# SEBRAG: Vers l'Utilisation des LLM pour une Tâche de Questions-Réponses Extractive

Quentin Signé[1, 2] Thiziri Belkacem[1] Jose G. Moreno[2] Mohand Boughanem[2]
(1) Airbus Protect, 36 Rue Raymond Grimaud, 31700 Blagnac, France
(2) Université de Toulouse, IRIT UMR 5505, Toulouse, France
quentin.signe@airbus.com, thiziri.belkacem@airbus.com,
jose.moreno@irit.fr, mohand.boughanem@irit.fr

RÉSUMÉ

L'émergence des grands modèles de langage (LLM) a révolutionné le domaine des questions-réponses (QR). Cependant, leur tendance à halluciner représente un défi majeur en recherche d'information (RI), notamment en domaines critiques comme la maintenance aéronautique. Pour répondre à cette problématique, cet article explore la capacité des LLM pour des tâches de QR extractives, à l'instar des modèles encodeurs. Ainsi, nous proposons une approche de génération augmentée par recherche d'information (RAG) utilisant un outil d'extraction de chaînes de caractères, permettant au LLM d'extraire une réponse plutôt que de la générer. Les expériences réalisées sur un jeu de données de maintenance aéronautique révèlent que cette approche permet de mieux contrôler l'hallucination par rapport aux méthodes RAG traditionnelles, tout en gardant une précision comparable aux modèles encodeurs extractifs. Cette approche montre son potentiel pour des applications hautement techniques où la précision et la fiabilité sont primordiales.

ABSTRACT

**SEBRAG : Towards the Use of LLMs for an Extractive Question-Answering Task**

The emergence of Large Language Models (LLMs) has revolutionized the Question-Answering (QA) domain. However, their tendency to hallucinate, i.e., generate incorrect information, raises a significant challenge in Information Retrieval (IR), especially in critical fields such as aircraft maintenance. To address this challenge, this paper explores the capacity of LLMs to perform extractive QA, similarly to encoder models. We propose a novel Retrieval Augmented Generation (RAG) approach that leverages a substring extraction tool, allowing the LLM to extract a response rather than generate it. Experiments conducted on an aerospace maintenance dataset reveal that this approach allows for better control of hallucination than traditional RAG methods, with precision comparable to extractive encoder models. This approach demonstrates its potential in highly technical applications where precision and reliability are essential.

MOTS-CLÉS : Génération Augmentée par Recherche d'Information ; Questions-Réponses ; Maintenance Technique.

KEYWORDS: Retrieval Augmented Generation ; Question Answering ; Technical Maintenance.

ARTICLE : **Accepté à** CORIA 2025.

# 1 Introduction

Dans cet article, nous traitons une problématique de la mise en place d'un système de questions-réponses (QR) pour rendre plus efficace l'exécution des tâches de maintenance aéronautique. Dans ce domaine, la précision des systèmes de QR est cruciale. En effet, les opérateurs de maintenance s'appuient sur de nombreux manuels certifiés afin de garantir la sécurité et l'efficacité des opérations réalisées. Par conséquent, les systèmes de QR utilisés pour accéder à ce type de collections doivent éviter toute génération d'informations inexactes et privilégier l'extraction d'information provenant de documentations ayant été certifiées.

L'émergence des grands modèles de langage (LLM) a révolutionné les systèmes de QR, dans lesquels ces modèles sont devenus des composants clés des technologies actuelles. Ainsi, dans l'objectif de construire un système de QR appliqué à la maintenance aéronautique, nous avons choisi d'adapter une approche basée sur les LLM. Cependant, une limitation majeure de ces modèles est leur tendance à générer du contenu apparemment crédible, mais sans fondement (Tonmoy *et al.*, 2024), un phénomène connu sous le nom d'hallucination (Ji *et al.*, 2023; Zhang *et al.*, 2023). Ce comportement peut avoir de graves conséquences dans des domaines critiques [1] tels que la maintenance aéronautique. Dans ce dernier, chaque document est certifié pour garantir la sécurité des opérations et se conformer aux réglementations des autorités compétentes, rendant ainsi l'hallucination des modèles intolérable.

Afin d'atténuer ce problème, des approches ont été proposées, notamment la génération augmentée par recherche d'information (RAG, pour *Retrieval Augmented Generation*) (Lewis *et al.*, 2020; Gao *et al.*, 2023) qui s'est avérée prometteuse pour réduire les risques d'hallucination (Shuster *et al.*, 2021; Gao *et al.*, 2023). Néanmoins, ces méthodes RAG ne pallient que partiellement ce phénomène, en particulier en raison des limites intrinsèques des capacités de génération des LLM qui ne permettent pas d'éliminer les risques d'hallucination. Ce risque résiduel est considéré comme inacceptable dans les domaines critiques soumis à de nombreuses normes. Notamment, la sécurité des opérations aéronautiques repose sur la certification des manuels de maintenance aéronautique (AMM, pour *Aircraft Maintenance Manual*), documents de référence utilisés dans le système QR en cours de développement. Par conséquent, pour garantir l'absence d'erreurs potentiellement fatales, il est impératif que toute réponse fournie par le système soit extraite directement et exclusivement de ces AMM.

Pour pallier les limites des méthodes RAG traditionnelles, nous proposons SEBRAG (*Substring Extraction-Based RAG*), une nouvelle approche combinant un LLM avec un outil d'extraction de chaînes de caractères dans un pipeline RAG. Cette méthode est inspirée par les approches proposant des LLM outillés telles que *Toolformer* (Schick *et al.*, 2023). Cependant, SEBRAG diffère par son objectif et son implémentation. Alors que *Toolformer* vise à améliorer les capacités des LLM en entraînant les modèles à faire appel à diverses API, SEBRAG utilise un unique outil d'extraction de chaînes de caractères, avec pour objectif principal de forcer l'extraction de segments de texte plutôt que de générer des réponses susceptibles de contenir des inexactitudes. De plus, SEBRAG n'implique pas de réentraînement pour apprendre à utiliser l'outil. L'outil d'extraction, dans notre cas, est une fonction prédéfinie dont l'utilisation par le LLM est induite par le prompt. Cette approche vise ainsi à minimiser les hallucinations en s'appuyant exclusivement sur les informations présentes dans les documents sources.

Nous avons évalué SEBRAG à l'aide d'un jeu de données spécifiquement créé pour une tâche de QR

1. Un domaine critique est un secteur d'activité où chaque erreur, défaillance ou mauvaise décision peut avoir des conséquences graves, voire irréversibles, sur la sécurité des personnes, l'intégrité des systèmes ou la continuité des opérations.

et composé de 100 requêtes techniques basées sur un véritable AMM [2]. Les résultats expérimentaux montrent que SEBRAG réduit le risque d'hallucination par rapport aux méthodes RAG traditionnelles et obtient des performances comparables à celles des modèles encodeurs extractifs.

Dans cet article, nos contributions sont les suivantes :

— Une méthode extractive et novatrice pour réduire les hallucinations dans les modèles de QR, basée sur un outil d'extraction de chaînes de caractères ;
— Un nouveau jeu de données spécifiquement conçu pour évaluer les systèmes de QR dans le domaine de la maintenance aéronautique ;
— Des évaluations expérimentales des performances de SEBRAG et de sa capacité à réduire les risques d'hallucination.

## 2 État de l'Art

Les récentes avancées dans les tâches de questions-réponses ont conduit à l'adoption généralisée des approches RAG. Ces méthodes peuvent être classées en trois catégories : naïves, avancées ou modulaires (Gao *et al.*, 2023). Les RAG naïfs suivent un pipeline traditionnel qui comprend l'indexation des documents, la recherche de contextes pertinents et la génération de réponses (Lewis *et al.*, 2020). Cependant, ces méthodes présentent plusieurs inconvénients, en particulier leur tendance à halluciner en générant des informations non présentes dans les contextes extraits (Gao *et al.*, 2023; Tonmoy *et al.*, 2024). Pour pallier cette limitation, des variantes plus sophistiquées des approches RAG ont été développées. Ainsi, les RAG avancés intègrent des optimisations telles que des techniques de pré-recherche et post-recherche d'information, tandis que les RAG modulaires s'appuient sur ces améliorations et introduisent des modules spécialisés pour optimiser le pipeline RAG (Gao *et al.*, 2023). En pratique, ces approches RAG améliorées intègrent diverses techniques, telles que les Graphes de Connaissances (GC) (Wang *et al.*, 2024; Guo *et al.*, 2024; Edge *et al.*, 2024; Peng *et al.*, 2024), la recherche d'information multi-tours (Yang *et al.*, 2024) ou encore la compression des entrées des LLM et l'amélioration de la traçabilité des informations générées (Hofstätter *et al.*, 2023).

Bien que ces méthodes offrent une meilleure maîtrise de l'hallucination que les LLM classiques grâce à leur recherche d'informations contextuelles (Shuster *et al.*, 2021), elles restent susceptibles de générer de l'hallucination. Aucune méthode existante ne parvient à supprimer tout risque d'hallucination (Tonmoy *et al.*, 2024), ainsi, la réduction de ce phénomène dans les domaines techniques demeure un défi important, en particulier lorsque les données d'entraînement des modèles de langage ne contiennent pas le vocabulaire spécialisé du secteur d'activité (Sharma *et al.*, 2024). Toutefois, les approches RAG et leurs variantes restent couramment employées pour améliorer le contrôle des risques d'hallucination (Tonmoy *et al.*, 2024). Par conséquent, de nombreuses recherches récentes se penchent sur l'optimisation des méthodes RAG afin de résoudre cette problématique en domaine spécifique. Notamment, Sharma *et al.* (2024) proposent d'affiner les LLM et d'augmenter les requêtes, tandis que Elaraby *et al.* (2023) présentent une approche professeur-élève, permettant d'entraîner des modèles plus petits sur des connaissances spécifiques. En outre, une méthode prometteuse pour améliorer l'extraction des réponses, et ainsi réduire les risques d'hallucination, consiste à utiliser des outils externes en combinaison avec les LLM. Le modèle *Toolformer* (Schick *et al.*, 2023) montre comment les LLM peuvent décider, de manière autonome, de faire appel à des outils externes afin

2. Le document source est accessible ici : https://github.com/quentin-sgn/SEBRAG/blob/main/AS-AMM-01-000_I1_R1_20180202.pdf

d'améliorer leur précision. Cette utilisation d'outils externes par les LLM est un domaine de recherche en pleine expansion, visant à étendre les capacités de ces modèles en améliorant par exemple leur acquisition des connaissances, leur expertise, leur efficacité et leur adaptabilité, tout en renforçant la robustesse des réponses et la transparence du processus de génération (Qu *et al.*, 2025). Ainsi, Shen *et al.* (2023) proposent *HuggingGPT*, un système utilisant un LLM comme orchestrateur dont le rôle est de sélectionner le modèle expert à utiliser pour accomplir des tâches complexes. Ces modèles experts sont considérés comme des outils auxquels le LLM principal doit faire appel. Un défi majeur de l'utilisation de ces outils est de pouvoir en utiliser plusieurs de manière séquentielle, menant à des approches telles que *ToolChain** (Zhuang *et al.*, 2023) qui représente l'espace des actions possibles sous la forme d'un arbre décisionnel et utilise des algorithmes de recherche pour optimiser le processus de décision du LLM lors de l'utilisation d'outils. Dans ce contexte, notre approche SEBRAG ne vise pas une utilisation d'un large éventail d'outils, mais se concentre sur l'intégration d'un outil bien précis d'extraction de chaînes de caractères dans un pipeline RAG. L'objectif principal est d'adresser la problématique des risques d'hallucination et d'améliorer la factualité des réponses dans les domaines critiques, en contraignant le LLM à extraire une réponse plutôt qu'à la générer librement.

Des études récentes proposent des systèmes basés sur diverses approches pour des tâches de QR dans des domaines liés à l'aéronautique. Par exemple, Agarwal *et al.* (2022) démontrent l'efficacité de l'utilisation conjointe d'un graphe de connaissance avec un modèle basé sur BERT (Devlin, 2018) pour extraire des segments de texte pertinents de documents relatifs à la sécurité aérienne. Cette méthode extractive correspond parfaitement aux défis posés par l'utilisation de systèmes de QR pour de la documentation certifiée. Ainsi, l'extraction directe des réponses dans les documents plutôt que leur génération contribue à réduire les risques d'hallucination dans ces systèmes. Bien que cette approche améliore les performances de leur approche de référence, elle est surpassée par GPT-3 (Brown *et al.*, 2020) en termes de rappel et de précision. Par ailleurs, Li *et al.* (2022) proposent une méthode basée sur des ontologies. Cette dernière s'appuie sur de la reconnaissance d'entités nommées et de l'extraction de relations entre entités, pour améliorer un système de QR destiné à la maintenance des commandes de vol, accélérant ainsi l'exécution des tâches associées.

Les recherches présentées dans cet article se concentrent sur le domaine de la maintenance aéronautique, un secteur technique dans lequel les systèmes de questions-réponses ont été explorés depuis des décennies (Waltz, 1978; Rinaldi *et al.*, 2004). Dans ce contexte, nous présentons une méthode RAG avancée et extractive, inspirée de *Toolformer* (Schick *et al.*, 2023), permettant aux LLM d'utiliser un outil d'extraction de segments de texte pour mieux contrôler l'hallucination dans les systèmes RAG.

# 3 Questions-Réponses avec un RAG basé sur l'Extraction de Chaînes de Caractères

Notre approche RAG basée sur l'extraction de chaînes de caractères (SEBRAG) s'appuie sur une méthode RAG traditionnelle dans laquelle un LLM a accès à un outil d'extraction de chaînes de caractères. Ainsi, au lieu de générer une réponse directe à la requête de l'utilisateur, l'objectif du modèle de langage est de déterminer et de générer les valeurs des paramètres de l'outil. L'appel à la fonction de l'outil, avec les arguments générés, correspond à la portion de texte que le LLM souhaite extraire pour répondre à la requête. Nous définissons la fonction d'extraction de chaînes de caractères

de l'outil, substring, que le LLM devra utiliser, comme suit :

$$\text{substring}(id_{\text{document}}, pos_{\text{premier mot}}, pos_{\text{dernier mot}})$$

où $id_{\text{document}}$ est l'identifiant du document du contexte de la méthode RAG, et $pos_{\text{premier mot}}$ et $pos_{\text{dernier mot}}$ sont les limites de la portion de texte à extraire, correspondant respectivement à la position du premier et du dernier mot de la chaîne de caractères. L'identifiant d'un mot est défini par son index après une tokenisation sur les espaces et la ponctuation, comme illustré ci-dessous.

Pour implémenter l'appel de fonction de l'outil, nous utilisons la bibliothèque Hugging Face[3], qui permet de demander au LLM d'utiliser une fonction prédéfinie comme sortie, dans le format standardisé suivant :

```
{
   "name": nom de la fonction,
   "parameters": dictionnaire des arguments et leur valeur associée
}
```

Par exemple, pour la requête *« What is this aircraft used for ? »* et le document du contexte suivant :

```
{
   "doc1": {
      "content": "The aircraft WT9 Dynamic LSA / Club is a single engine, two seat
         (arranged side by side), cantilever low wing aircraft with a cruciform
         tail. [...] The aircraft is intended for sporting, recreation and tourist
         flying and is approved for VFR day operation only.",
      "numbering": {
         1: "The", 2: "aircraft", ..., 149: "sporting,", 150: "recreation",
         151: "and", 152: "tourist", 153: "flying", ..., 161: "only."
      }
   }
}
```

La réponse attendue du LLM serait :

```
{
   "name": "substring",
   "parameters": {
      "id_document": "doc1",
      "pos_premier mot": 149,
      "pos_dernier mot": 153
   }
}
```

Cette sortie correspond au texte extrait *« sporting, recreation and tourist flying »* après l'exécution de la fonction d'extraction de chaîne de caractères. Pour inciter le LLM à générer cet appel de fonction avec les paramètres adéquats, nous fournissons dans le prompt du LLM : une description de la tâche, une présentation de l'outil `substring` et sa signature, des exemples d'utilisation de cette fonction et l'instruction au modèle de formuler sa réponse sous la forme d'un appel à cet outil.

La figure 1 illustre un RAG classique, qui servira de référence pour l'évaluation, ainsi que l'architecture de nos approches proposées, SEBRAG et E-SEBRAG :

3. https://huggingface.co/blog/unified-tool-use

— **SEBRAG** : Cette configuration utilise le même RAG que la référence, tout en donnant accès au LLM à l'outil d'extraction de chaîne de caractères. Ainsi, dans cette approche, le LLM reçoit une question d'un utilisateur et les passages pertinents récupérés par le module de recherche d'information. Ce LLM est ensuite instruit par le prompt de formuler sa réponse non pas en générant du texte libre, mais en déterminant et en générant les valeurs des arguments de la fonction substring, soit $id_{\text{document}}$, $pos_{\text{premier mot}}$ et $pos_{\text{dernier mot}}$. La sortie du LLM est donc directement l'appel à la fonction, spécifiant l'identifiant du document et les positions de début et de fin du segment permettant de répondre à la question.

— **E-SEBRAG** : Cette configuration est une variante de notre approche SEBRAG opérant en deux temps et également basée sur le RAG classique. Elle implique deux appels distincts au LLM.

  — Premier appel : Le LLM reçoit la question de l'utilisateur et les passages pertinents récupérés par le module de recherche d'information. Il est ensuite sollicité pour générer une réponse textuelle à la question, cette réponse devant être un segment de texte compris dans les passages fournis, sans faire usage de l'outil d'extraction. L'objectif de cette étape est d'identifier le contenu des documents pertinent pour répondre à la question.

  — Second appel : La réponse textuelle générée lors du premier appel, ainsi que les passages pertinents initialement récupérés sont fournis au LLM dans cette seconde étape. Le LLM est alors chargé d'extraire cette réponse textuelle des passages en utilisant l'outil d'extraction substring. La motivation derrière E-SEBRAG est de décomposer la tâche. L'idée est de fournir au LLM, lors de ce second appel, la chaîne de caractères précise à localiser et à extraire. Ceci vise, en théorie, à simplifier l'utilisation de l'outil d'extraction par le modèle en lui donnant une cible explicite.

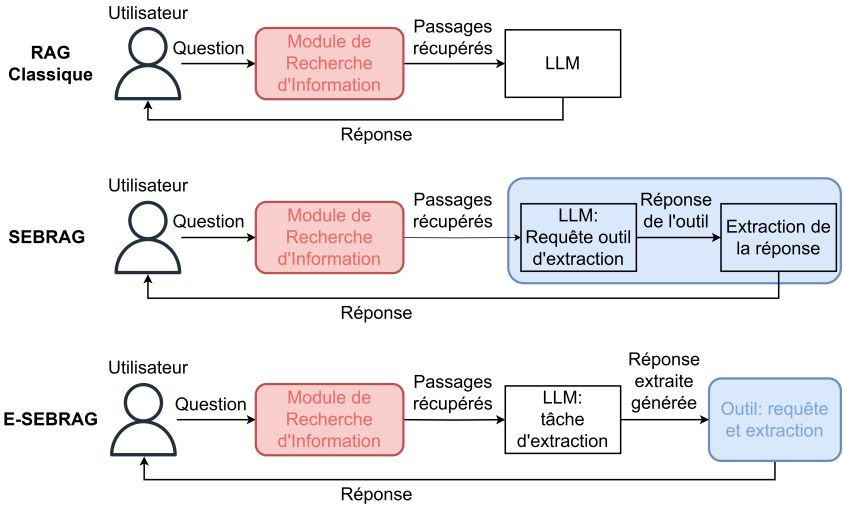


FIGURE 1 – Aperçu des trois approches : RAG classique (référence), ainsi que SEBRAG et E-SEBRAG (méthodes proposées).

L'utilisation de l'approche SEBRAG permet de minimiser les incohérences entre le contexte et la sortie, et ainsi de mieux contrôler les risques d'hallucination, car la réponse renvoyée est directement extraite du document. L'atténuation de ce risque est cruciale dans des contextes techniques et critiques

où il est essentiel de renvoyer des informations issues de documentations officielles.

Bien qu'un risque d'hallucination subsiste encore lors de la génération des valeurs des paramètres de l'outil, notre approche atténue de manière significative ce phénomène. La méthode E-SEBRAG peut également aider le LLM à optimiser l'utilisation de son outil d'extraction de chaîne de caractères en divisant son processus de raisonnement en une tâche de réponse et une tâche d'extraction.

# 4 Résultats Expérimentaux

## 4.1 Jeu de Données de Maintenance Aéronautique

Pour évaluer les performances de SEBRAG, nous avons créé un nouveau jeu de données[4] spécifiquement conçu pour les tâches de questions-réponses en lien avec la maintenance aéronautique. Ce jeu de données est basé sur le manuel de maintenance d'un avion de loisirs[5], fournissant ainsi un contexte pertinent pour évaluer les approches proposées. À l'aide de Gemini 1.5 Pro (Team *et al.*, 2024), nous avons généré 100 paires de questions-réponses techniques à partir de ce manuel. Elles ont ensuite été soigneusement validées manuellement afin de garantir une grande diversité de scénarios et de sujets. De plus, pour assurer une large couverture du manuel, les passages servant de base à la génération des questions par Gemini 1.5 Pro ont été échantillonnés à travers différentes sections du document, évitant une concentration sur des zones spécifiques. En outre, nous avons associé chaque requête à une « réponse extraite », correspondant à un extrait de texte directement issu du document et répondant avec précision à la question. Ces réponses constituent notre référence pour l'évaluation. Les questions dans le jeu de données varient en fonction de la longueur des réponses attendues, allant de simples valeurs numériques (par exemple, des réglages de pression) à des descriptions de procédures complexes (par exemple, la procédure de purge du système de frein). Par ailleurs, les requêtes couvrent divers aspects de la maintenance aéronautique, tels que les spécifications du moteur, les systèmes de carburant ou les configurations des antennes. Ce jeu de données nous permet d'évaluer la capacité de nos approches à extraire des informations pertinentes à partir de documents techniques de maintenance aéronautique. Voici un exemple provenant du jeu de données, comprenant une question accompagnée de sa réponse et de sa réponse extraite :

| | |
|---|---|
| **Question :** | *How to clean the fuel strainer ?* |
| **Réponse :** | *Use compressed air to clean the fuel strainer.* |
| **Réponse Extraite (Vérité terrain) :** | *with compressed air.* |

## 4.2 Métriques et Détails Techniques

Nous avons employé plusieurs métriques couramment utilisées en Traitement Automatique des Langues Naturelles (TALN) (Gao *et al.*, 2023) pour évaluer les performances de nos approches et les comparer aux approches de référence sélectionnées. Ces métriques incluent BLEU (Papineni *et al.*, 2002), ROUGE (Lin, 2004) et ses variantes, ainsi que BERTScore (Zhang *et al.*, 2019). Ces

4. https://github.com/quentin-sgn/SEBRAG/blob/main/QA100_data.json
5. https://github.com/quentin-sgn/SEBRAG/blob/main/AS-AMM-01-000_I1_R1_20180202.pdf

métriques mesurent un score entre une référence et une hypothèse. BLEU et ROUGE sont basées sur le chevauchement des n-grammes et se concentrent respectivement sur la précision et le rappel. D'autre part, BERTScore utilise des *embeddings* contextuels pour calculer une similarité au niveau des tokens, offrant ainsi une évaluation orientée sur la sémantique. De plus, pour évaluer la capacité des LLM à extraire une chaîne de caractères d'un texte en utilisant l'outil, nous avons utilisé l'Exact Match (EM) et le Word Error Rate (WER) (McCowan *et al.*, 2004). Ces métriques nous permettent de mesurer la similarité entre une référence et une hypothèse au niveau des mots. L'EM est une métrique stricte qui vaut 1 si l'hypothèse et la référence correspondent exactement et 0 sinon. En revanche, le WER fournit une évaluation plus nuancée en mesurant la distance entre la réponse du modèle et la référence, en fonction du nombre de mots corrects, supprimés ou insérés. Plus le score de cette métrique est faible et plus les deux textes sont similaires.

Plusieurs études ont développé des métriques visant à évaluer l'hallucination dans les réponses des modèles de questions-réponses. Certaines approches, comme le Hughes Hallucination Evaluation Model (HHEM) (Bao *et al.*, 2024), évaluent la cohérence entre un texte et son résumé. Cette métrique retourne un score compris entre 0 et 1, les valeurs les plus élevées indiquant moins d'hallucinations entre le contexte et le résumé. Ce modèle est entraîné pour estimer la probabilité qu'un texte soit cohérent avec un autre. Nous avons également adopté la métrique de *Faithfulness* (Fidélité) (Ji *et al.*, 2023), qui évalue à quel point une réponse est factuelle par rapport à son contexte. Cette méthode repose sur un *LLM-as-a-judge* (Chiang & Lee, 2023), ce qui signifie qu'elle utilise un LLM externe pour mesurer la métrique, rendant ainsi son processus d'évaluation plus lent. Ainsi, le score de *Faithfulness* est calculé en fonction du nombre d'affirmations dans la réponse que le LLM juge comme n'ayant aucune contradiction avec le contexte. Ce score est défini comme :

$$faithfulness = \frac{|V|}{|S|}$$

où $|V|$ représente le nombre d'affirmations dans la réponse qui sont supportées par le contexte et $|S|$ désigne le nombre total d'affirmations dans la réponse (Es *et al.*, 2023). Nous avons utilisé la bibliothèque DeepEval pour mesurer cette métrique[6].

Le RAG classique, SEBRAG et E-SEBRAG (figure 1) partagent le même pipeline de base, composé d'une base de données FAISS (Douze *et al.*, 2024) contenant le manuel de maintenance découpé en passages et d'un module de recherche d'information utilisant le modèle `all-MiniLM-L6-v2`[7] de la bibliothèque *Sentence Transformers* (Reimers & Gurevych, 2019). Ces méthodes ne diffèrent que par la manière dont le modèle est utilisé et le type de sortie qu'il produit. La référence est un RAG classique qui suit ce pipeline et utilise un LLM pour la génération, tandis que nos approches intègrent en plus l'outil d'extraction de chaînes de caractères.

Nous avons également numéroté les mots dans le contexte retrouvé afin de faciliter l'extraction des segments de texte par le LLM à l'aide de l'outil. Ce processus de numérotation consiste à associer chaque mot du document à sa position dans le texte et à fournir cette correspondance en entrée du modèle, en complément du contenu du document. L'exemple ci-dessous illustre un document présenté sans numérotation, puis avec numérotation des mots.

6. https://docs.confident-ai.com/docs/metrics-faithfulness
7. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Exemple 1 - Document sans numérotation

```
{
  "doc1": {
    "content": "The aircraft WT9 Dynamic
      LSA / Club is a single engine, two
      seat (arranged side by side),
      cantilever low wing aircraft with a
      cruciform tail. [...] The aircraft
      is intended for sporting,
      recreation and tourist flying and
      is approved for VFR day operation
      only."
  }
}
```

Exemple 2 - Document avec numérotation

```
{
  "doc1": {
    "content": "The aircraft WT9 Dynamic
      LSA / Club is a single engine, two
      seat (arranged side by side),
      cantilever low wing aircraft with a
      cruciform tail. [...] The aircraft
      is intended for sporting,
      recreation and tourist flying and
      is approved for VFR day operation
      only.",
    "numbering": {
      1: "The", 2: "aircraft", ...,
      160: "operation", 161: "only."
    }
  }
}
```

Nous avons testé deux LLM distincts quantifiés selon la méthode AWQ (Lin *et al.*, 2024) :

— Llama 3.3 70B Instruct [8]

— DeepSeek-R1 (Guo *et al.*, 2025) distillé dans Llama 3.3 70B Instruct [9]

Le processus de distillation permet à DeepSeek de transférer ses connaissances et ses capacités de raisonnement vers un modèle plus petit (Hinton, 2015), en l'occurrence, Llama 3.3 70B. Ces modèles ont été choisis dans leur version quantifiée afin de les rendre compatibles avec la puissance de calcul que nous avons à disposition. De plus, l'utilisation de modèles de même taille (70 milliards de paramètres) permet une comparaison plus équitable de leurs performances. Dans la suite de cet article, nous désignerons le modèle Llama 3.3 70B Instruct par "Llama 70B" et le modèle DeepSeek-R1 distillé par "DeepSeek 70B".

De plus, afin d'évaluer nos approches par rapport aux méthodes extractives classiques, nous avons intégré deux modèles encodeurs dans notre pipeline RAG de base. Ces modèles, basés sur l'architecture BERT (Devlin, 2018), ont été ré-entraînés pour une tâche de questions-réponses en utilisant le jeu de données SQuAD 2.0 (Rajpurkar *et al.*, 2018). Le premier modèle est basé sur BERT Base [10], et le second sur ModernBERT Large [11], un modèle apportant plusieurs améliorations à l'architecture originale de BERT, notamment la gestion de contextes d'entrée plus longs et l'intégration de Flash Attention (Dao *et al.*, 2022). Ces modèles extractifs sont utilisés à la place du LLM dans le pipeline RAG. Ainsi, ils reçoivent la question et les passages pertinents récupérés par le module de recherche d'information et prédisent directement les indices de début et de fin du segment de texte constituant la réponse conformément à leur entraînement sur des tâches de questions-réponses extractives.

8. https://huggingface.co/casperhansen/llama-3.3-70b-instruct-awq
9. https://huggingface.co/Valdemardi/DeepSeek-R1-Distill-Llama-70B-AWQ
10. https://huggingface.co/deepset/bert-base-cased-squad2
11. https://huggingface.co/Praise2112/ModernBERT-large-squad2-v0.1

## 4.3 Résultats

L'évaluation se concentre sur trois questions de recherche clés : (Q1) Les LLM peuvent-ils utiliser efficacement l'outil d'extraction de chaîne de caractères pour extraire des segments de texte dans le contexte d'un RAG ? (Q2) Quelle est l'exactitude des réponses générées par nos approches dans une tâche de QR par rapport aux modèles encodeurs ? (Q3) Les réponses contiennent-elles des hallucinations ?

**Q1. Évaluation de la Capacité d'Extraction des LLM.** Dans cette première évaluation, notre objectif est d'isoler et de mesurer spécifiquement la capacité intrinsèque des LLM à utiliser l'outil d'extraction de chaînes de caractères pour retrouver un segment de texte explicitement fourni, indépendamment de leur aptitude à d'abord comprendre la question et à identifier la réponse pertinente au sein du contexte. Ainsi, dans cette évaluation, le modèle reçoit cinq extraits du document et est sollicité pour utiliser l'outil d'extraction afin d'extraire une portion de texte prédéfinie, présente dans ces extraits. Cette configuration simule une situation où la réponse a déjà été identifiée (similaire à la sortie de la première étape de E-SEBRAG) et où seule la tâche d'extraction via l'outil est évaluée. La requête adressée au LLM est similaire à :

```
Extrait la chaîne de caractères : <chaîne>.
```

où `<chaîne>` est le segment de texte à extraire, présent dans un des trois passages fournis en contexte.

Le tableau 1 montre que la numérotation des mots améliore considérablement la capacité des modèles à utiliser l'outil d'extraction, comme le reflètent les scores d'EM plus élevés et de WER plus faibles pour les deux modèles de langage. En effet, sans la numérotation, les LLM doivent prédire les positions des mots dans les passages, une tâche qui dépasse actuellement leurs capacités. De plus, DeepSeek 70B surpasse Llama 70B, atteignant la plus grande précision d'extraction, suggérant que les capacités de raisonnement de DeepSeek (Guo *et al.*, 2025) lui permettent d'utiliser l'outil de manière plus précise et efficace, en particulier lorsque la numérotation des mots est fournie.

| Modèle | Numérotation | EM | WER |
|---|---|---|---|
| Llama 70B | Non | 3 % | 129 % |
| | Oui | 59 % | 60 % |
| DeepSeek 70B | Non | 6 % | 147 % |
| | Oui | **76 %** | **21 %** |

TABLE 1 – Évaluation des performances d'extraction des LLM à l'aide de l'outil d'extraction de chaîne de caractères et influence de la numérotation des mots du contexte. Les valeurs en gras indiquent la meilleure performance obtenue pour une métrique spécifique.

**Q2. Évaluation pour une Tâche de Questions-Réponses.** Nous avons ensuite évalué nos approches dans le contexte d'un RAG. Étant donné une requête de notre jeu de données, le module de recherche d'information identifie et sélectionne les $k$ passages les plus pertinents en fonction de leur similarité. Pour les expérimentations de **Q2** et **Q3**, nous avons utilisé $k = 5$ passages qui sont ensuite fournis en contexte du LLM. Les performances des configurations illustrées dans la figure 1 ont été évaluées

en comparant les réponses générées à la vérité terrain en utilisant plusieurs métriques. Nous avons également évalué les méthodes en utilisant un module de recherche d'information dans lequel la « réponse idéale » était garantie d'être parmi les $k$ premiers passages, que nous désignerons, dans la suite de l'article, par « module de recherche d'information parfait ». Afin de déterminer les différences statistiquement significatives dans les résultats des métriques, nous avons réalisé un test de Student entre chaque configuration et le RAG classique pour un même LLM. Nous avons ensuite appliqué la correction de Bonferroni aux valeurs-$p$ obtenues et comparé ces valeurs-$p$ ajustées à un seuil significatif de $\alpha = 5\%$.

| Module de Recherche d'Information | Modèle | Méthode | BS-P | BS-F1 | R1 | R-L | BLEU |
|---|---|---|---|---|---|---|---|
| Classique | BERT base | RAG classique | 0.687 | 0.667 | 0.317 | 0.315 | 0.089 |
| | ModernBERT large | | 0.710 | 0.682 | 0.375 | 0.367 | 0.089 |
| | Llama 70B | RAG classique | **0.732** | **0.746** | **0.477** | **0.458** | **0.227** |
| | | SEBRAG | 0.675* | 0.701* | 0.377* | 0.369* | 0.204 |
| | | E-SEBRAG | 0.664* | 0.695* | 0.364* | 0.353* | 0.215 |
| | DeepSeek 70B | RAG classique | 0.641 | 0.679 | 0.313 | 0.295 | 0.109 |
| | | SEBRAG | 0.672 | 0.670 | 0.365 | 0.354* | 0.203* |
| | | E-SEBRAG | 0.611 | 0.650 | 0.307 | 0.298 | 0.156* |
| « Parfait » | BERT base | RAG classique | 0.721 | 0.699 | 0.415 | 0.412 | 0.115 |
| | ModernBERT large | | 0.783 | 0.750 | 0.562 | 0.564 | 0.141 |
| | Llama 70B | RAG classique | **0.808** | **0.828** | **0.694** | **0.683** | **0.357** |
| | | SEBRAG | 0.706* | 0.741* | 0.568* | 0.520* | 0.330 |
| | | E-SEBRAG | 0.737* | 0.777* | 0.528* | 0.560* | 0.349 |
| | DeepSeek 70B | RAG classique | 0.660 | 0.707 | 0.408 | 0.388 | 0.138 |
| | | SEBRAG | 0.659 | 0.697 | 0.457 | 0.453 | 0.282* |
| | | E-SEBRAG | 0.689 | 0.740* | 0.478* | 0.476* | 0.287* |

TABLE 2 – Évaluation de la performance de la tâche de QR avec le RAG classique et le RAG basé sur l'extraction de chaîne de caractères, avec et sans module de recherche d'information « parfait ». Les valeurs en gras indiquent la meilleure performance obtenue pour un module de recherche d'information donné sur une métrique spécifique. (*) indique des différences statistiquement significatives par rapport au RAG classique à $\alpha = 5\%$. Les métriques utilisées incluent la précision (BS-P) et le score F1 (BS-F1) de BERTScore, ROUGE1 (R1), ROUGE-L (R-L) et BLEU.

Comme le montre le tableau 2, les résultats révèlent des différences de performances notables en fonction du module de recherche d'information et du modèle utilisé. Tout d'abord, nous observons que pour une configuration donnée, les meilleures performances sont obtenues avec le modèle Llama 70B, contrairement aux résultats de la table 1, qui montraient une meilleure performance d'extraction de DeepSeek 70B. Ces résultats s'expliquent par les conceptions distinctes et les forces respectives des deux modèles. Ainsi, DeepSeek 70B surpasse Llama 70B dans les tâches d'extraction d'une chaîne de caractères quelconque, mais cette tendance est inversée dans les tâches d'extraction pour

répondre à une question d'un utilisateur. De plus, le modèle basé sur BERT obtient des performances compétitives pour une tâche de questions-réponses, mais restant toutefois généralement inférieures à celles des deux LLM.

Par ailleurs, les résultats mettent en lumière l'impact du module de recherche d'information sur les performances. Comme attendu, avoir la réponse parmi les passages retournés par le « module de recherche d'information parfait » améliore les performances sur chaque métrique. Enfin, le RAG classique associé à Llama 70B améliore significativement les métriques, notamment BERTScore et ROUGE, que ce soit avec ou sans « module de recherche d'information parfait », comparativement aux approches proposées. Plus précisément, cette combinaison obtient le meilleur score de similarité sémantique (BERTScore) entre la réponse générée et la réponse de référence, ainsi que le meilleur taux de recouvrement lexical (ROUGE). Ces résultats suggèrent que Llama 70B éprouve des difficultés à utiliser efficacement l'outil d'extraction de chaîne de caractères. Cependant, avec le modèle DeepSeek 70B, la performance de l'approche varie en fonction du module de recherche d'information utilisé. Lorsque le module de recherche d'information classique est utilisé, SEBRAG affiche de meilleures performances, obtenant les meilleurs scores ROUGE et BLEU, avec une amélioration statistiquement significative par rapport au RAG classique dans certains cas. Avec un « module de recherche d'information parfait », l'approche E-SEBRAG se révèle la plus performante sur toutes les métriques évaluées.

Ces résultats montrent que les méthodes basées sur l'extraction de chaîne de caractères sont particulièrement efficaces lorsqu'elles sont combinées avec DeepSeek 70B et ses capacités de raisonnement dans un contexte de recherche d'information « parfait ». En revanche, Llama 70B n'a pas besoin de l'outil d'extraction pour atteindre les meilleures performances. Ainsi, la distillation de DeepSeek-R1 dans Llama 3.3 70B a eu un impact négatif sur les performances de ce modèle pour une tâche de questions-réponses. Toutefois, les performances de nos approches s'avèrent comparables à celles des modèles encodeurs extractifs, mettant en évidence la capacité des LLM, lorsqu'ils disposent de l'outil d'extraction, à rivaliser avec les modèles basés sur BERT pour des tâches de questions-réponses extractives.

**Q3. Évaluation des Hallucinations.**

Finalement, nous avons évalué les hallucinations et la fiabilité des réponses générées par nos approches en employant les métriques HHEM et de *Faithfulness*. Les résultats, présentés dans le tableau 3, montrent que les méthodes SEBRAG et E-SEBRAG améliorent la *Faithfulness* et réduisent de manière significative les scores HHEM par rapport au RAG classique associé à un module de recherche d'information classique. Plus précisément, nos approches diminuent les incohérences (HHEM) et le nombre d'affirmations dans la réponse (*Faithfulness*) non soutenues par le contexte. En outre, dans un scénario idéal où le « module de recherche d'information parfait » est utilisé, et pour les deux LLM, nos méthodes présentent une réduction significative des hallucinations, comme le montrent les scores HHEM. Cependant, le RAG classique obtient la *Faithfulness* la plus importante. Par ailleurs, bien que nos approches présentent généralement une *Faithfulness* inférieure aux modèles basés sur BERT, elles les surpassent en termes de score HHEM.

Néanmoins, il convient de noter que les méthodes *LLM-as-a-judge* (Chiang & Lee, 2023), telles que la *Faithfulness* utilisée ici, peuvent présenter des limitations pour évaluer les hallucinations, notamment leurs réponses ne sont pas toujours cohérentes d'un exemple à l'autre. Dans certains scénarios de test, le LLM utilisé pour juger attribue un score de *Faithfulness* de 0, indiquant des hallucinations, même lorsque la réponse évaluée est correcte et supportée par le contexte. Par exemple, lorsque demandé : *« What are the primary methods for moving the aircraft on the ground ? »*, les deux

modèles, dans toutes les configurations, ont correctement répondu *« pushing/pulling, tow bar and taxing »*. Cependant, dans trois des six cas, le LLM utilisé pour juger a incorrectement estimé les réponses comme hallucinées. En revanche, la métrique HHEM a systématiquement obtenu des scores compris entre 0.93 et 0.98, indiquant correctement l'absence d'hallucinations. Ainsi, une méthode *LLM-as-a-judge* peut également introduire des risques d'hallucination lorsqu'elle est utilisée pour juger la *Faithfulness*.

| Module de Recherche d'Information | Modèle | Méthode | *Faithfulness* | HHEM |
|---|---|---|---|---|
| Classique | BERT base | RAG classique | 0.910 | 0.894 |
| | ModernBERT large | | **0.947** | 0.871 |
| | Llama 70B | RAG classique | 0.869 | 0.836 |
| | | SEBRAG | 0.881 | **0.939**$^*$ |
| | | E-SEBRAG | 0.924 | 0.926$^*$ |
| | DeepSeek 70B | RAG classique | 0.862 | 0.794 |
| | | SEBRAG | **0.947** | 0.931$^*$ |
| | | E-SEBRAG | 0.935 | 0.914$^*$ |
| « Parfait » | BERT base | RAG classique | **0.940** | 0.885 |
| | ModernBERT large | | **0.940** | 0.879 |
| | Llama 70B | RAG classique | 0.921 | 0.920 |
| | | SEBRAG | 0.896 | 0.943$^*$ |
| | | E-SEBRAG | 0.914 | 0.943$^*$ |
| | DeepSeek 70B | RAG classique | 0.921 | 0.876 |
| | | SEBRAG | 0.898 | 0.933$^*$ |
| | | E-SEBRAG | 0.900 | **0.947**$^*$ |

TABLE 3 – Évaluation des hallucinations avec les scores de *Faithfulness* et HHEM, avec et sans module de recherche d'information « parfait ». Les valeurs en gras indiquent la meilleure performance obtenue pour un module de recherche d'information donné sur une métrique spécifique. (*) indique des différences statistiquement significatives par rapport au RAG classique à $\alpha = 5\%$.

# 5 Discussion et Conclusion

Cet article présente *Substring Extraction-Based RAG* (SEBRAG), une nouvelle approche RAG pour les systèmes de questions-réponses, spécifiquement conçue pour atténuer le problème d'hallucination dans les domaines critiques tels que la maintenance aéronautique. Cette méthode s'appuie sur un outil d'extraction de chaînes de caractères, permettant au LLM d'extraire directement les informations des documents sources plutôt que de les générer, minimisant ainsi le risque d'introduire des informations erronées. Nous avons également introduit E-SEBRAG, une variante de notre approche qui décompose

le processus en deux étapes : génération d'une réponse puis extraction de cette réponse à l'aide de l'outil. Nous avons évalué nos méthodes sur un jeu de données technique de questions-réponses pour la maintenance aéronautique, spécialement conçu pour évaluer la précision et la fiabilité des réponses générées. Si nos expérimentations montrent que le RAG classique peut obtenir des performances supérieures à celles de nos approches, l'apport principal de SEBRAG et E-SEBRAG réside dans leurs capacités à améliorer la factualité des réponses et à réduire les risques d'hallucination. Ce compromis entre performances et contrôle des hallucinations est particulièrement pertinent dans les domaines critiques où la fiabilité prime. L'évaluation a révélé que SEBRAG et E-SEBRAG obtiennent des performances similaires aux modèles encodeurs sur des tâches extractives tout en réduisant significativement les risques d'hallucination par rapport au RAG classique, comme en témoignent notamment les scores HHEM.

Bien que les résultats obtenus soient prometteurs, une exploration plus approfondie est nécessaire dans cette voie pour exploiter pleinement le potentiel de l'outil d'extraction dans les systèmes de QR. Nous cherchons à étendre nos approches, en particulier pour les cas complexes où la réponse à une requête nécessite l'agrégation d'informations provenant de plusieurs passages du document. Dans de tels scénarios, le LLM devrait être capable d'invoquer l'outil d'extraction plusieurs fois et de combiner les extraits de manière cohérente. Un autre défi concerne les situations dans lesquelles le contexte récupéré par le module de recherche d'information ne contient pas la réponse à la question posée. Dans ce cas, forcer l'extraction d'une réponse à partir d'un contexte non pertinent peut conduire à des hallucinations. Il serait donc nécessaire d'intégrer un mécanisme permettant au LLM de détecter l'absence de réponse dans le contexte et de renvoyer un message approprié à l'utilisateur.

Les travaux futurs se concentreront également à l'obtention d'une évaluation plus complète et comparative, en étendant nos expérimentations à des jeux de données standards pour les QR et pour l'évaluation des hallucinations. Malgré ces limites, nos conclusions offrent une base solide pour les recherches futures dans cette direction novatrice, avec le potentiel de développer des systèmes de QR plus robustes et fiables, capables de réduire les hallucinations et de fournir des réponses précises dans des domaines critiques.

# Références


AGARWAL A., GITE R., LADDHA S., BHATTACHARYYA P., KAR S., EKBAL A., THIND P., ZELE R. & SHANKAR R. (2022). Knowledge graph-deep learning : a case study in question answering in aviation safety domain. *arXiv preprint arXiv :2205.15952*.

BAO F., LI M., LUO R. & MENDELEVITCH O. (2024). HHEM-2.1-Open. DOI : 10.57967/hf/3240.

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.

CHIANG C.-H. & LEE H.-Y. (2023). Can large language models be an alternative to human evaluations ? *arXiv preprint arXiv :2305.01937*.

DAO T., FU D., ERMON S., RUDRA A. & RÉ C. (2022). Flashattention : Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, **35**, 16344–16359.

DEVLIN J. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

DOUZE M., GUZHVA A., DENG C., JOHNSON J., SZILVASY G., MAZARÉ P.-E., LOMELI M., HOSSEINI L. & JÉGOU H. (2024). The faiss library.

EDGE D., TRINH H., CHENG N., BRADLEY J., CHAO A., MODY A., TRUITT S. & LARSON J. (2024). From local to global : A graph rag approach to query-focused summarization. *arXiv preprint arXiv :2404.16130*.

ELARABY M., LU M., DUNN J., ZHANG X., WANG Y., LIU S., TIAN P., WANG Y. & WANG Y. (2023). Halo : Estimation and reduction of hallucinations in open-source weak large language models. *arXiv preprint arXiv :2308.11764*.

ES S., JAMES J., ESPINOSA-ANKE L. & SCHOCKAERT S. (2023). Ragas : Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv :2309.15217*.

GAO Y., XIONG Y., GAO X., JIA K., PAN J., BI Y., DAI Y., SUN J. & WANG H. (2023). Retrieval-augmented generation for large language models : A survey. *arXiv preprint arXiv :2312.10997*.

GUO D., YANG D., ZHANG H., SONG J., ZHANG R., XU R., ZHU Q., MA S., WANG P., BI X. *et al.* (2025). Deepseek-r1 : Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv :2501.12948*.

GUO Z., XIA L., YU Y., AO T. & HUANG C. (2024). Lightrag : Simple and fast retrieval-augmented generation.

HINTON G. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*.

HOFSTÄTTER S., CHEN J., RAMAN K. & ZAMANI H. (2023). Fid-light : Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 1437–1447.

JI Z., LEE N., FRIESKE R., YU T., SU D., XU Y., ISHII E., BANG Y. J., MADOTTO A. & FUNG P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, **55**(12), 1–38.

LEWIS P., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., YIH W.-T., ROCKTÄSCHEL T. *et al.* (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, **33**, 9459–9474.

LI C., YANG X., LUO S., SONG M. & LI W. (2022). Towards domain-specific knowledge graph construction for flight control aided maintenance. *Applied Sciences*, **12**(24), 12736.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out*, p. 74–81.

LIN J., TANG J., TANG H., YANG S., CHEN W.-M., WANG W.-C., XIAO G., DANG X., GAN C. & HAN S. (2024). Awq : Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, **6**, 87–100.

MCCOWAN I. A., MOORE D., DINES J., GATICA-PEREZ D., FLYNN M., WELLNER P. & BOURLARD H. (2004). On the use of information retrieval measures for speech recognition evaluation.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, p. 311–318.

PENG B., ZHU Y., LIU Y., BO X., SHI H., HONG C., ZHANG Y. & TANG S. (2024). Graph retrieval-augmented generation : A survey. *arXiv preprint arXiv :2408.08921*.

QU C., DAI S., WEI X., CAI H., WANG S., YIN D., XU J. & WEN J.-R. (2025). Tool learning with large language models : A survey. *Frontiers of Computer Science*, **19**(8), 198343.

RAJPURKAR P., JIA R. & LIANG P. (2018). Know what you don't know : Unanswerable questions for squad. *arXiv preprint arXiv :1806.03822*.

REIMERS N. & GUREVYCH I. (2019). Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* : Association for Computational Linguistics.

RINALDI F., HESS M., DOWDALL J., ALIOD D. M. & SCHWITTER R. (2004). Question answering in terminology-rich technical domains. In *New directions in question answering*, p. 71–86.

SCHICK T., DWIVEDI-YU J., DESSÌ R., RAILEANU R., LOMELI M., HAMBRO E., ZETTLEMOYER L., CANCEDDA N. & SCIALOM T. (2023). Toolformer : Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, **36**, 68539–68551.

SHARMA S., YOON D. S., DERNONCOURT F., SULTANIA D., BAGGA K., ZHANG M., BUI T. & KOTTE V. (2024). Retrieval augmented generation for domain-specific question answering. *arXiv preprint arXiv :2404.14760*.

SHEN Y., SONG K., TAN X., LI D., LU W. & ZHUANG Y. (2023). Hugginggpt : Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, **36**, 38154–38180.

SHUSTER K., POFF S., CHEN M., KIELA D. & WESTON J. (2021). Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv :2104.07567*.

TEAM G., GEORGIEV P., LEI V. I., BURNELL R., BAI L., GULATI A., TANZER G., VINCENT D., PAN Z., WANG S. *et al.* (2024). Gemini 1.5 : Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv :2403.05530*.

TONMOY S., ZAMAN S., JAIN V., RANI A., RAWTE V., CHADHA A. & DAS A. (2024). A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv :2401.01313*.

WALTZ D. L. (1978). An english language question answering system for a large relational database. *Communications of the ACM*, **21**(7), 526–539.

WANG Y., LIPKA N., ROSSI R. A., SIU A., ZHANG R. & DERR T. (2024). Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, p. 19206–19214.

YANG D., RAO J., CHEN K., GUO X., ZHANG Y., YANG J. & ZHANG Y. (2024). Im-rag : Multi-round retrieval-augmented generation through learning inner monologues. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 730–740.

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

ZHANG Y., LI Y., CUI L., CAI D., LIU L., FU T., HUANG X., ZHAO E., ZHANG Y., CHEN Y. *et al.* (2023). Siren's song in the ai ocean : a survey on hallucination in large language models. *arXiv preprint arXiv :2309.01219*.

ZHUANG Y., CHEN X., YU T., MITRA S., BURSZTYN V., ROSSI R. A., SARKHEL S. & ZHANG C. (2023). Toolchain* : Efficient action space navigation in large language models with a* search. *arXiv preprint arXiv :2310.13227*.