Résumé automatique de textes d'enquêtes judiciaires : retour d'expérience

Thibault Roy

Résumé automatique de textes d'enquêtes judiciaires : retour d'expérience

Thibault ROY
OPPSCIENCE, Paris, France
troy@oppscience.com

RESUME

Cet article présente un travail en cours de réalisation sur le résumé automatique de textes d'enquêtes judiciaires. Ces textes sont de différentes natures : procès-verbaux, comptes-rendus d'audition / d'enquête ou encore décisions de justice. A travers la mise en œuvre d'un processus de résumé automatique de tels textes et son évaluation sur un corpus de test contenant des textes en français et en anglais, nous montrons qu'il est possible d'utiliser de grands modèles de langues (LLMs), sans fine-tuning spécifique à notre contexte, pour atteindre un niveau de pertinence acceptable pour les résumés produits.

ABSTRACT

Abstractive Summarization for Investigative Texts: Experience Feedback.
This article presents a work-in-progress about an abstractive summarization process dedicated to investigative texts. These texts are official reports, examination reports, investigative reports or else court rulings. Through the implementation of an automatic summarization process and its evaluation on a test corpus including French and English texts, we show that it is possible to use large language models (LLMs), without fine-tuning specific to our context, to produce relevant summaries of investigative texts.

MOTS-CLES : Résumé automatique, aide à l'enquête judiciaire, LLM
KEYWORDS : Abstractive summarization, investigation assistance, LLM

Introduction

OPPSCIENCE[1] est un éditeur de logiciels français proposant des solutions d'analyses d'ensembles documentaires. Parmi ces solutions, la solution Spectra[2] est dédiée à l'accès à l'information dans un contexte d'enquêtes judiciaires. Les documents traités par Spectra sont de différentes natures : procès-verbaux, comptes-rendus d'audition / d'enquête, décisions de justice, etc. Des processus de traitement automatique des langues sont mis en œuvre principalement pour extraire des entités nommées, des événements et des relations entre entités / événements. Ces extractions permettent ensuite aux utilisateurs d'accéder à des contenus annotés et à des représentations en graphes des données extraites.

[1] https://oppscience.com
[2] https://oppscience.com/fr/solutions/spectra

En complément à ces extractions, nous cherchons également à produire automatiquement des résumés d'un ou plusieurs textes selon différents critères liés au contexte d'aide à l'enquête. Cet article présente un travail en cours de réalisation dont l'objectif est de valider la possibilité d'utiliser de grands modèles de langues (LLMs), sans *fine-tuning* spécifique au contexte des enquêtes judiciaires, pour produire des résumés pertinents d'un ou plusieurs textes d'enquêtes en anglais et en français. Un tel *fine-tuning* avec des données liées à des enquêtes judiciaires permettrait certainement d'obtenir de meilleurs résultats mais poserait également un très grand nombre de questions aussi bien sur la disponibilité de telles données, sur des aspects éthiques ou encore sur le respect de la vie privée des personnes et organisations mentionnées.

La production de ces résumés se fait sur demande des utilisateurs et doit donc être mise à disposition de ces derniers dans un temps acceptable. Les données d'enquête étant très sensibles, il nous est également impossible d'appeler des services tiers. Le processus de génération de résumés doit être utilisable sur un *hardware* plausible pour nos utilisateurs avec un nombre limité d'unités de calculs GPU.

Dans la première partie de cet article, nous décrivons brièvement la tâche de production des résumés envisagée dans notre contexte d'aide à l'enquête. Nous proposons ensuite une revue des travaux associés. La partie suivante présente nos choix de mise en œuvre. Enfin, une évaluation des résultats obtenus est réalisée avant de conclure sur les travaux réalisés et à venir.

1 Description de la tâche

Dans le cadre d'une enquête judiciaire, les agents ont à leur disposition un important volume de documents hétérogènes. Leur permettre d'accéder à un résumé d'un ou plusieurs de ces documents est très utile pour leur permettre d'en appréhender plus rapidement les éléments essentiels. Dans cet objectif, nous considérons les deux niveaux de résumé détaillés ci-dessous.

Le **premier niveau** consiste à produire un **résumé d'un seul texte**, potentiellement long, lié à l'enquête. Ce résumé doit être court (moins d'une demi-page) et contenir des éléments clefs établis selon les besoins des utilisateurs. Il s'agit d'un certain nombre d'entités, de faits ou d'événements qui sont critiques dans leur processus métier (la définition précise de ces éléments dans le cadre d'une enquête ne peut pas être donnée ici).

Le **deuxième niveau** consiste à produire un **résumé de plusieurs textes**, potentiellement en grand nombre, liés à l'enquête. Ce résumé peut faire jusqu'à quelques pages et doit contenir, selon les besoins de l'utilisateur, les mêmes éléments que pour le premier niveau ou une chronologie des événements avec les différentes entités impliquées.

Pour répondre à de tels besoins, nous présentons dans la partie suivante un état de l'art des procédés de résumé automatique.

2 État de l'art

Des approches par **extraction** et par **abstraction** sont communément utilisées dans les processus de résumé automatique. Pour un état de l'art très complet sur les techniques récentes, nous renvoyons à (Klymenko et al., 2020), (Widyassari et al., 2022) et à (Cajuiero et al., 2023).

L'approche de **résumé par extraction** cherche à construire un résumé d'un texte en sélectionnant certaines parties de ce texte. L'enjeu principal dans cette approche est de sélectionner des séquences pertinentes dans le texte à résumer. Cette sélection est généralement basée sur des critères de position et de longueur des phrases, de présence d'entités nommées, de fréquences des mots, etc. Nous renvoyons à (Saggion et Poibeau, 2012) pour plus de détails sur ces critères. Plus récemment, des techniques de résumé par extraction basées sur des réseaux de neurones (Nallapati et al., 2017) ou sur des LLMs (Zhang H. et al., 2023) ont également été proposées.

Le **résumé par abstraction** (ou résumé par génération) a pour objectif de reformuler le texte d'origine par un texte complétement nouveau reprenant les idées principales de façon condensée. Cette approche, particulièrement suivie avec les progrès de l'IA générative ces dernières années, est considérée comme plus complexe que l'approche par extraction. Toutefois, elle permet de produire des résumés plus lisibles, cohérents et diversifiés, comme le souligne (Retkowski, 2023) qui propose également une revue de techniques de résumé basées sur cette approche par abstraction. Parmi ces techniques, on retrouve notamment des techniques basées sur des modèles *encoder-decoder* comme Pegasus (Zhang J. et al., 2020) ou sur des LLMs dont l'usage est de plus en plus fréquent et offre de bons résultats. Nous renvoyons à (Van Veen et al., 2024) pour un tel usage dans le domaine médical, à (Chang et al., 2024) pour le résumé de longs textes et à (Wang et al., 2023) pour le résumé d'articles de presse focalisés sur les entités nommées et les événements.

Quelle que soit l'approche retenue, la question complexe de l'**évaluation des résumés** produits se pose. La métrique ROUGE (Lin, 2004) est fréquemment utilisée afin de mesurer la proximité lexicale (usage des n-grammes) entre le résumé produit et un résumé de référence. Plus récemment, des métriques comme BertScore (Zhang T. et al., 2020) et BARTScore (Yuan et al., 2021) ont été proposées afin de permettre une meilleure prise en considération des proximités sémantiques entre le résumé généré et un résumé de référence en utilisant des représentations vectorielles contextuelles.

Une autre approche proposée plus récemment consiste à utiliser des **LLMs comme des évaluateurs**, ces évaluateurs sont généralement appelés « juges » (Zheng et al., 2024) ou « jurés » (Verga et al., 2024). Avec cette approche, des LLMs, connus pour leur efficacité dans des tâches de compréhension de la langue, par exemple comme GPT-4, sont interrogés pour savoir si une génération liée à un prompt donné répond à certains critères, comme des critères de pertinence, de complétude, de concision, etc. Dans (Zheng et al., 2024), les retours de ces modèles de langues « juges » sont confrontés à des évaluations humaines et le taux d'accord s'élève à plus de 80%.

3 Implémentation

Compte tenu des contraintes liées aux deux niveaux de résumé que nous envisageons, l'approche par abstraction nous semble la plus appropriée. Cette approche permet de construire de nouveaux contenus résumant un ou plusieurs textes avec les éléments attendus pour chaque niveau. De façon très globale, le processus prend un ou plusieurs textes en entrée, interroge un LLM avec un ou plusieurs prompts dédiés, puis retourne le résumé final à l'utilisateur.

Afin de traiter de longs textes ou plusieurs textes, il est nécessaire de les découper en parties de taille inférieure au contexte maximum autorisé par le LLM utilisé. Dans le cadre de notre implémentation, nous avons dans la mesure du possible privilégié un découpage des textes en parties délimitées prioritairement par des fins de paragraphes et de phrases.

Une fois les textes découpés en parties, deux stratégies sont généralement utilisées (Chang et al., 2024) :

- ***MapReduce*** : les différentes parties sont résumées indépendamment puis un résumé global de ces résumés est produit.
- ***Refinement*** : la première partie est résumée, chacune des parties suivantes est ensuite résumée de façon itérative en prenant en considération à chaque fois le résumé obtenu précédemment.

Le schéma suivant présente le processus utilisé dans nos travaux où nous avons souhaité avoir la possibilité d'expérimenter à la fois les stratégies *MapReduce* et *Refinement*.

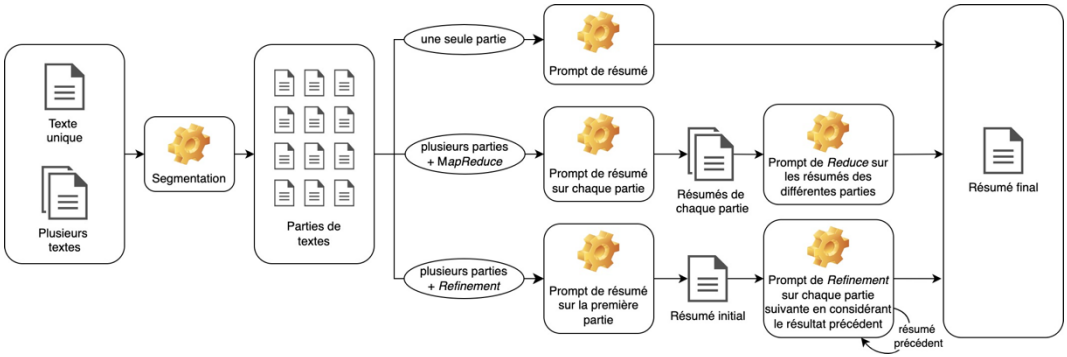

FIGURE 1: Processus de résumé utilisé dans le cadre de nos travaux

Au cours de nos expérimentations, nous avons testé différents prompts en anglais et en français afin de répondre à chaque niveau de résumé envisagé.

Nous donnons respectivement en figures 2 et 3, un exemple de prompt pour les résumés de niveaux 1 et 2 ainsi qu'un exemple de prompt pour la stratégie de *Refinement* pour le résumé de longs / multiples textes.

À noter que nous ne pouvons pas retranscrire ici de façon complète les prompts utilisés dans notre implémentation, les exemples donnés ne font qu'en refléter le principe général.

```
Considère l'intégralité du texte suivant et retourne en sortie un résumé de ce texte
incluant les éléments suivants si disponibles : principales entités, principaux événements,
type et description du crime ou délit commis.

Texte: "{text}"

Sortie:
```

FIGURE 2: Exemple simplifié de prompt en français utilisé pour le résumé de niveaux 1 et 2. Dans ce prompt, la variable `{text}` représente le texte à résumer.

```
Considère l'intégralité du texte suivant et retourne en sortie un résumé de ce texte en
prenant en considération le contexte donné.

Texte : "{text}"

Contexte : "{context}"

Sortie:
```

FIGURE 3: Exemple simplifié de prompt en français utilisé pour la stratégie de *Refinement.* Dans ce prompt, les variables `{text}` et `{context}` représentent respectivement le texte à résumer et le résumé des textes ou parties de textes précédents.

La création de ces prompts a été itérative avec différents essais sur un corpus de test. Des prompts en français et en anglais ont été produits afin de traiter respectivement des textes à notre disposition dans ces langues. Des prompts dans une langue avec un objectif de génération dans une autre langue ont également été testés mais n'ont pas été retenus pour le moment. Nous revenons dans la partie suivante sur les critères d'évaluation qui nous ont permis de retenir les prompts offrant de meilleurs résultats.

Compte tenu du caractère sensible des textes que nous traitons, il ne nous est pas permis d'utiliser des services en ligne pour l'interrogation des LLMs et nous devons donc déployer un LLM localement. Nous avons retenu des modèles avec un nombre restreint de paramètres afin de garantir des temps de réponse inférieurs à une dizaine de secondes sur notre environnement de test[3] proche des environnements de nos clients.

En termes d'implémentation, nous avons choisi de nous baser sur LangChain[4]. Une API FastAPI[5] a été mise en place pour interroger le LLM avec différents prompts et le choix d'une stratégie *MapReduce* ou *Refinement*. Pour certains LLM, la bibliothèque vLLM[6] a été utilisée pour optimiser les temps de génération. Un *tracking* basé sur MLflow[7] a été activé afin de conserver une trace de l'ensemble de nos tests. Une évaluation de cette implémentation sur un corpus de test est détaillée dans la partie suivante de cet article.

4 Évaluation

Comme évoqué précédemment, l'évaluation de résumés produits par des systèmes automatiques fait généralement intervenir des métriques telles que ROUGE et, plus récemment, BertScore et BARTScore. Ces métriques nécessitent d'avoir des données de référence pour être calculées. Dans le cadre de nos travaux, nous n'avons pas de telles données et il est assez difficile et coûteux de les produire.

Nous avons tout d'abord appliqué une évaluation manuelle basée sur des critères reprenant les éléments attendus dans les résumés. Dans un second temps, nous avons expérimenté une approche

[3] 2 x AMD Epyc 7513 - 4 x NVIDIA RTX A5000 – 512 Gb RAM
[4] https://www.langchain.com
[5] https://fastapi.tiangolo.com
[6] https://docs.vllm.ai/en/stable/
[7] https://mlflow.org

basée sur les mêmes critères utilisant un LLM comme un évaluateur (Zheng et al., 2024). Pour mener ces évaluations, nous avons défini des critères, sous forme de questions en français et en anglais, reprenant chaque élément attendu pour les résumés de niveau 1. Nous ne pouvons pas donner ici les critères exacts que nous avons utilisés, la figure 4 en présente des versions légèrement modifiées reflétant le principe général.

```
{
    "criterion_1": "Est-ce que le résumé contient les principales entités du texte ?",
    "criterion_2": "Est-ce que le résumé contient les principaux événements du texte ?",
    "criterion_3": "Est-ce que le résumé contient le type de crime / délit commis ?",
    "criterion_4": "Est-ce que le résumé contient une description du crime / délit commis ?"
}
```

FIGURE 4: Critères d'évaluation simplifiés en français utilisés pour les résumés de niveau 1

Le corpus de test est constitué de 10 textes (5 en anglais, 5 en français) dont la taille est comprise entre 388 et 6 096 tokens. Ces textes sont des rapports d'enquête (2 textes), des procès-verbaux (3 textes) et des décisions de justice (5 textes). Ce corpus est de petite taille, l'évaluation manuelle impliquant une lecture approfondie de chacun des textes afin de valider la pertinence des résumés produits.

Plusieurs LLMs ont été expérimentés dans le cadre de ce travail (versions quantisées ou non) : Mistral 7B, Phi 1.5 et 2, Llama 2 7B et 13B, Stable LM 2 et Palmyra. En termes de temps de traitement, les temps de génération sont naturellement liés à la taille des textes. C'est d'autant plus le cas dans notre approche puisque les textes, dont la taille dépasse le contexte maximum du LLM, sont découpés et plusieurs requêtes sont soumises au LLM. La formule suivante donne le nombre de requêtes avec la stratégie de *Refinement* :

$$\textit{nombre\_de\_requêtes (texte, LLM)} = \textit{partie\_entière(nombre\_de\_tokens(texte) / contexte\_maximum(LLM))} + 1$$

La grille d'évaluation manuelle présentée dans le tableau 1 a été appliquée pour l'ensemble des LLMs testés. Pour chacun des LLMs, nous avons utilisé une valeur de température identique, égale à 0,1. Nous avons utilisé une valeur basse afin de limiter les phénomènes d'hallucinations inhérents aux modèles génératifs (Huang et al., 2023).

La grille d'évaluation du tableau 1 a été obtenue avec le modèle Mistral 7B (version 0.1 du modèle, température égale à 0,1 et usage de la bibliothèque vLLM). C'est avec ce modèle que nous avons obtenu les meilleurs résultats selon les critères de la figure 4. Nous donnons également en figure 5 un exemple de résumé avec ce modèle pour le texte EN_2 (les noms des personnes et des lieux ont été modifiés dans le résumé).

Dans le tableau 1, le symbole « ✅ » signifie que les éléments extraits sont corrects et complets alors que le symbole « ❌ » signifie que les éléments extraits sont incorrects et / ou incomplets. Dans l'ensemble, les résumés produits avec ce modèle sont satisfaisants et aucune hallucination n'a été observée. Nous avons toutefois pu constater des incohérences dans certains résumés où les rôles de personnes avec le même nom de famille ont été inversés. Des événements et des entités importantes sont également manquants de certains résumés.

Textes	Ratio résumé / texte (en caractères)	Temps	Critère 1	Critère 2	Critère 3	Critère 4
EN_1	1 361 / 9 449 = 14,4%	8,7s	✅	✅	✅	✅
EN_2	875 / 6 718 = 13,0%	4,0s	✅	❌	✅	Absent du texte original
EN_3	855 / 2 325 = 36,8%	4,2s	✅	✅	✅	Absent du texte original
EN_4	756 / 7 735 = 9,8%	5,7s	✅	❌	✅	✅
EN_5	715 / 1 672 = 42,8%	5,7s	✅	✅	✅	✅
FR_1	1 733 / 25 396 = 6,8%	17,9s	❌	✅	✅	✅
FR_2	1 040 / 3 974 = 26,2%	4,8s	✅	✅	✅	✅
FR_3	1 573 / 12 493 = 12,6%	4,7s	❌	✅	✅	✅
FR_4	1 161 / 6 252 = 18,6%	6,8s	✅	❌	✅	✅
FR_5	1 726 / 27 082 = 6,4%	17,6s	❌	❌	✅	✅

TABLEAU 1: Évaluation manuelle des résumés produits sur le corpus de test avec Mistral 7B 0.1

```
On March 28, 2021, a team of five surveillance agents monitored the activity of Olkhazar
Marianov, a potentially radicalized individual, at his residence in Pantin. The team arrived
at 7:00 PM and observed Marianov leaving the residence and getting into a grey Toyota Yaris.
They followed him to a hotel, where he met with another man, and then to a tobacco shop,
where they purchased cigarettes and engaged in conversation with the shop owner. Later, a
white Volkswagen and a black Hyundai arrived at the residence, and several other individuals
entered and exited the premises throughout the night. The team requested reinforcements and
documented the license plates of all vehicles in the vicinity. No further activity was
recorded until the team was relieved by the day surveillance team at 7:00 AM. Marianov
remained indoors and out of view throughout the operation.
```

FIGURE 5: Exemple de résumé obtenu

Les autres LLMs testés ont révélé, lors de l'évaluation manuelle, un plus grand nombre d'incohérences dans les résumés produits. Des dates de naissance, des adresses postales et des numéros de téléphones associés à des personnes ont été inventés dans certains résumés. Des prénoms ont été inventés pour des personnes dénommées uniquement par leur nom de famille, de mêmes que des personnes complétement nouvelles ont été mentionnées dans quelques cas. Un lien inventé avec les attaques du 11 septembre 2001 a également été fait dans l'un des résumés. Dans quelques cas, le résumé produit était un contenu en pseudo-code ou reprenait quasiment à l'identique le début du texte original.

Après avoir réalisé cette évaluation manuelle, nous avons cherché à l'automatiser avec une approche utilisant un LLM comme évaluateur. Par cette approche, un LLM de référence, différent du LLM utilisé pour la production des résumés, est interrogé pour chacun des critères d'évaluation définis afin d'évaluer s'il est valide pour le résumé produit compte tenu du texte original.

En termes d'implémentation, nous avons utilisé les fonctionnalités d'évaluation par critère de LangChain[8] en appliquant les critères de la figure 4. Pour chacun d'eux, il est ainsi retourné un

[8] https://python.langchain.com/v0.1/docs/guides/productionization/evaluation/string/criteria_eval_chain/

booléen indiquant s'il est validé ou non, ainsi qu'un texte (appelé « raisonnement ») justifiant ce choix. Ce raisonnement nous est utile pour évaluer les éléments qui ont permis d'aboutir à la réponse retournée.

Un prompt exprime la demande de validation par rapport au texte, au résumé et au critère d'évaluation pris en entrée. Différentes versions de ce prompt en français et en anglais ont été testées afin d'évaluer les résumés respectivement dans ces langues. Le prompt de la figure 6, reprenant certains éléments du prompt par défaut utilisé dans LangChain pour l'évaluation par critère, nous a permis d'obtenir de premiers résultats pertinents.

```
Consider the whole following text and a related summary.

Text: "{input}"

Summary: "{output}"

Criterion : "{criterion}"

Does the summary meet the criterion? First, write out your reasoning to be sure that your
conclusion is correct. Then print only the single character "Y" or "N" corresponding to the
correct answer of whether the summary meets the criterion.
```

FIGURE 6: Prompt en anglais utilisé pour l'évaluation. Les variables `{input}`, `{output}` et `{criterion}` représentent respectivement le texte à résumer, le résumé associé et le critère d'évaluation à considérer.

Pour faire cette évaluation, nous avons utilisé une version quantisée de Llama 2 70B (Touvron et al., 2023). Nous avons choisi ce modèle car nous souhaitions utiliser un modèle à l'état de l'art lorsque nous avons mené cette expérimentation. Le choix de ce modèle a également été motivé par de premiers tests qui ont montré des résultats encourageants avec des prompts d'évaluation en anglais et en français. Toutefois, il faut noter le très faible volume de données en français utilisé lors de l'entrainement de ce modèle : seulement 0,16% des données d'entrainement sont en français alors que 89,70% des données sont en anglais.

Llama 2 70B autorisant un contexte de 32k tokens, nous avons pu lui soumettre les prompts d'évaluation incluant chaque texte et son résumé sans segmentation préalable. Nous avons ainsi appliqué ce prompt sur les 10 textes de notre corpus de test, leurs résumés associés, et les critères d'évaluation définis en figure 4. Une valeur de température égale à 0,1 a été choisie pour cette évaluation.

Les résultats obtenus ont été très mitigés. Sur les 10 textes du corpus de test, des évaluations pertinentes pour certains critères ont été retournées pour seulement 4 textes en anglais et aucun en français. Nous donnons en figure 7 des exemples de résultats pertinents. Dans quelques cas, nous avons également pu observer un raisonnement correct mais une réponse booléenne incohérente avec ce raisonnement (figure 8). A noter que les exemples des figures 7 et 8 ont été légèrement modifiés afin de permettre leur diffusion dans cet article.

```
# Exemple de résultat pour le critère 2
The text mentions the date, location, primary subject, objective, briefing time, surveillance team, vehicles observed, license plates recorded, individuals observed entering/exiting the residence, and time of relief by day surveillance team. All of these events are present in the summary. Therefore, the summary meets the main events criterion.
Answer: Y

# Exemple de résultat pour le critère 3
The summary should contain the crime, which in this case is "terrorism". If the summary does not contain the word "terrorism", then it does not meet the criterion. Therefore, the summary meets the crime type criterion.
Answer: Y

# Exemple de résultat pour le critère 4
The summary contains the crime description (traffic stop and search of vehicle). Therefore, the summary meets the criterion.
Answer: Y
```

FIGURE 7: Exemples d'évaluations pertinentes obtenues pour certains critères

```
# Exemple de résultat pour le critère 1
The summary contains the names of the defendant, Alan Dupont, and the judge, Mélanie Durand. It also mentions the Vitry-sur-Seine Police Department. Therefore, it meets the criterion of containing all the main entities. This indicates that the summary meets the criterion.
Answer: N
```

FIGURE 8: Exemple d'un raisonnement pertinent obtenu lors d'une évaluation mais avec une réponse booléenne incohérente.

Parmi les erreurs fréquemment rencontrées, nous avons principalement observé des hallucinations dans les raisonnements retournés (inventions de personnes, d'événements ou de lieux), des séquences non pertinentes répétées (séquences sans aucune cohérence ou à la syntaxe incorrecte) ou encore du pseudo-code généré. De telles erreurs peuvent notamment s'expliquer par l'usage des très longs prompts induits par la tâche visée ainsi que par le double objectif de génération (production d'un raisonnement et d'une réponse booléenne).

Compte tenu de ces erreurs, une telle évaluation automatique n'est donc pas applicable dans l'état à notre contexte. Toutefois, les résultats pertinents qui ont pu être obtenus (cf. figure 7) nous incitent à poursuivre nos travaux dans cette direction.

Conclusion

Dans cet article, nous avons présenté un travail en cours de réalisation sur la production de résumés dans un contexte d'aide à l'enquête judiciaire. L'enseignement principal que nous tirons de ce travail est qu'il est possible d'utiliser un LLM pour produire des résumés pertinents de textes d'enquêtes, compte tenu des différentes contraintes que nous avons : différents éléments attendus dans le résumé, temps de traitement, pas d'appel de services externes, etc.

Nous avons évalué manuellement la pertinence des résumés produits compte tenu des différents éléments attendus. Le résultat de cette évaluation confirme globalement la pertinence des résumés produits, même sans *fine-tuning* spécifique au contexte des enquêtes judiciaires. Il faut toutefois noter que cette évaluation manuelle a porté sur un petit volume de textes, l'évaluation automatique, qui nous aurait permis d'évaluer plus de textes, n'étant pas utilisable dans l'état. En effet, notre usage d'un LLM « juge » pour évaluer la pertinence des résumés produits a montré les limites importantes de l'approche dans notre contexte mais a également révélé certains raisonnements corrects qui nous encouragent à poursuivre nos travaux sur le sujet.

Il faudra toutefois être particulièrement vigilant dans notre usage des résultats de cette évaluation automatique. Les raisonnements produits par le LLM lors de cette évaluation peuvent être biaisés par les données d'entrainement et / ou se baser sur des éléments inventés. L'évaluation par critère expérimentée ici peut exacerber ce phénomène. Par exemple, le critère vérifiant la présence des principales entités dans le résumé peut amener le modèle à considérer dans son raisonnement des entités non pertinentes (secondaires ou même inventées) et ainsi produire une évaluation pouvant paraître plausible mais qui serait tout de même faussée.

L'évaluation menée dans cet article a porté uniquement sur le premier niveau de résumé que nous avons défini (résumé court d'un seul texte avec certains éléments attendus). Nous continuons actuellement à travailler sur ce niveau, mais aussi sur le second (résumés multi-textes avec différents objectifs). Nous cherchons notamment à améliorer la stratégie de *prompting* afin de prendre en considération les recommandations des utilisateurs sur les résumés produits. Nous testons également des LLMs récents (Mixtral, Llama3, etc.) offrant de meilleures performances que les modèles testés dans cet article. Enfin, une évaluation sur un corpus plus large est envisagée à court terme afin de nous confronter à un plus grand nombre de textes et d'évaluer la robustesse de notre implémentation.

Références

Cajueiro D. O., Nery A. G., Tavares I., De Melo M. K., Dos Reis S. A., Weigang L. & Celestino V. R. R. (2023). A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding, DOI: 10.48550/arXiv.2301.03403.

Chang Y., Kyle L., Goyal T. & Iyyer M. (2024). BooookScore: A systematic exploration of book-length summarization in the era of LLMs. In *The Twelfth International Conference on Learning Representations*.

Huang L., Yu W., Ma W., Zhong W., Feng Z., Wang H., Chen Q., Peng W, Feng X., Qin B. & Liu T. (2023). A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, DOI: 10.48550/arXiv.2311.05232.

Klymenko O., Braun D. & Matthes F. (2020). Automatic Text Summarization: a State-of-the-art Review. In *22nd International Conference on Enterprise Information Systems (ICEIS 2020)* – Volume 1, p. 648-655. DOI: 10.5220/0009723306480655.

Lin C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, p. 74–81. Barcelona, Spain.

Nallapati R., Zhai F., & Zhou B. (2017). SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. In *AAAI*, p. 3075–3081.

Retkowski F. (2023). The current state of summarization. In *Beyond Quantity*, p. 291-312. DOI: 10.1515/9783839467664-016.

SAGGION H. & POIBEAU T. (2013). Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, p. 3–21. Springer.

TOUVRON H., MARTIN L., STONE K., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

VAN VEEN D., VAN UDEN C., BLANKEMEIER L., DELBROUCK J.B., AALI A., BLUETHGEN C., PAREEK A., POLACIN M., REIS E.P., SEEHOFNEROVÁ A., ROHATGI N., HOSAMANI P., COLLINS W., AHUJA N., LANGLOTZ C.P., HOM J., GATIDIS S., PAULY J. & CHAUDHARI A.S. (2024). Adapted large language models can outperform medical experts in clinical text summarization. Nature Medicine. 2024 Apr 30 (4), p. 1134-1142. DOI: 10.1038/s41591-024-02855-5.

VERGA P., HOFSTATTER S., ALTHAMMER S., SU Y., PIKTUS A., ARKHANGORODSKY A., XU M., WHITE N., & LEWIS P. (2024). Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. DOI: 10.48550/arXiv.2404.18796.

WANG Y., ZHANG Z. & WANG R. (2023). Element-aware Summarization with Large Language Models: Expert-aligned Evaluation and Chain-of-Thought Method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Volume 1: Long Papers, pages 8640–8665.

WIDYASSARI A. P., RUSTAD S., SHIDIK G. F., NOERSASONGKO E., SYUKUR A., AFFANDY A. & SETIADI D. R. I. M. (2022). Review of automatic text summarization techniques & methods. In *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, Issue 4, 2022, p. 1029-1046, DOI: 10.1016/j.jksuci.2020.05.006.

YUAN W., NEUBIG G. & LIU P. (2021). BARTScore: Evaluating Generated Text as Text Generation. In *Advances in Neural Information Processing Systems*, Vol. 34, p. 27263–27277.

ZHANG J., ZHAO Y., SALEH M., & LIU P. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning* (pa. 11328–11339).

ZHANG H., LIU X. & ZHANG J. (2023). Extractive Summarization via ChatGPT for Faithful Summary Generation, DOI: 10.48550/arXiv.2304.04193.

ZHANG T., KISHORE V., WU F., WEINBERGER K. Q., & ARTZI Y. (2020). BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations* (ICLR) 2020.

ZHENG L., CHIANG W.-L., SHENG Y., ZHUANG S., WU Z., ZHUANG Y., LIN Z., LI Z., LI D., XING E., ZHANG H., GONZALEZ J. E. & STOICA I. (2024). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, 36.