies de transformation pour créer un ensemble diversifié de benchmarks dérivés $B = \{B_{ref}, B_{pert_1}, ..., B_{trad_1}, ...\}$:

— **Variantes Perturbées (B_{pert}) :** Ces versions visent à simuler des imperfections ou des variations courantes dans les prompts réels. Nous distinguons plusieurs types de perturbations :

  — **Bruit Orthographique et Typographique :** Introduction contrôlée d'erreurs de saisie communes (fautes de frappe, inversion/omission/ajout de caractères adjacents, erreurs d'accentuation). Ceci est réalisé, par exemple, en utilisant des opérations au niveau caractère (insertion, suppression, substitution aléatoires) et la simulation d'erreurs de proximité clavier (via des librairies comme `nlpaug` (Ma, 2019)) avec un taux de perturbation cible d'environ 8% au niveau caractère. L'objectif est de tester la tolérance des LLM aux

erreurs de saisie mineures.

— **Reformulations Sémantiques (Paraphrases) :** Génération de variations des prompts originaux qui conservent le sens fondamental mais utilisent une formulation différente. Ces paraphrases sont obtenues via une combinaison de traduction aller-retour (back-translation), en utilisant des langues pivots variées (par exemple FR↔DE, FR↔ES, FR↔ZH pour diversifier les reformulations via des services comme DeepL. Cela permet d'évaluer si le modèle est sensible à la forme de la question plutôt qu'à son contenu sémantique.

— **Variantes Traduites (B_{trad}) :** Pour évaluer la performance cross-lingue du LLM cible (sa capacité à réaliser la tâche dans une autre langue que celle du benchmark original) ou sa robustesse face à des documents potentiellement traduits automatiquement en amont, nous générons des versions du benchmark B_{ref} traduites dans une ou plusieurs langues cibles. Dans cette étude, nous réalisons une traduction directe de l'intégralité du contenu textuel pertinent de chaque document de B_{ref} (originellement en français) vers l'anglais (en tant que langue cible principale). Cette traduction est effectuée en utilisant l'API DeepL, afin d'obtenir une version B_{trad}.

L'application de ces différentes transformations résulte en une collection étendue de benchmarks $B = \{B_{ref}, B_{pert_1}, B_{pert_2}, ..., B_{trad_1}, B_{trad_2}, ...\}$. Chaque $B_i \in B$ (avec $i \neq ref$) représente une variation spécifique de B_{ref}. C'est sur cet ensemble diversifié que les LLM candidats sont ensuite évalués (Section 3.2), permettant une analyse fine non seulement de leur performance moyenne mais surtout de leur stabilité et de leur robustesse face à ces variations contrôlées. Une illustration de ces variations est présentée en Figure 2.

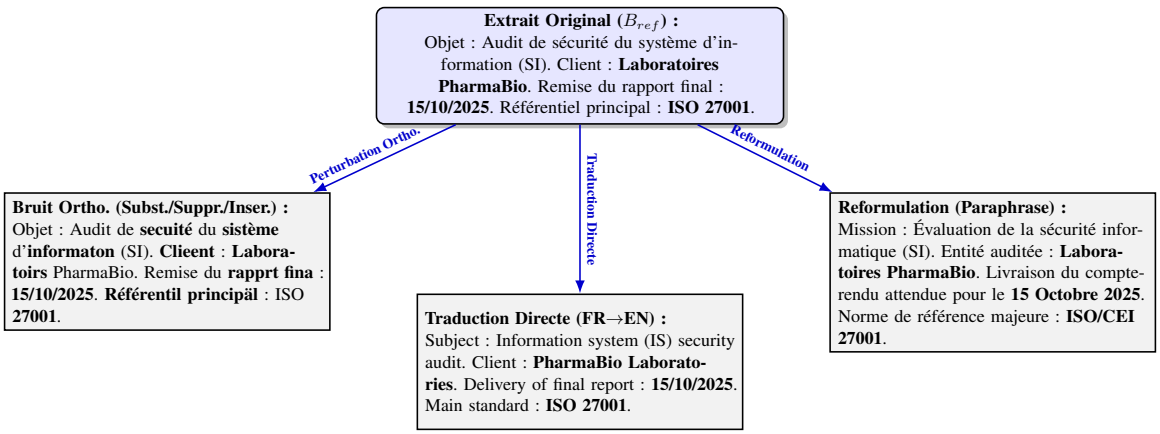

FIGURE 2 – Illustration des types de variations appliquées.

3.2 Génération des réponses par les modèles LLM

Une fois l'ensemble étendu de benchmarks $B = \{B_i\}$ généré, l'étape suivante consiste à soumettre chaque variante B_i au modèle de langage cible m que nous souhaitons évaluer, afin de collecter ses réponses respectives $R_{i,m}$.

L'inférence est réalisée en mode zero-shot, avec un prompt système général et neutre. Afin d'assurer la reproductibilité et de minimiser la variabilité aléatoire lors de cette phase d'évaluation comparative,

nous privilégions des paramètres de génération déterministes ou quasi-déterministes. Typiquement, nous utilisons une température faible (ex : $T = 0.1$).

Ce processus est appliqué systématiquement pour collecter l'ensemble des réponses $R = \{R_{i,m} | B_i \in B\}$, où $R_{i,m}$ est la sortie produite par le modèle cible m pour la variante B_i. Ces réponses collectées R alimentent la phase d'évaluation détaillée dans la Section 3.3.

3.3 Double évaluation automatique et détection des conflits

Après avoir collecté l'ensemble des réponses $R = \{R_{i,m}\}$ pour chaque modèle $m \in M$ et chaque variante de benchmark $B_i \in B$, nous procédons à leur évaluation via une double approche automatique. Cette approche vise à augmenter la fiabilité des scores tout en identifiant les cas où les méthodes automatiques pourraient être insuffisantes ou contradictoires. Pour chaque réponse $R_{i,m}$, deux évaluations indépendantes sont menées en parallèle.

1. Évaluation par LLM-Juge : Premièrement, nous employons une évaluation qualitative assistée par un modèle de langage tiers jouant le rôle de juge (*LLM-as-a-judge*). Nous utilisons pour cette tâche un LLM externe, GPT-4-Turbo, choisi pour ses capacités d'analyse et de raisonnement. Pour chaque réponse $R_{i,m}$, le LLM-juge reçoit le prompt original B_i, la réponse générée $R_{i,m}$, et, si disponible, la réponse de référence attendue Ref_i. À travers un prompt spécifiquement conçu pour cette tâche, le juge est instruit d'évaluer la qualité globale de $R_{i,m}$ (selon des critères comme la pertinence, la correction factuelle, la complétude, la cohérence, etc.) sur une échelle de Likert de 1 (très mauvaise) à 5 (excellente). Cette évaluation produit un score $S_{\text{juge}}(R_{i,m})$.

2. Évaluation par Métriques : Deuxièmement, une évaluation quantitative est réalisée en parallèle à l'aide de métriques automatiques établies, dont le choix dépend de la nature de la tâche évaluée par le benchmark B_{ref}.

- Pour les tâches de génération de texte (ex : résumé, réponse à une question ouverte), nous utiliserions un ensemble de métriques complémentaires comme BERTScore (Zhang *et al.*, 2019), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004).
- Pour les tâches d'extraction d'information (comme dans notre étude de cas Section 4), nous utilisons les métriques standard du domaine : Précision, Rappel et F1-score, ainsi que l'Exact Match (EM) sur les valeurs extraites.

Ces métriques comparent la réponse générée $R_{i,m}$ à la réponse de référence Ref_i et produisent un ensemble de scores $S_{\text{metriques}}(R_{i,m})$. Pour faciliter la comparaison avec le score du LLM-juge, ces scores sont agrégés en un score unique $S_{\text{agg\_metriques}}(R_{i,m})$, par exemple via une moyenne arithmétique simple des F1-scores par champ (macro-average F1) pour notre tâche d'extraction.

3. Détection de Conflit : L'étape suivante consiste à comparer les résultats des deux approches automatiques pour identifier les désaccords significatifs. Pour ce faire, les scores issus des deux méthodes, qui peuvent avoir des échelles natives différentes (par exemple, une échelle de 1 à 5 pour S_{juge} et une échelle souvent entre 0 et 1 pour $S_{\text{agg\_metriques}}$), doivent être ramenés à une échelle commune comparable, typiquement [0, 1]. La technique de normalisation Min-Max est envisagée pour borner les scores dans [0, 1]. Pour un score X avec des bornes théoriques ou observées $X_{\min}$ et $X_{\max}$, le score normalisé X_{norm} est calculé par :

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Par exemple, un score S_{juge} sur une échelle de 1 à 5 serait normalisé en utilisant $X_{\min} = 1$ et $X_{\max} = 5$, donnant $\frac{S_{\text{juge}}-1}{4}$. Pour les métriques $S_{\text{agg\_metriques}}$ qui sont déjà théoriquement comprises entre 0 et 1 (comme un F1-score), cette normalisation peut ne pas être nécessaire ($X_{\min} = 0, X_{\max} = 1$) ou peut être appliquée en utilisant les bornes observées sur l'ensemble du jeu de données si la plage [0, 1] n'est pas entièrement couverte en pratique. Un conflit est donc signalé si la différence absolue entre les scores ainsi normalisés dépasse un seuil de tolérance δ prédéfini :

$$|\text{MinMax}(S_{\text{juge}}(R_{i,m})) - \text{MinMax}(S_{\text{agg\_metriques}}(R_{i,m}))| > \delta \quad (1)$$

où $\text{MinMax}(\cdot)$ désigne l'application de la transformation Min-Max décrite ci-dessus. La valeur de δ est un hyperparamètre du framework. Nous avons fixé $\delta = 0.2$, ce qui correspond à un écart d'un point sur l'échelle Likert 1-5 normalisée.

Transition : Pour chaque réponse évaluée $R_{i,m}$, cette étape produit donc une information cruciale : l'existence ou non d'un conflit entre les évaluateurs automatiques. Les cas jugés *cohérents* (différence $\leq \delta$) sont considérés comme fiables et leurs scores peuvent procéder à l'étape d'agrégation finale. En revanche, les cas marqués comme *conflictuels* (différence $> \delta$) sont mis de côté pour une évaluation humaine experte, garantissant ainsi une supervision ciblée là où l'automatisation montre ses limites.

3.4 Intervention humaine experte (Human-in-the-loop)

Le mécanisme de détection de conflit, détaillé en Section 3.3, identifie les paires réponse-évaluation $(R_{i,m}, \text{scores auto})$ pour lesquelles les signaux issus du LLM-juge (S_{juge}) et des métriques automatiques ($S_{\text{agg\_metriques}}$) divergent de manière significative (Équation 1). Pour ces cas spécifiques, qui représentent une fraction des évaluations totales où la confiance dans le jugement purement automatique est réduite, notre framework intègre une boucle d'intervention humaine (*human-in-the-loop*). L'évaluation des cas conflictuels est confiée à un pool de 3 experts du domaine des appels d'offres, ayant participé à la définition du schéma d'annotation et formés aux critères d'évaluation spécifiques. Pour assurer la robustesse des jugements humains, chaque instance conflictuelle $(B_i, R_{i,m}, Ref_i)$ est évaluée indépendamment par au moins deux de ces experts. L'accord inter-annotateurs (IAA) est systématiquement calculé en utilisant le Kappa de Cohen pondéré (Cohen, 1968) (puisque nous utilisons une échelle ordinale). En cas de désaccord majeur entre les évaluateurs humains (Kappa < 0.6), une procédure de résolution par discussion et consensus est appliquée, potentiellement arbitrée par le troisième expert. Les évaluateurs utilisent une interface web dédiée qui leur présente de manière claire toutes les informations nécessaires : le prompt original B_i, la réponse de référence Ref_i, la réponse générée par le modèle $R_{i,m}$, et, pour contextualiser le conflit, les scores et éventuellement les justifications issus des deux méthodes automatiques (S_{juge} et $S_{\text{metriques}}$). La tâche assignée aux évaluateurs est de fournir un score de qualité global sur la même échelle que le LLM-juge (1-5) en justifiant leur note par un court commentaire identifiant les forces et faiblesses de la réponse $R_{i,m}$ par rapport aux critères d'évaluation. Ce processus aboutit à l'obtention d'un score final validé par l'humain, noté $S_{\text{humain}}(R_{i,m})$, pour chaque réponse $R_{i,m}$ initialement marquée comme conflictuelle. Ce score représente l'évaluation considérée comme la plus fiable pour cet item.

3.5 Agrégation Finale des Scores et Analyse

La dernière étape du framework consiste à agréger les scores obtenus pour chaque réponse afin de produire une évaluation globale pour chaque modèle LLM candidat, ainsi que des mesures de

robustesse et de fiabilité du processus.

Pour une réponse donnée $R_{i,m}$, le score final retenu, noté $S_{final}(R_{i,m})$, est déterminé comme suit :

— Si le cas a été jugé *cohérent* en Section 3.3 (différence $\leq \delta$), $S_{final}(R_{i,m})$ est obtenu par une agrégation des scores automatiques normalisés. Nous utilisons une moyenne simple, donnant un poids égal aux deux méthodes : $S_{final}(R_{i,m}) = 0.5 \cdot \text{MinMax}(S_{\text{juge}}) + 0.5 \cdot \text{MinMax}(S_{\text{agg\_metriques}})$.

— Si le cas a été jugé *conflictuel* (différence $> \delta$), le score issu de l'évaluation humaine est utilisé : $S_{final}(R_{i,m}) = \text{MinMax}(S_{\text{humain}}(R_{i,m}))$, après normalisation MinMax de l'échelle humaine (1-5) vers [0, 1].

À partir de ces scores finaux par réponse, plusieurs indicateurs globaux sont calculés pour chaque modèle $m \in M$:

— **Performance moyenne globale :** $\bar{S}_m = \frac{1}{|B| \times |Items|} \sum_i S_{final}(R_{i,m})$, calculée sur l'ensemble des items du benchmark et de ses variantes B_i.

— **Score de Robustesse :** Mesuré par la baisse de performance relative entre le score moyen sur le benchmark de référence B_{ref} et le score moyen sur l'ensemble des variantes perturbées (B_{pert}) et traduites (B_{trad}). Une faible baisse indique une meilleure robustesse. Alternativement, l'écart-type des scores $S_{final}(R_{i,m})$ d'un modèle m à travers les différentes variantes B_i pour les mêmes items peut être utilisé.

— **Taux de Conflit (par modèle ou global) :** Pourcentage des réponses pour lesquelles l'équation 1 a été vérifiée ($\frac{|\{(i,m)|\text{conflit détecté}\}|}{|B| \times |Items| \times |M|}$), nécessitant une intervention humaine. Cet indicateur renseigne sur la fiabilité intrinsèque de la combinaison LLM-juge / métriques pour un modèle ou une tâche donnée.

Ces indicateurs permettent une comparaison nuancée des LLM, allant au-delà d'un simple classement basé sur la performance moyenne sur B_{ref}.

4 Cas d'usage industriel : Création du Benchmark sur Appels d'Offres

La conception de ce benchmark s'est déroulée en plusieurs étapes structurées, visant à assurer sa pertinence métier et sa validité scientifique, en s'inspirant notamment des principes du micro-benchmarking (Babonnaud, 2024).

Analyse des cas d'usage. Pour ancrer le benchmark dans les besoins réels de l'entreprise, nous avons initié notre démarche par une série d'ateliers participatifs. L'objectif était de recueillir et de prioriser les cas d'usage où les LLM pourraient apporter une valeur ajoutée significative dans les tâches quotidiennes des collaborateurs. Définissant un cas d'usage comme une tâche professionnelle spécifique, nous avons sollicité des employés de divers départements. Via une méthode de brainstorming individuel (post-its virtuels) suivie d'un classement collectif sur une matrice Recurrence vs Complexité (échelles de 0 à 5), nous avons identifié 230 cas d'usage potentiels. Ces cas ont ensuite été classifiés selon leur exploitabilité (Doublon, Expertise requise, Exploitable, Hors périmètre, Inexploitable), permettant d'isoler 65 cas jugés Exploitables.

Sélection de la tâche et du périmètre. Parmi les cas exploitables les plus récurrents et complexes, et tenant compte de la disponibilité des données et des impératifs d'efficacité en contexte industriel, deux

tâches principales ont émergé : le résumé d'AO et l'extraction d'informations clés des AO. Pour ce travail, nous nous sommes concentrés sur la tâche d'extraction d'information, jugée particulièrement critique pour la qualification rapide des opportunités. L'objectif était de concevoir un benchmark représentatif des défis réels de cette tâche.

Sélection des données source. Bien que notre entreprise traite un flux important d'AO, la nature souvent confidentielle de ces documents nous a conduits à constituer notre benchmark à partir de données publiques pour permettre une potentielle réutilisation et comparaison par la communauté scientifique. Nous avons donc collecté un corpus initial d'AO en langue française, disponibles publiquement en ligne, principalement via les plateformes officielles. La sélection finale des documents pour le benchmark a été guidée par deux critères :

— **Longueur :** Conscients de la sensibilité des LLM à la longueur du contexte (Liu & ..., 2023) et pour maîtriser les coûts d'annotation, nous avons ciblé des documents d'une longueur comprise entre 10 et 20 pages, considérés comme suffisamment riches mais traitables.

— **Domaine :** Nous avons privilégié des AO correspondant aux thématiques principales de l'entreprise (énergie, transport, service public, etc).

À l'issue de cette sélection, nous avons retenu un corpus de 60 documents : 10 pour la phase pilote (formation des annotateurs, finalisation des guidelines) et 50 pour la constitution effective du benchmark de référence B_{ref}.

Annotation. Le processus d'annotation a consisté à définir un schéma pertinent en collaboration étroite avec des experts métier de l'entreprise partenaire (gestion de projet, réponse aux AO). Guidés par les besoins d'analyse rapide et de qualification des appels d'offres, nous avons identifié les informations fondamentales à extraire : organisme émetteur, objectifs du projet, critères de sélection, exigences techniques ou administratives clés, et structuration éventuelle du marché en lots. Un outil web dédié a été développé afin de faciliter l'annotation, présentant côte à côte le document source et un formulaire structuré. Chaque document a été annoté indépendamment par deux annotateurs, et les divergences relevées ont été arbitrées par un troisième expert référent, garantissant ainsi la fiabilité et la qualité des annotations finales.

5 Conclusion et Perspectives

Dans cet article, nous avons proposé un framework systématique pour l'évaluation robuste et fiable des grands modèles de langage (LLM). Nous avons argumenté qu'en combinant la génération de variantes de benchmarks pour tester la robustesse, une double évaluation automatique (LLM-juge et métriques spécifiques à la tâche) et une intervention humaine experte ciblée déclenchée par la détection de conflits. Parallèlement, nous avons présenté la création détaillée d'un benchmark industriel pour l'extraction d'information à partir d'appels d'offres, qui constitue un terrain d'application pertinent pour la méthodologie proposée.

Ce travail constitue une proposition méthodologique et une première étape de mise en œuvre (via la création du benchmark). Le développement de l'outil logiciel intégrant l'ensemble des composants de notre framework d'évaluation (génération de variantes, double évaluation, détection de conflit, interface HITL, agrégation) est actuellement en cours.

À plus long terme, au-delà de cette première application, nous envisageons d'étendre l'utilisation de ce framework et de l'outil associé à d'autres tâches identifiées comme prioritaires.

Remerciements

Nous remercions William Babonnaud, à qui appartient l'idée originale du projet et de l'élaboration du cadre théorique initial. Nos remerciements s'adressent également à Irina Deramond et Sarah Taghouti, ainsi qu'à toutes autres personnes engagées dans ce projet, dont l'expertise précieuse a contribué à la pertinence opérationnelle de ces travaux pour les collaborateurs de Talan. Enfin, on remercie Laurent Cervoni, directeur du centre de recherche de Talan, pour son soutien constant et son engagement en faveur de la recherche en milieu industriel.

Références

ALZANTOT M., SHARMA Y., ELGOHARY A., HO B.-J., SRIVASTAVA M. & CHANG K.-W. (2018). Generating natural language adversarial examples. *arXiv preprint arXiv :1804.07998*.

ARTSTEIN R. & POESIO M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*.

BABONNAUD W. (2024). Vers une conceptualisation du micro-benchmarking pour l'évaluation des LLM dans un cadre opérationnel. In *EvalLLM2024 - Atelier sur l'évaluation des modèles génératifs (LLM) et challence d'extraction d'information few-shot*, Toulouse, France : AMIAD, Ministères des Armées. HAL : hal-04678359.

BAI Y., KADAVATH S., KUNDU S., ASKELL A., KERNION J., JONES A., CHEN A., GOLDIE A., MIRHOSEINI A., MCKINNON C. *et al.* (2022). Constitutional ai : Harmlessness from ai feedback. *arXiv preprint arXiv :2212.08073*.

BANERJEE S. & LAVIE A. (2005). Meteor : An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, p. 65–72.

BELINKOV Y. & BISK Y. (2017). Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv :1711.02173*.

BELZ A. & KOW E. (2008). Comparing automatic and human evaluation of nlg systems. *CL*.

BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. *Advances in neural information processing systems*, **33**, 1877–1901.

CHANG Y. *et al.* (2023). A survey on evaluation of large language models.

COHEN J. (1968). Weighted kappa : Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, **70**(4), 213.

DENG C., ZHAO Y., HENG Y., LI Y., CAO J., TANG X. & COHAN A. (2024). Unveiling the spectrum of data contamination in language model : A survey from detection to remediation. In L.-W. KU, A. MARTINS & V. SRIKUMAR, Éds., *Findings of the Association for Computational Linguistics : ACL 2024*, p. 16078–16092, Bangkok, Thailand : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-acl.951.

DODGE J. *et al.* (2021). Measuring true multitask performance...

DRORI I. & TE'ENI D. (2024). Human-in-the-loop ai reviewing : feasibility, opportunities, and risks. *Journal of the Association for Information Systems*, **25**(1), 98–109.

GU J., JIANG X., SHI Z., TAN H., ZHAI X., XU C., LI W., SHEN Y., MA S., LIU H., WANG S., ZHANG K., WANG Y., GAO W., NI L. & GUO J. (2025). A survey on llm-as-a-judge.

HENDRYCKS D. *et al.* (2021). Measuring massive multitask language understanding.

JIN D., JIN Z., ZHOU J. T. & SZOLOVITS P. (2020). Is bert really robust ? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, p. 8018–8025.

KANG M., HWANG S. J., LEE G. & CHO J. (2024). Latent paraphrasing : Perturbation on layers improves knowledge injection in language models. *Advances in Neural Information Processing Systems*, **37**, 119689–119716.

KIM H., MITRA K., CHEN R. L., RAHMAN S. & ZHANG D. (2024). Meganno+ : A human-llm collaborative annotation system. *arXiv preprint arXiv :2402.18050*.

KOCMI T. & FEDERMANN C. (2021). Ship It ! A Large-Scale Log Analysis of METEOR, chrF, and BLEU versus Human Judgments. In *WMT*.

LI M., CHEN H., WANG Y., ZHU T., ZHANG W., ZHU K., WONG K.-F. & WANG J. (2025). Understanding and mitigating the bias inheritance in llm-based data augmentation on downstream tasks.

LIANG P., BOMMASANI R., LEE T., TSIPRAS D., SOYLU D., YASUNAGA M., ZHANG Y., NARAYANAN D., WU Y., KUMAR A. *et al.* (2022). Holistic evaluation of language models. *arXiv preprint arXiv :2211.09110*.

LIN C.-Y. (2004). ROUGE : A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, p. 74–81, Barcelona, Spain : Association for Computational Linguistics.

LIU N. F. & … (2023). Lost in the middle : How language models use long contexts.

MA E. (2019). Nlp augmentation. https ://github.com/makcedward/nlpaug.

MANNING C. D., RAGHAVAN P. & SCHÜTZE H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

MATHUR N., BALDWIN T. & COHN T. (2020). Tangled up in bleu : Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv :2006.06264*.

OPENAI (2023). Gpt-4 technical report.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In P. ISABELLE, E. CHARNIAK & D. LIN, Éds., *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318, Philadelphia, Pennsylvania, USA : Association for Computational Linguistics. DOI : 10.3115/1073083.1073135.

RAJAEI D., TAHERI Z. & FANI H. (2024). No query left behind : Query refinement via backtranslation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, p. 1961–1972.

RAJI I. D., BENDER E. M., PAULLADA A., DENTON E. & HANNA A. (2021). AI and the everything in the whole wide world benchmark. *CoRR*, **abs/2111.15366**.

REITER E. (2018). A structured review of the validity of BLEU. In *Computational Linguistics*.

RIBEIRO M. T., SINGH S. & GUESTRIN C. (2016). "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier. In *KDD*.

RIBEIRO M. T., WU T., GUESTRIN C. & SINGH S. (2020). Beyond accuracy : Behavioral testing of nlp models with checklist. *arXiv preprint arXiv :2005.04118*.

SELLAM T., DAS D. & PARIKH A. P. (2020). Bleurt : Learning robust metrics for text generation. *arXiv preprint arXiv :2004.04696*.

SISKA C., MARAZOPOULOU K., AILEM M. & BONO J. (2024). Examining the robustness of llm evaluation to the distributional assumptions of benchmarks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 10406–10421.

TEAM G., ANIL R., BORGEAUD S., ALAYRAC J.-B., YU J., SORICUT R., SCHALKWYK J., DAI A. M., HAUTH A., MILLICAN K. *et al.* (2023). Gemini : a family of highly capable multimodal models. *arXiv preprint arXiv :2312.11805*.

TOUVRON H., LAVRIL T., IZACARD G., MARTINET X., LACHAUX M.-A., LACROIX T., ROZIÈRE B., GOYAL N., HAMBRO E., AZHAR F. *et al.* (2023). Llama : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*.

WANG A., PRUKSACHATKUN Y., NANGIA N., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2019). Superglue : A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, **32**.

WANG A., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. R. (2018). Glue : A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv :1804.07461*.

WANG T., HE Z., YU W.-Y., FU X. & HAN X. (2024). Large language models are good multilingual learners : When llms meet cross-lingual prompts. *arXiv preprint arXiv :2409.11056*.

WATAOKA K., TAKAHASHI T. & RI R. (2024). Self-preference bias in llm-as-a-judge.

WEI J. & ZOU K. (2019). Eda : Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP-IJCNLP*.

WU H., ZHANG Z. & WU Q. (2021). Exploring syntactic and semantic features for authorship attribution. *Applied Soft Computing*, **111**, 107815. DOI : https ://doi.org/10.1016/j.asoc.2021.107815.

WU T., TERRY M. & CAI C. J. (2022). Ai chains : Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, p. 1–22.

ZHANG T., KISHORE V., WU F., WEINBERGER K. Q. & ARTZI Y. (2019). Bertscore : Evaluating text generation with bert. *arXiv preprint arXiv :1904.09675*.

ZHAO W., PEYRARD M., LIU F., GAO Y., MEYER C. M. & EGER S. (2019). Moverscore : Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv :1909.02622*.

ZHENG L., CHIANG W.-L., SHENG Y., ZHUANG S., WU Z., ZHUANG Y., LIN Z., LI Z., LI D., XING E. P., ZHANG H., GONZALEZ J. E. & STOICA I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena.