

CNAM Paris

RCP217

Intelligence artificielle pour des données multimédia

Projet Individuel: Génération de résumés de textes

Peggi ABREU

31 janvier 2026

Table des Matières

1. Introduction	3
2. État de l'art du résumé automatique	3
3. Architecture et méthodologie	4
3. Analyse préliminaire des données	6
4. Méthodologie expérimentale	7
5. Résultats Itération 1 : Baseline Extractive	8
6. Résultats Itération 2 : Abstractif T5-small	10
7. Analyse Comparée Extractif vs Abstractif	12
8. Résultats Itération 3 : Ablation et sensibilité	14
9. Résultats Itération 4 - Hybride extractif+abstractif	15
10.Résultats Itération 5 - Résumé hiérarchique	16
11.Conclusions	18
12.Limites et perspectives	18
Références	19

1. Introduction

1.1. Contexte

Ce projet s'inscrit dans le cadre du cours CNAM RCP217 Intelligence artificielle pour des données multimédia.

1.2. Sujet : Génération de résumé

Enoncé du sujet : “*Il s’agit de produire des résumés de textes à partir d’un corpus de dépêches de CNN qui propose pour chaque document une série de courts résumés. Chacun des documents étant assez clairement séparé en phrases (à raison d’une phrase par ligne), on pourra essayer d’utiliser cette structure pour limiter les problèmes dûs à la taille du texte.*”

1.3. Source de données

Le jeu de données utilisé (CNN_STORIES_TOKENIZED) est disponible sur : <https://github.com/JafferWilson/Process-Data-of-CNN-DailyMail>

Il contient un ensemble de 92850 dépêches CNN sous forme de fichiers “.story”. Chaque fichier “.story” contient une dépêche/article avec le texte complet de l’article, ainsi que des résumés de référence fournis par CNN(ou “highlights”).

1.4. Objectif

L’objectif de ce projet est de générer automatiquement quelques phrases résumant l’essentiel de l’article en explorant différentes solutions afin de pouvoir les comparer entre elles, avec les standards actuels, en utilisant des métriques adaptées.

1.5. Approche

On a choisi d’adopter une approche itérative, afin d’explorer les deux types principaux de génération de résumés. La vue d’ensemble des étapes est présentée dans la section méthodologie expérimentale.

2. État de l'art du résumé automatique

2.1. Méthodes existantes

Le résumé automatique se divise en approches extractives, qui sélectionnent des phrases existantes, et abstratives, qui génèrent de nouvelles formulations. Les méthodes extractives classiques (TextRank, LexRank) préservent généralement le contenu du texte source, mais produisent des résumés souvent peu fluides. L’avènement des modèles Transformers (BERT, GPT) a favorisé le développement de méthodes abstratives reposant sur des architectures encoder-decoder pré-entraînées, telles que BART, PEGASUS et T5.

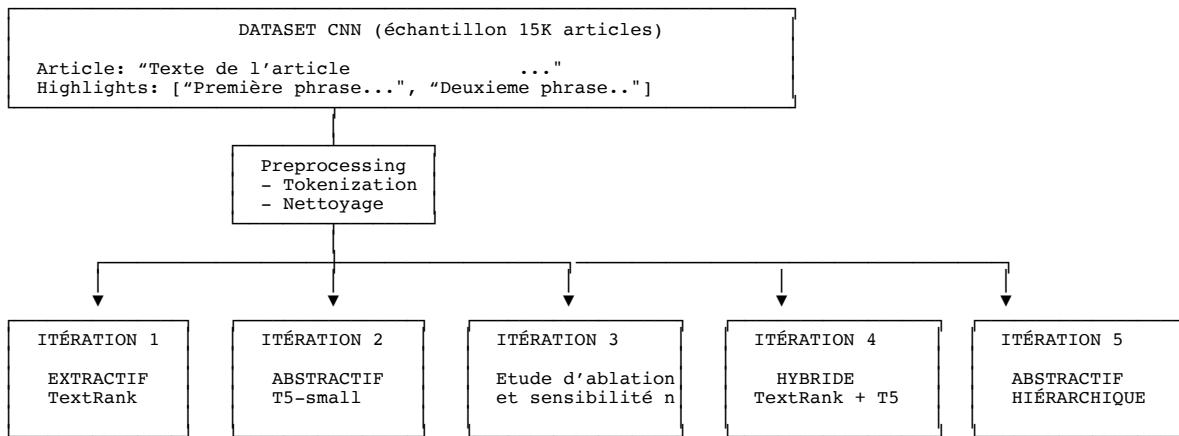
2.2. Métriques d'évaluation

La qualité des résumés est évaluée à l'aide des métriques ROUGE, qui mesurent le chevauchement lexical entre un résumé généré et des résumés de référence. Malgré leurs limites, notamment leur sensibilité limitée aux reformulations et à l'exactitude des informations, ces métriques restent le standard du domaine pour leur simplicité et leur reproductibilité.

3. Architecture et méthodologie

3.1. Vue d'ensemble des itérations

Notre approche s'articule autour d'un pipeline de traitement développé et évalué de façon itérative, au cours de cinq itérations successives au total. La figure ci-dessous illustre l'enchaînement des itérations.



Les étapes principales sont les suivantes :

1. Préparation des données : Chargement et nettoyage du dataset CNN/Daily Mail, avec séparation en ensembles d'entraînement (80%), validation (10%) et test (10%).
2. Application successive des approches sur le même corpus :
 1. Itération 1 - extractive
 2. Itération 2 - abstrajective et évaluation comparative (métriques ROUGE + analyse qualitative)
 3. Itération 3 - étude d'ablation et analyse de sensibilité
 4. Iteration 4 - hybride extractive-abstrajective
 5. Itération 5 - abstrajective hiérarchique

Cette démarche permet une comparaison équitable des approches extractive et abstrajective sur des ensembles de données identiques, avec des métriques standardisées.

2.6. Architecture des modèles

2.6.1. Architecture de TextRank pour le résumé extractif

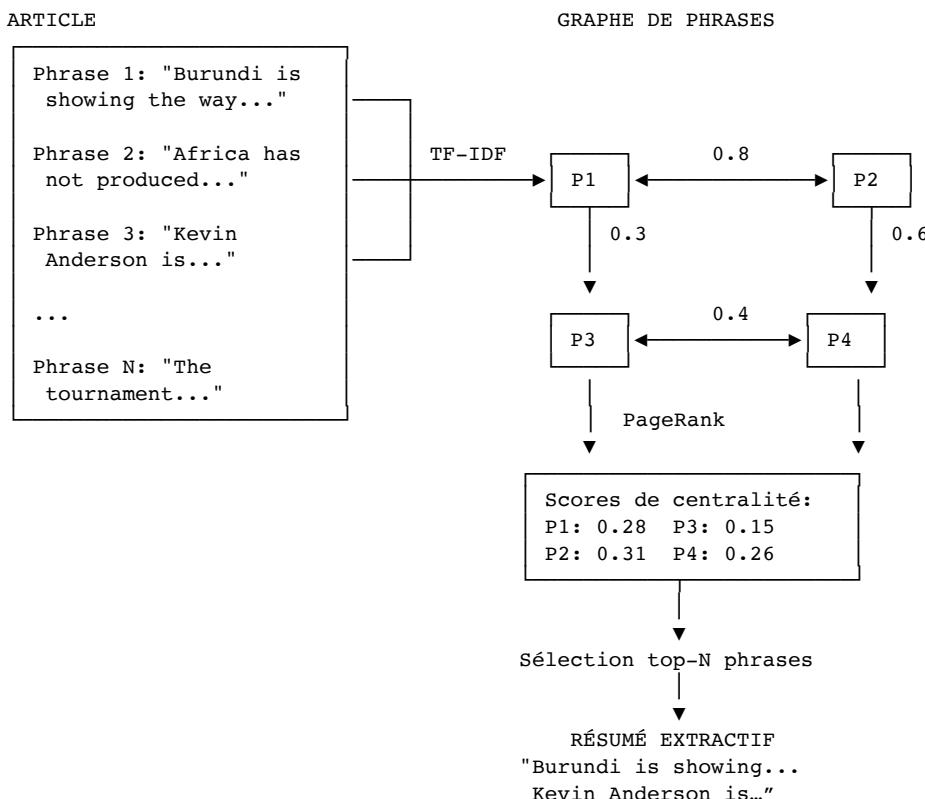
TextRank est un algorithme de résumé extractif inspiré de PageRank, proposé par Mihalcea et Tarau (2004), et adapté au traitement du langage naturel. L'architecture repose sur la construction d'un graphe de phrases où les arêtes représentent la similarité sémantique.

Étapes du modèle :

- ✓ Segmentation : L'article est découpé en phrases (une par ligne dans le dataset CNN).
- ✓ Vectorisation TF-IDF : Chaque phrase est représentée par un vecteur TF-IDF capturant son contenu sémantique, avec suppression des mots vides (stop words).

- ✓ Matrice de similarité: Calcul de la similarité cosinus entre toutes les paires de phrases, avec un seuil de similarité pour éliminer les connexions faibles.
- ✓ PageRank: Application de l'algorithme PageRank sur le graphe résultant pour calculer un score de centralité pour chaque phrase. Les phrases avec les scores les plus élevés sont celles qui sont connectées à plus de phrases; elles sont considérées comme “centrales” (car similaires à beaucoup d’autres phrases dans l’article), et donc comme les plus représentatives du contenu global.
- ✓ Sélection: Extraction des top-N phrases dans leur ordre d'apparition original pour préserver la cohérence narrative.

ARCHITECTURE TextRank



2.6.2. Architecture T5-small pour le résumé abstractif

Pour notre deuxième modèle, le choix s'est porté sur T5, un modèle formulant les tâches de traitement du langage naturel comme des transformations texte-à-texte et intégrant nativement la génération séquentielle. Ce choix a été motivé par sa légèreté, sa disponibilité via la bibliothèque Hugging Face et sa facilité d'intégration dans un cadre expérimental contraint en ressources.

L'approche repose sur du transfer learning en deux étapes. Premièrement, nous utilisons le modèle T5-small qui est un modèle encoder-decoder basé sur l'architecture Transformer. Il a été pré-entraîné par Raffel et al. (2020) sur le corpus C4. Deuxièmement, nous effectuons un fine-tuning supervisé sur le dataset CNN pour spécialiser le modèle à la tâche de résumé abstractif de dépêches journalistiques.

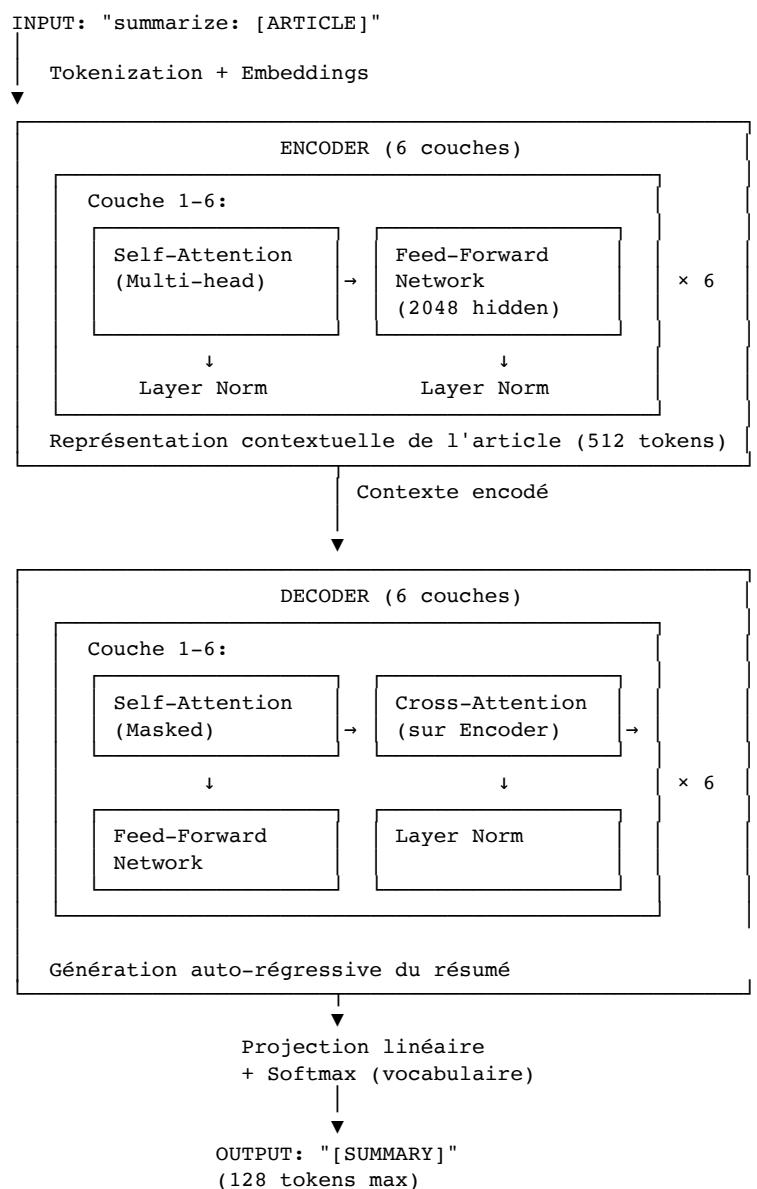
Cette approche par transfer learning permet de bénéficier des connaissances linguistiques générales acquises durant le pré-entraînement (~750 GB de texte web) tout en adaptant le modèle à notre domaine cible avec un corpus d'entraînement beaucoup plus réduit (10K exemples, soit ~50 MB).

Etapes du modèle:

- ✓ Pré-traitement: L'article est fourni en entrée sous la forme d'une instruction textuelle de type “summarize:”, suivie du texte à résumer.

- ✓ Tokenisation et embeddings : Le texte d'entrée est découpé en tokens et projeté dans un espace vectoriel via des embeddings partagés entre l'encodeur et le décodeur.
- ✓ Encodage : L'encodeur Transformer traite la séquence d'entrée et produit une représentation “contextuelle” de l'article, capturant les dépendances entre les mots à l'aide de mécanismes d'auto-attention (self-attention). Le sens d'un mot est interprété en fonction du contexte (tous les autres mots de l'article).
- ✓ Décodage : Le décodeur génère le résumé de manière auto-régressive, en s'appuyant à la fois sur les tokens déjà générés (self-attention masquée) et sur les représentations de l'encodeur via un mécanisme d'attention croisée (Cross-Attention).
- ✓ Génération : À chaque étape, le modèle prédit le token suivant jusqu'à l'obtention du résumé final ou l'atteinte de la longueur maximale fixée.

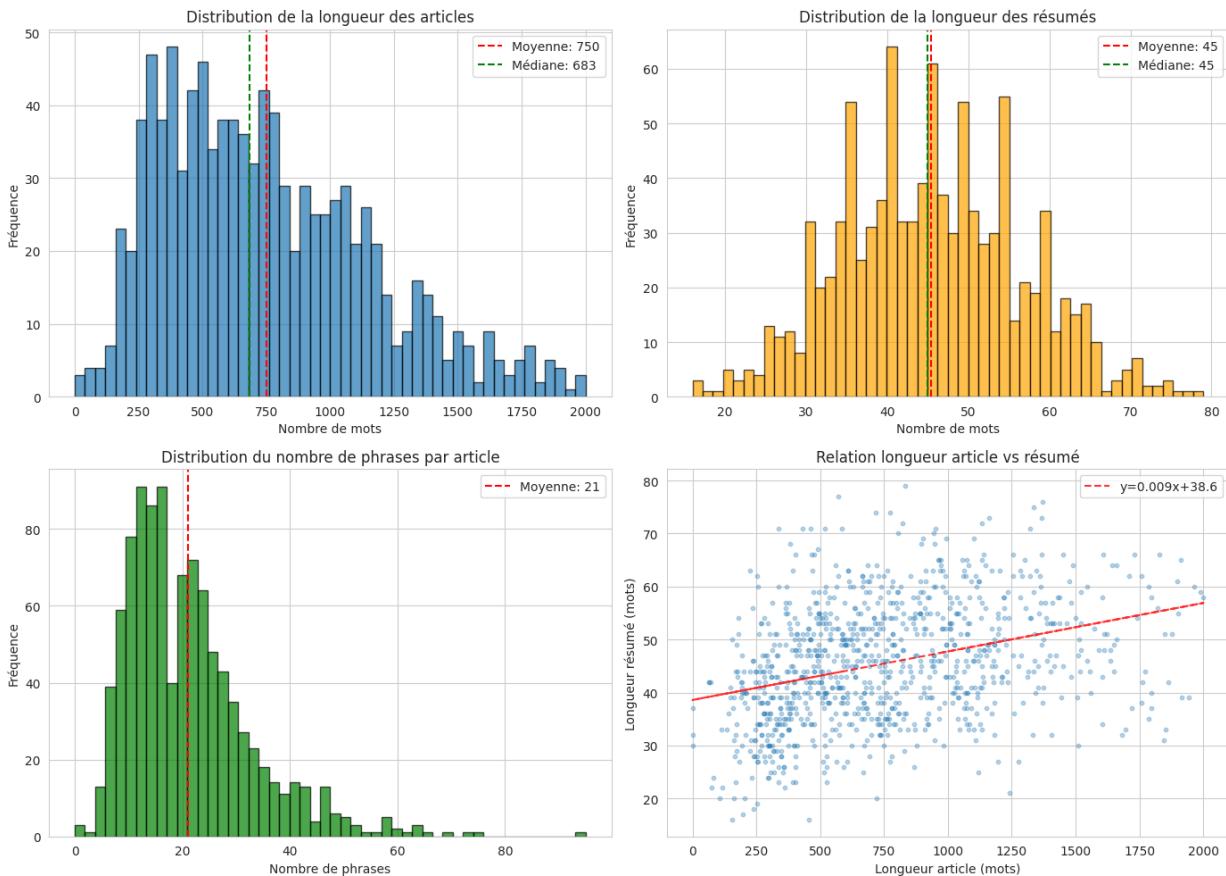
ARCHITECTURE T5-SMALL



3. Analyse préliminaire des données

Le dataset contient au total 92579 fichiers “.story”. En analysant un échantillon de 1000 articles, on obtient les statistiques descriptives suivantes:

- ✓ Articles: 750 mots en moyenne (médiane: 683) et 20 phrases en moyenne
- ✓ Résumés: 45 mots en moyenne (médiane: 45)
- ✓ Taux de compression entre article et résumés de référence : 6.1%
- ✓ Nombre de highlights par article: 3.5
- ✓ Correlation positive entre la longueur du résumé par rapport à la longueur de l’article



4. Méthodologie expérimentale

Un échantillon de 15K articles du dataset CNN a été divisé selon la répartition standard : Train : 12000 exemples (80%), validation : 1500 exemples (10%), test : 1500 exemples (10%).

Ensuite, comme présentée dans la section “Vue d’Ensemble des itérations”, le travail a été décomposé en 5 iterations dont le détail des phases est résumé dans les sections suivantes.

4.1. Itération 1 : Baseline Extractive (TextRank)

Ce premier modèle, le plus simple, nous permet d’avoir une base de référence pour pouvoir comparer les performances avec le modèle T5 de l’IT 2.

L’impléménation a été effectuée en trois phases principales :

- ✓ Phase 1 - Validation du pipeline (500 exemples)
- ✓ Phase 2 - Optimisation hyperparamètres par Grid search (validation sur 1500 exemples).
Configurations testées : $(\text{top_n} \in \{2,3,4,5\} \times \text{threshold} \in \{0.0, 0.1, 0.2\})$

- ✓ Phase 3 - Évaluation finale avec paramètres optimaux (train+val+test = 10000 exemples)

4.2. Itération 2 : Résumé Abstractif (T5-small).

L'implémantation peut se découper aussi en trois phases principales :

- ✓ Phase 1 - Entraînement rapide pour validation du pipeline et calibration initiale (train: 1000 exemples, validation : 200 exemples)
- ✓ Phase 1bis - Évaluation initiale du modèle phase 1 (validation : 500 exemples)
- ✓ Phase 2 - Entraînement complet (entraînement : 10000 exemples, test : 5000 exemples, validation : 1000 exemples).

4.3. Itération 3 : Étude d'ablation et de sensibilité

Afin de comprendre les facteurs déterminants de la performance de notre modèle T5-small, nous avons conduit une étude d'ablation systématique. Pour chaque expérience, un unique hyperparamètre est modifié tandis que les autres restent à leur valeur baseline (configuration IT2 phase 2 finale). L'évaluation est réalisée sur un échantillon stratifié de 2000 articles du test set.

4.4. Itération 4 : Approche hybride (extractive + abstractive)

Cette itération est rajoutée suite à l'étude de sensibilité à la longueur des mots comme solution envisagée aux limites des méthodes précédentes.

4.5. Itération 5 : Résumé hiérarchique (T5-small)

Cette itération explore une solution qui s'appuie sur la structure segmentée des articles CNN (1 phrase par ligne) pour éviter la troncature T5 sur les articles longs.

On utilise une architecture à 2 niveaux. Dans un premier niveau, l'article est découpé en chunks de 10 phrases contiguës. Chaque chunk est résumé indépendamment par T5 (30 mots max). Puis dans un deuxième temps, les résumés de chunks sont concaténés et résumés à nouveau par T5 pour produire le résumé final (50 mots). Cette approche vise à s'assurer qu'aucune information n'est perdue par troncature, en respectant les limites du modèle à chaque étape.

5. Résultats Itération 1 : Baseline Extractive

5.1. IT1 - Phase 1 : Développement (500 exemples)

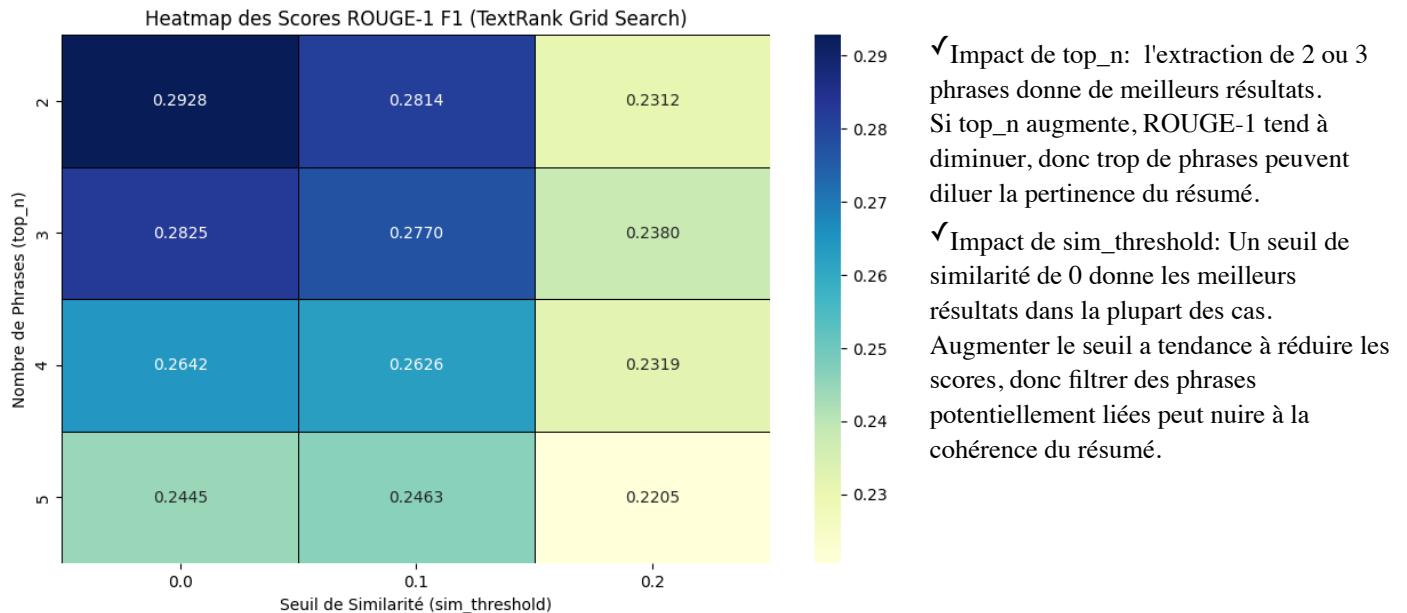
Le tableau suivant donne les scores moyens obtenus pour les différentes métriques.

Métrique	F1-score	Précision	Rappel	Observations
ROUGE-1	0.2807	0.2145	0.4594	F1 = 0.28 montre une Performance correcte sur les mots isolés. Le rappel élevé (0.46) indique que le résumé couvre une bonne partie du contenu de référence, mais la précision plus faible suggère la présence d'informations superflues. En moyenne 28% des mots uniques présents dans les résumés de référence sont capturés par le résumé extractif.
ROUGE-2	0.0953	0.0719	0.1579	F1 = 0.095 est un score faible sur les bigrammes, ce qui signifie que la structure des phrases et l'enchaînement des mots sont peu alignés avec le résumé de référence. C'est fréquent pour des modèles extractifs simples.
ROUGE-L	0.1804	0.1385	0.2961	F1 = 0.18, équivaut à un alignement partiel sur les séquences longues. Le modèle capte certaines structures globales, mais reste limité dans la cohérence globale du résumé.

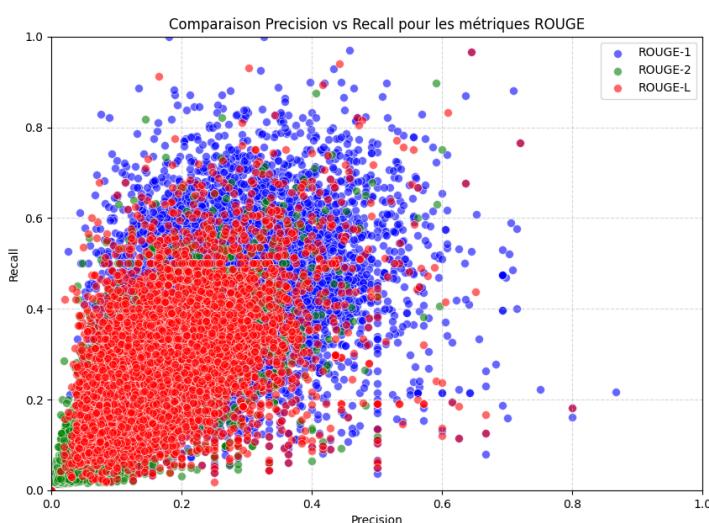
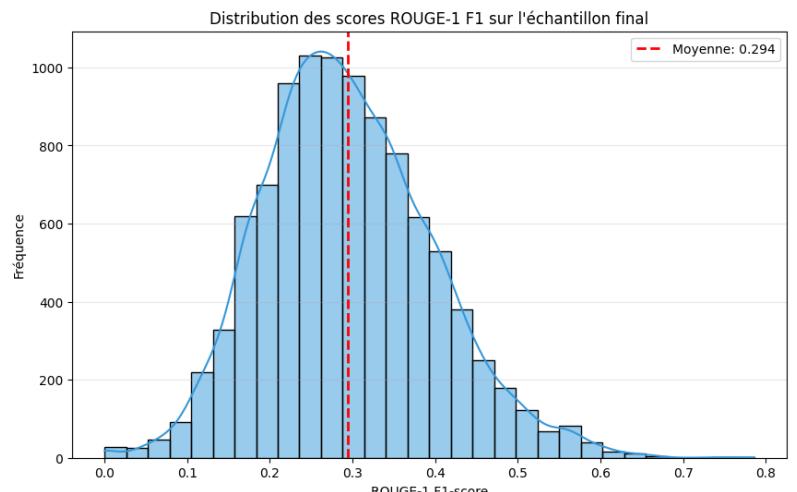
5.2. IT1- Phase 2 : Optimisation des hyper-paramètres (Gridsearch)

Les graphiques ci-dessous illustrent les résultats de la grille de recherche et montrent que pour cette tâche et ce modèle, des résumés plus courts (2 phrases) avec une inclusion large des relations de similarité entre les phrases sont les plus efficaces.

La carte de chaleur (heatmap) ci-dessous montre l'impact des hyperparamètres sur le ROUGE-1 F1-score :



Le modèle n'a pas les mêmes performances sur tous les exemples, mais il y a une concentration autour de la valeur moyenne.



Le scatterplot ci-contre est typique des résumés extractifs.

La plupart des points bleus (ROUGE-1) ont des valeurs de rappel plus élevées que de précision. Le modèle sélectionne des phrases entières de l'article original (bonne couverture des informations importantes, donc un bon rappel), mais ces phrases peuvent ne pas correspondre parfaitement aux choix de mots spécifiques du résumé de référence (ce qui peut réduire la précision).

Les points verts (ROUGE-2) sont nettement plus regroupés vers l'origine, indiquant des valeurs de

précision et de rappel beaucoup plus faibles. Contrairement à ROUGE-1, qui ne considère que des mots isolés, ROUGE-2 mesure deux mots consécutifs dans le même ordre, ce qui en fait une métrique plus stricte et plus difficile à satisfaire. En effet, un modèle extractif comme TextRank, a tendance à copier les phrases telles quelles depuis le texte source, sans reformulation. Alors que les résumés de référence proposent souvent des reformulations. Ce qui fait baisser le score ROUGE-2.

En général, un bon modèle de résumé chercherait à maximiser à la fois la précision et le rappel, et donc à déplacer ces nuages de points vers le coin supérieur droit du graphique (1,1). Les résultats actuels (nuage concentré dans le coin inférieur gauche) montrent donc les limitations inhérentes à un modèle extractif simple comme TextRank.

5.3. IT1 - Phase 3 validation finale Baseline extractive

Les scores ci-dessous obtenus après validation finale, constituent notre baseline pour la comparaison avec les approches abstractives qu'on verra dans l'itération 2.

SCORES Rouge IT1 Phase 1

Métrique	F1-score	Précision	Rappel
ROUGE-1	0.2941	0.2499	0.4009
ROUGE-2	0.0927	0.0788	0.1261
ROUGE-L	0.1885	0.1610	0.2566

6. Résultats Itération 2 : Abstractif T5-small

L'objectif de cette itération est de dépasser les performances de la baseline extractive et d'avoir si possible un ROUGE-1 > 0.35 (vs 0.2941 pour la baseline extractive).

6.1. Phase 1 Entrainement rapide du T5-small

Les courbes de perte ci-dessous montrent une diminution nette au cours de l'entraînement, passant d'environ 3.0 à 2.30, avec des valeurs finales identiques sur les ensembles d'entraînement et de validation.

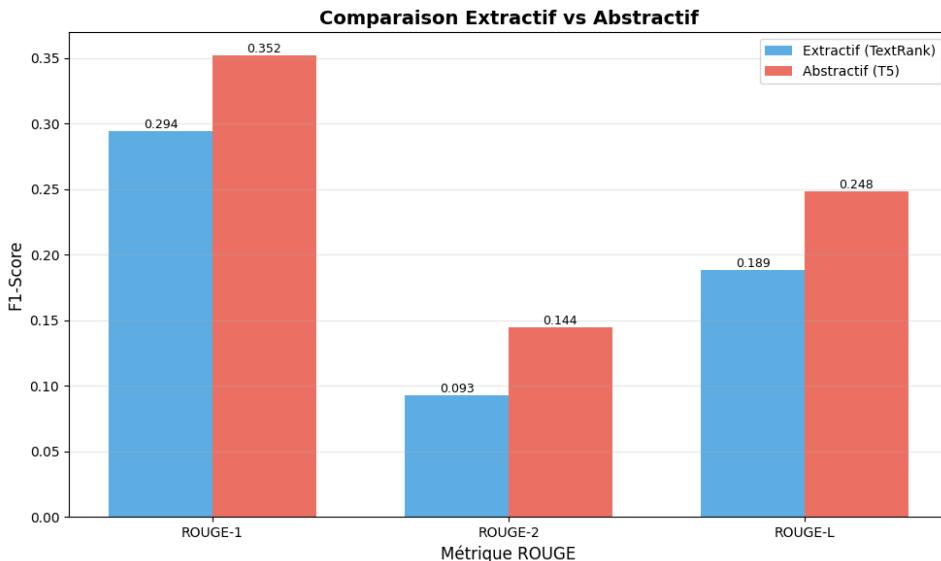
Cette convergence indique l'absence d'overfitting et suggère une bonne capacité de généralisation du modèle.



Bien que ce niveau de perte soit satisfaisant pour un nombre limité d'exemples (1000) et un faible nombre d'époques (3), le modèle n'a pas encore totalement convergé. Des meilleures performances sont attendues avec un entraînement plus long en phase 2.

SCORES Rouge IT2 Phase 1

Métrique	F1-score	Précision	Rappel
ROUGE-1	0.3519	0.3812	0.3407
ROUGE-2	0.1443	0.1562	0.1403
ROUGE-L	0.2483	0.2688	0.2408



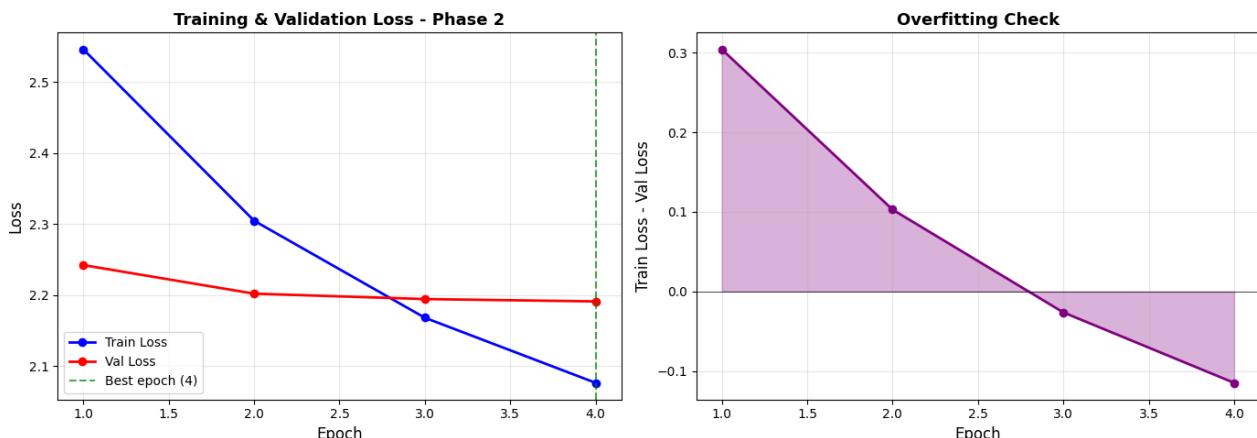
6.2. Phase 2 Entraînement complet T5-SMALL

Après un entraînement complet sur 10 000 exemples, le score ROUGE-1 est légèrement inférieur à celui de la phase 1 (0,3489 vs 0,3519) sur l'ensemble de test final.

Cette observation est classique en apprentissage automatique : un modèle optimisé sur un petit ensemble de validation peut montrer une légère baisse de performance sur un ensemble de test plus large et plus représentatif. Cela ne remet pas en cause la robustesse du modèle, mais souligne l'importance d'une évaluation sur des données variées et représentatives.

Les courbes d'apprentissage ci-dessous illustrent une convergence efficace du modèle T5-small sur 4 époques. La training loss décroît régulièrement de 2.6 à 2.08, indiquant un apprentissage progressif et stable. La validation loss, quant à elle, reste constante autour de 2.20, sans augmentation significative, ce qui confirme l'absence de surapprentissage (overfitting).

Le graphique de droite (Overfitting Check) visualise la différence entre la train loss et la val loss : l'aire sous la courbe devient légèrement négative en fin d'entraînement, ce qui signifie que le modèle généralise bien. Ce comportement est typique d'un modèle bien réglé et robuste sur des données inédites.



7. Analyse Comparée Extractif vs Abstractif

7.1. Performance des Modèles

Cette section compare les résultats obtenus par les deux approches — extractive (TextRank) et abstractive (T5-small fine-tuné) — sur le corpus de dépêches CNN.

Scores ROUGE sur le Test Set (5000 exemples pour T5, 10000 pour TextRank)

Métrique	Extractif (TextRank)	T5 (Phase 1)	T5 (Phase 2)	Gain relatif
ROUGE-1	0,2941	0,3519	0,3489	18,6 %
ROUGE-2	0,0927	0,1443	0,1368	47,6 %
ROUGE-L	0,1885	0,2483	0,2425	28,6 %

Le modèle T5 abstractif surpassé systématiquement la baseline extractive sur toutes les métriques, avec un gain particulièrement marqué sur ROUGE-2 (+47,6%). Cela indique une meilleure capacité à capturer des bigrammes et des séquences plus longues, proches des résumés de référence. Le gain sur ROUGE-L (+28,6%) confirme que le modèle abstractif produit des résumés plus fluides et cohérents, en tenant compte de la continuité narrative plutôt que de simplement juxtaposer des phrases extraites.

7.2. Comparaison avec les Standards du Domaine

Notre modèle T5-small se positionne dans la fourchette attendue pour un modèle léger fine-tuné sur un corpus réduit (ordre de grandeur ROUGE-1 $\approx 0.32\text{--}0.36$ rapportées pour T5-small dans la littérature, *Raffel et al., 2020*). Les résultats sont cohérents avec les contraintes du projet (ressources limité, corpus 16% du dataset complet) et valident l'efficacité du transfer learning pour le résumé abstractif.

Fourchettes par catégorie de modèles (littérature)

Catégorie	ROUGE-1	ROUGE-2	ROUGE-L	Notre position
Baselines extractives	0.30 - 0.35	0.12 - 0.15	0.25 - 0.30	TextRank: limite basse
Modèles légers (<100M)	0.32 - 0.36	0.13 - 0.17	0.28 - 0.32	T5-small: dans la fourchette
Modèles moyens (100-300M)	0.39 - 0.43	0.17 - 0.20	0.35 - 0.39	-
État de l'art (>300M)	0.44 - 0.47	0.21 - 0.24	0.40 - 0.44	-

Notre approche démontre qu'un modèle T5-small (60M paramètres) fine-tuné sur un corpus réduit (15K exemples, 16% du dataset) peut atteindre des performances respectables (ROUGE-1 = 0.349), se situant dans la fourchette attendue pour sa catégorie et surpassant significativement les approches extractives simples (+18.6%).

L'écart avec l'état de l'art illustre le compromis classique en ML entre performance et contraintes de ressources. Pour un projet d'études avec ressources limitées (GPU Colab gratuit), ces résultats valident l'efficacité du transfer learning.

7.3. Comparaison qualitative

Les exemples comparatifs ci-dessous illustrent les forces et limites respectives des approches extractive (TextRank) et abstractive (T5).

7.3.1.Exemples de résumé extractif (TextRank)

TextRank se montre efficace pour reproduire des citations et des éléments factuels (avec reprise des phrases clés), mais tend à omettre des informations stratégiques ou contextuelles.

#	Résumé de référence	Résumé généré (TextRank)	Observations qualitatives
2	(49 mots): <ol style="list-style-type: none"> 1. Rep. Steve Israel who chairs the Democratic Congressional Campaign Committee briefed reporters Wednesday . 2. `` Tough and unpredictable " and `` admittedly difficult " were two phrases used by Democrats . 3. Democrats ' objective is to minimize losses on their own side as oppose to picking up seats . 	(77 mots): <p>`` Tough and unpredictable " and `` admittedly difficult " were two phrases used by Rep Steve Israel , D-New York , the chairman of the Democratic Congressional Campaign Committee , House Democrats ' campaign arm used to describe the landscape to reporters on Capitol Hill on Wednesday . A spokeswoman for the House GOP 's campaign committee said Israel 's focus on outside factors ignored the drag that the president was having on congressional Democrats</p>	Reprend correctement certaines citations clés et le contexte immédiat, mais ne mentionne pas explicitement l'objectif stratégique des démocrates, pourtant central dans le résumé de référence. Le contenu est cohérent mais partiel, avec un accent mis sur les déclarations plutôt que sur les enjeux électoraux.
3	(43 mots): <ol style="list-style-type: none"> 1. NEW : Attack strikes people celebrating Iraq team 's win in World Cup qualifier 2. 29 wounded in predominantly Kurdish town in Diyala province 3. Twenty female suicide bombings in Iraq this year , a sharp uptick 4. Insurgent recruiters target troubled women , authorities say 	(57 mots): <p>BAGHDAD , Iraq -LRB- CNN -RRB-- A female suicide bomber in northern Iraq targeted buoyant soccer fans who had just watched their national team win a big match against China , authorities told CNN . Authorities said that al Qaeda in Iraq is recruiting women and that more women are offering themselves up for missions .</p>	Capture l'événement principal et le contexte de l'attaque, mais omet plusieurs informations comme le nombre de blessés et la localisation précise. Il met l'accent sur le recrutement de femmes par des groupes insurgés, ce qui modifie légèrement le focus par rapport au résumé de référence.

7.3.2. Exemples de résumé abstractif (T5)

À l'inverse, T5 produit des résumés plus fluides et synthétiques (ex. : exemple #3, correspondance quasi parfaite sur le conseil HIV/AIDS), mais peut dévier du focus original (ex. : exemple #2, accent mis sur la première du film plutôt que sur son statut culte actuel) ou simplifier excessivement (ex. : exemple #4, réduction de la diversité des points de vue)

#	Résumé de référence	Résumé généré (T5)	Observations qualitatives
2	2003 box-office bomb “The Room” is now a cult favorite. Tommy Wiseau was the director, producer and lead actor in the film. Film has regular screenings around the world.	The story of “The Room” started with its premiere in Los Angeles in 2003. There was some buzz about the film, helped by a billboard on a main drag in Hollywood. Some in the audience walked out of the showing.	Cohérent et fluide, mais met l'accent sur le contexte initial du film plutôt que sur son statut actuel de film culte. Plusieurs informations clés de la référence sont absentes.
3	Woman has headed nonprofit South Carolina HIV/AIDS Council since 1995. Since 2000, the council has provided free HIV tests to more than 8,600 people. One of the most effective tools is a mobile testing unit.	Bambi Gaddist's nonprofit has provided free HIV tests to more than 8,600 people since 2000. The mobile unit of the South Carolina HIV/AIDS Council brings confidential HIV testing and information to communities across the state.	Très forte correspondance factuelle et sémantique. Reprend fidèlement les informations essentielles et se rapproche fortement du résumé de référence.

8. Résultats Itération 3 : Ablation et sensibilité

8.1. Etude d'ablation

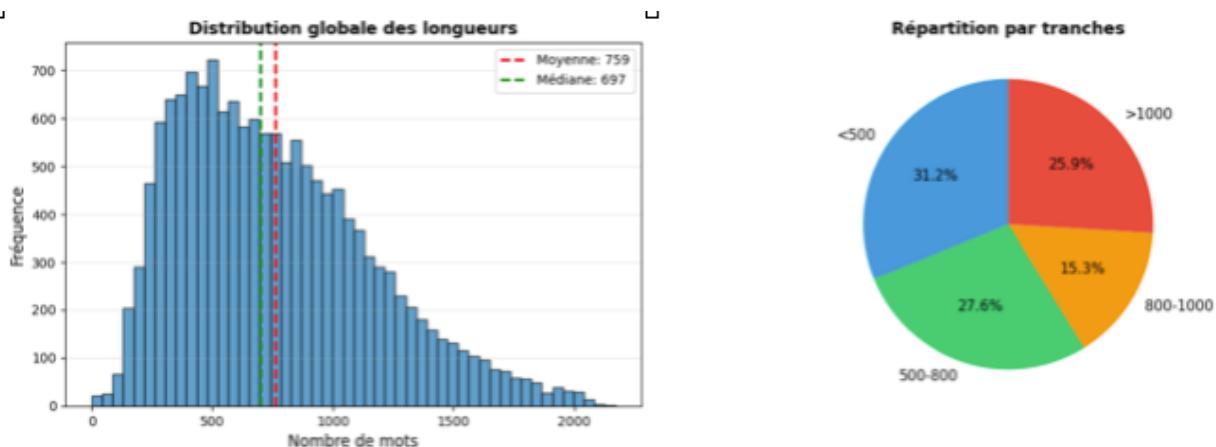
Pour valider notre configuration T5 optimale IT2 Phase 2 (512 tokens entrée, 128 sortie, beam=4, penalty=1.0), nous avons testé systématiquement chaque hyper-paramètre sur 1497 articles test.

Résultat principal : Aucune modification n'améliore significativement les performances (variations $<\pm 1.5\%$).

Résultats par type d'ablation sur approche abstractive T5 (Phase2)

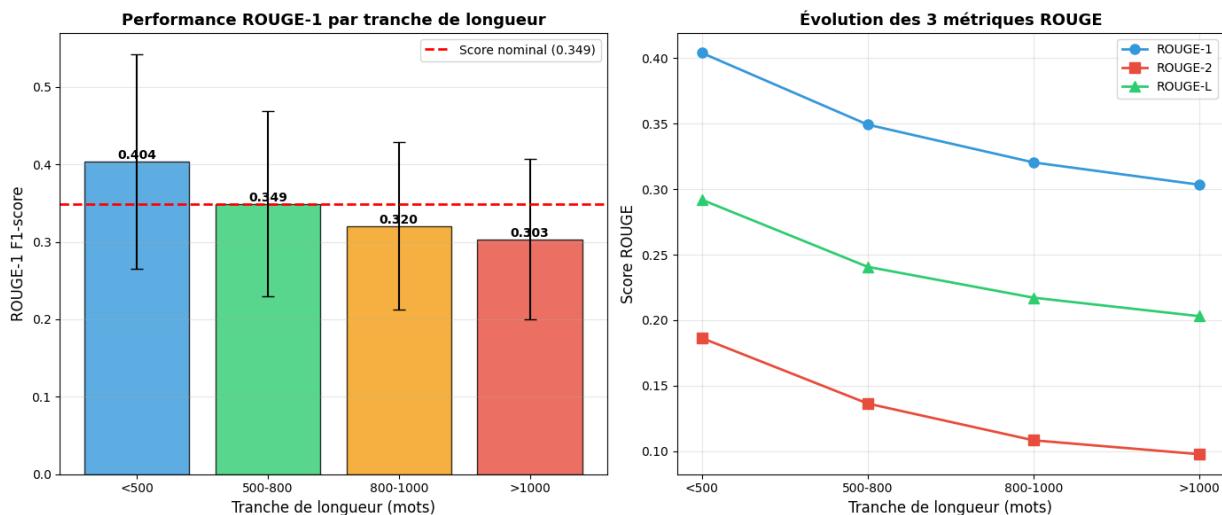
Hyperparamètre	Configurations testées	Meilleur résultat	Variation par rapport à la baseline
Source length	256, 384, 512 , 768	512 tokens	-0.8% (256), -0.2% (768)
Target length	64, 96, 128 , 192	128 tokens	-0.7% (64), $\pm 0\%$ (192)
Beam search	1, 2, 4 , 6, 8	2 beams	+1.0% R-1, -0.6% R-2
Length penalty	0.5, 1.0 , 1.5, 2.0	2.0	+1.1% (mais +15% longueur)
Preprocessing	naturel, SEP, numéros	naturel	$\pm 0\%$

8.2. Analyse de la distribution des longueurs d'articles



8.3. Sensibilité à la longueur des articles

La figure suivante montre les scores obtenus selon la longueur des articles en entrée. On observe que les articles de plus de 1000 mots (10% du corpus) subissent une dégradation de -13% (ROUGE-1 = 0.30 vs 0.34 nominal).



Scores ROUGE-1 du T5 (Phase 2) en fonction de la longueur des articles

Longueur (mots)	Part du corpus (%)	Nombre d'articles	ROUGE-1	Écart vs nominal
< 500	30.3	453	0.4040	+15.8 %
500 – 800	28.0	419	0.3493	+0.1 %
800 – 1 000	15.5	232	0.3204	-8.2 %
> 1 000	26.3	393	0.3034	-13.0 %

L'architecture Transformer de T5 impose une limite stricte de 512 tokens en entrée. Au-delà, le texte est tronqué automatiquement. Pour les articles de 1500 mots (~1875 tokens), cela signifie perdre environ 73% du contenu source, avec conservation des seuls premiers paragraphes. Cette limitation est inhérente aux transformers standards.

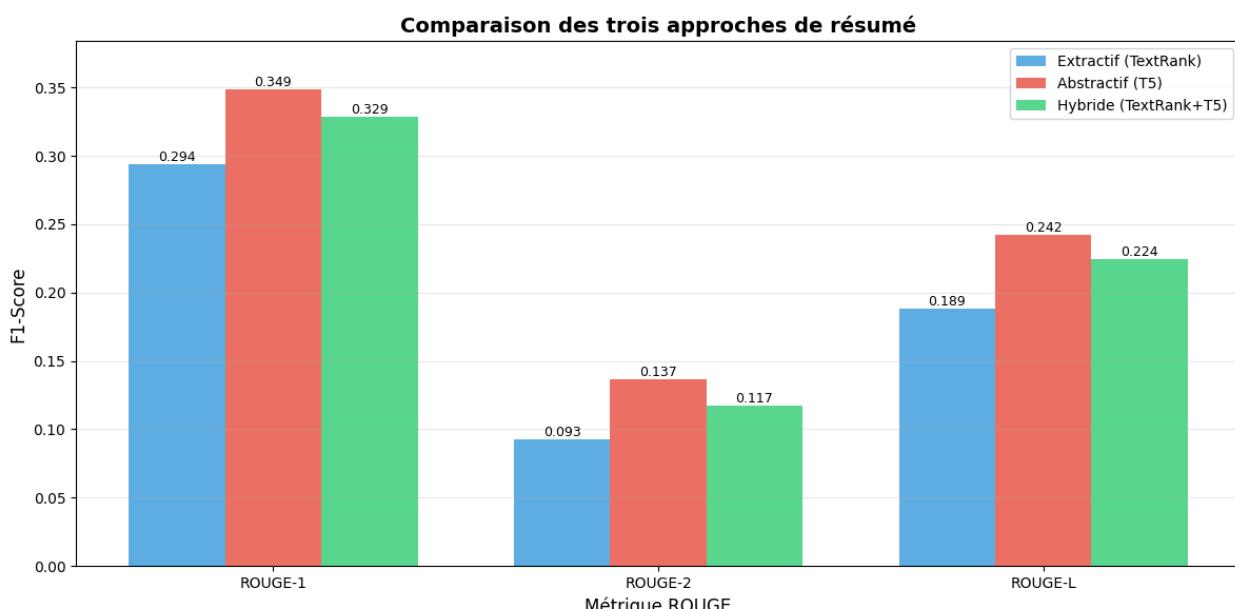
Les architectures récentes (Longformer, BigBird) lèvent cette contrainte mais nécessitent un réentraînement complet.

Solution étudiée en Itération 4 (Hybride) : Pour adresser cette limitation, nous avons développé une approche hybride en deux étapes : (1) Pré-extraction TextRank des phrases saillantes pour compression à ~300 mots, (2) Résumé abstractif T5 sur ce contenu filtré. Les résultats de cette approche sont présentés dans la section suivante.

9. Résultats Itération 4 - Hybride extractif+abstractif

9.1. Iteration 4 Phase 1: Hybride v1

L'approche hybride extractive+abstractive obtient des scores ROUGE intermédiaires entre les modèles extractifs et abstractifs (voir figure ci-dessous). Ce résultat s'explique par le fait que la phase extractive agit comme un goulot d'étranglement, limitant le contexte accessible au modèle génératif.



Cette contrainte permet néanmoins de contrôler la longueur d'entrée, de réduire le coût de calcul et d'améliorer la robustesse du système sur des documents longs. L'approche hybride serait peut être plus

pertinente dans un contexte constraint (documents très longs ou ressources limitées) que dans un cadre où le modèle abstractif peut exploiter un contexte suffisamment riche.

Tentative de diagnostic

L'approche hybride initiale (IT4.Phase1) a obtenu des résultats contre-intuitifs sur les articles longs, avec une dégradation de -17.9% contre -13% pour l'abstractif pur. Une analyse diagnostique a identifié deux causes principales :

- ✓ Problème 1 : La compression à 40% d'un article de 1309 mots produit 524 mots, dépassant encore la limite effective de T5 (~384 mots / 512 tokens). Le modèle subit donc une double troncature : compression TextRank puis troncature T5.
- ✓ Problème 2 : Sélection de phrases inadaptée. TextRank sur articles longs sélectionne des phrases dispersées, perdant la structure pyramide inversée typique des dépêches journalistiques.

9.2. Iteration 4 Phase 2 : Hybride variations v2, v3 et v4

Pour aller plus loin dans la compréhension des résultats précédents, nous avons testé nos hypothèses afin de vérifier notre tentative de diagnostic. Le tableau ci-dessous récapitule les résultats obtenus.

Scores ROUGE des différents modèles hybrides v1, v2, v3, v4

Approche	ROUGE-1	ROUGE-2	ROUGE-L	Écart vs v1	Écart vs abstractif
Abstractif pur	0.3034	—	—	—	Baseline
Hybride v1 (compression à 40 %, TextRank)	0.2872	—	—	Baseline	-5.3 %
Hybride v2 (compression supplémentaire à 28 %, TextRank)	0.2846	0.0867	0.1819	-0.9 %	-6.2 %
Hybride v3 (selection du début de l'article avec Lead-N, 350 mots)	0.2881	0.0873	0.1894	+0.3 %	-5.1 %
Hybride v4 (Lead-N + compression adaptative 25-28% selon longueur de l'article)	0.2823	0.0831	0.1847	-1.7 %	-7.0 %

Les approches hybrides évaluées n'apportent aucune amélioration par rapport au modèle abstractif pur. L'intégration de phrases extractives contraint la génération et réduit la capacité de reformulation du modèle, ce qui conduit à des performances globalement inférieures.

10. Résultats Itération 5 - Résumé hiérarchique

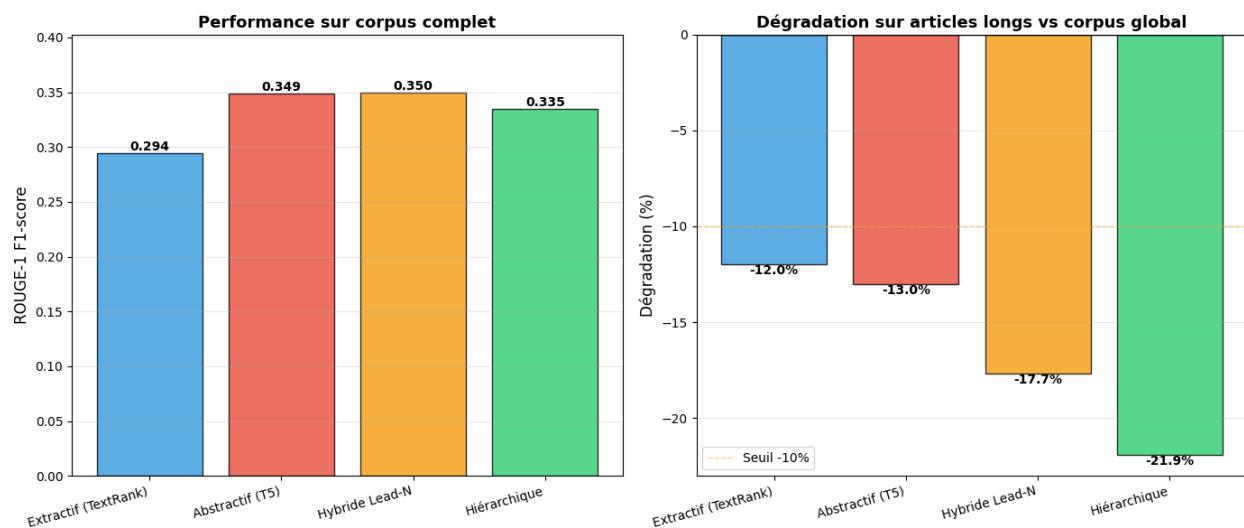
Suite aux résultats obtenus en itération 4, qui n'ont pas permis d'améliorer les scores de l'approche abstractive seule ni de l'approche hybride extractive–abstractive, nous avons introduit une nouvelle itération visant à mieux exploiter la structure du corpus, dans lequel chaque article est segmenté en une phrase par ligne.

Les itérations précédentes ont en effet confirmé que la troncature imposée par T5 (limite de 512 tokens) dégrade significativement les performances sur les articles longs (jusqu'à -13%). L'énoncé du projet suggérait par ailleurs d'exploiter cette segmentation naturelle du corpus.

Les résultats ci-dessous montrent que l'approche abstractif hiérarchique obtient, une fois encore, des performances inférieures à celles de l'approche abstractif directe (IT2), aussi bien sur le corpus complet que sur le sous-ensemble des articles longs.

Toutes les approches subissent une dégradation sur les articles longs, mais les méthodes hybrides et hiérarchiques amplifient la perte de performance, confirmant que la compression préalable agit fait perdre de l'information utile.

Approche	Corpus complet (ROUGE-1)	Articles longs (ROUGE-1)	Dégradation articles longs
IT1 – Extractif (TextRank)	0.2941	0.2589	-12.0 %
IT2 – Abstractif (T5)	0.3489	0.3034	-13.0 %
IT4 – Hybride Lead-N	0.3500	0.2881	-17.7 %
IT5 – Hiérarchique	0.3350	0.2615	-21.9 %



Par ailleurs, la comparaison avec l’itération 2 indique que la troncature imposée par T5 n’est pas aussi pénalisante que prévu dans ce cadre expérimental. Le modèle abstractif direct semble capable de sélectionner implicitement les informations les plus pertinentes, même lorsque le contexte d’entrée est partiellement tronqué.

L’approche hiérarchique ne constitue donc pas une solution efficace dans notre configuration expérimentale et met en évidence les limites des stratégies multi-passes lorsqu’elles reposent sur des résumés intermédiaires générés automatiquement.

11. Conclusions

Ce projet a permis de comparer des approches extractives, abstractives, hybrides et hiérarchiques pour la tâche de résumé automatique sur le corpus CNN. Les résultats confirment la supériorité des modèles abstractifs en termes de qualité mesurée par les métriques ROUGE.

L'étude d'ablation menée sur plusieurs paramètres du modèle abstractif n'a pas mis en évidence d'amélioration significative, indiquant que, dans ce cadre expérimental, le modèle est déjà correctement calibré et relativement robuste aux variations testées.

Parmi les variantes explorées, l'approche hybride de type Lead-N combinée à un modèle abstractif obtient des performances très légèrement supérieures sur le corpus complet. Cette amélioration s'explique par la structure des dépêches journalistiques, dans lesquelles les informations essentielles sont généralement concentrées en début d'article, rendant cette stratégie de sélection adaptée.

À l'inverse, les approches hybrides basées sur une sélection extractive non positionnelle, telles que TextRank, n'apportent pas de gain mesurable et peuvent même dégrader les performances en contraignant excessivement le contenu fourni au modèle génératif.

Les approches hiérarchiques évaluées n'ont pas non plus permis d'améliorer les résultats, suggérant que la réduction de la troncature ou la structuration en plusieurs niveaux ne se traduisent pas nécessairement par une meilleure qualité de résumé.

12. Limites et perspectives

L'évaluation reposant principalement sur les métriques ROUGE, l'analyse reste centrée sur le recouvrement lexical, sans exploration approfondie de la qualité sémantique ou de la cohérence des résumés, ce qui constitue une limite du protocole expérimental.

L'étude est conduite sur un seul corpus journalistique, CNN, dont le style rédactionnel et la structure peuvent favoriser certaines approches. Étendre l'analyse à d'autres types de documents permettrait d'évaluer la robustesse et la capacité de généralisation des méthodes étudiées.

Enfin, les contraintes de ressources de calcul ont limité l'exploration de modèles de plus grande taille ainsi que la réalisation d'analyses qualitatives à grande échelle.

Des perspectives futures incluent l'étude de modèles spécifiquement conçus pour les documents longs, l'intégration de mécanismes de contrôle de l'exactitude des informations, ainsi qu'une analyse plus approfondie des erreurs de génération, notamment dans les architectures multi-passes.

Une suite logique de ce travail consisterait par exemple à explorer d'autres modèles encoder-decoder dédiés à la génération de texte, tels que BART ou PEGASUS, afin d'évaluer leur capacité à améliorer la qualité des résumés.

Références

1. See, A., Liu, P. J., & Manning, C. D. (2017). *Get To The Point: Summarization with Pointer-Generator Networks*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017) (pp. 1073–1083).
<https://doi.org/10.18653/v1/P17-1099>
2. 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 (ICML 2020).
<https://arxiv.org/abs/1912.08777>
3. Raffel, C., et al. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. Journal of Machine Learning Research, 21(140), 1–67.
<https://arxiv.org/abs/1910.10683>
4. Lin, C. Y. (2004). *ROUGE: A Package for Automatic Evaluation of Summaries*. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (pp. 74–81).
<https://aclanthology.org/W04-1013.pdf>
5. Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Texts. Proceedings of EMNLP 2004, 404–411
<https://aclanthology.org/W04-3252.pdf>