Résumé interactif de documents

Raoufdine Said, Adrien Guille

Université de Lyon, Lyon 2, ERIC UR 3083
5 avenue Pierre Mendes France, 69676 Bron, France
contact : adrien.guille@univ-lyon2.fr

Résumé. Avec l'avénement des agents conversationnels modernes, le résumé automatique est devenu une pratique courante pour faciliter l'accès à l'information. Toutefois, les résumés abstractifs générés par ces outils peuvent être biaisés, non informatifs voire même mensongers. Ainsi, dans certains contextes sensibles (*e.g.* résumé d'articles scientifiques ou d'articles de presse), le résumé extractif reste une approche plus fiable. Dans cet article, nous présentons une méthode originale pour le résumé extractif de documents, couplée à une interface permettant aux utilisateurs de composer les résumés de manière interactive.

1 Introduction

Les agents conversationnels modernes, tels que ChatGPT [1], Bing Chat [2] ou Bard [3], ont rapidement été adoptés par un grand nombre d'utilisateurs, grâce notamment à leur apparente polyvalence et à leur interface conviviale. Parmi les nombreux usages possibles, un des plus fréquents est le résumé automatique (Azaria et al., 2023). Bien que les résumés générés puissent être pertinents, ces outils se montrent tout de même peu fiables pour cette tâche. En effet, ils peuvent générer des résumés tantôt inutiles – s'ils ne parviennent pas à suivre les instructions données par l'utilisateur (Ouyang et al., 2022), ou alors biaisés – dans la mesure où les modèles de langue sur lesquels ils reposent sont eux-mêmes biaisés (*e.g.* concernant le genre (Jentzsch et Turan, 2022) ou l'inclinaison politique (Rozado, 2023)), voire même parfois mensongers – à cause du phénomène d'hallucination dont souffrent les modèles de langue (Ji et al., 2023). Ceci est particulièrement problématique lorsqu'il s'agit de résumer des documents sensibles, par exemple des articles scientifiques, des articles encyclopédiques ou des articles de presse, où l'altération de l'information n'est pas tolérable. L'approche extractive est dans ces cas là une alternative viable, les résumés étant composés uniquement de phrases présentes dans les documents.

Dans cet article, nous proposons d'abord une nouvelle méthode pour le résumé extractif, basée sur un réseau de neurones profond de type encodeur-décodeur. Le code nécessaire à l'entraînement de ce réseau est disponible ici : `https://github.com/baragouine/radsum`. Ensuite, nous présentons une interface graphique permettant aux utilisateurs de résumer de manière interactive des documents à l'aide de cette méthode. Le code pour le dé-

1. `https://chat.openai.com/`
2. `https://www.bing.com/`
3. `https://bard.google.com/chat`

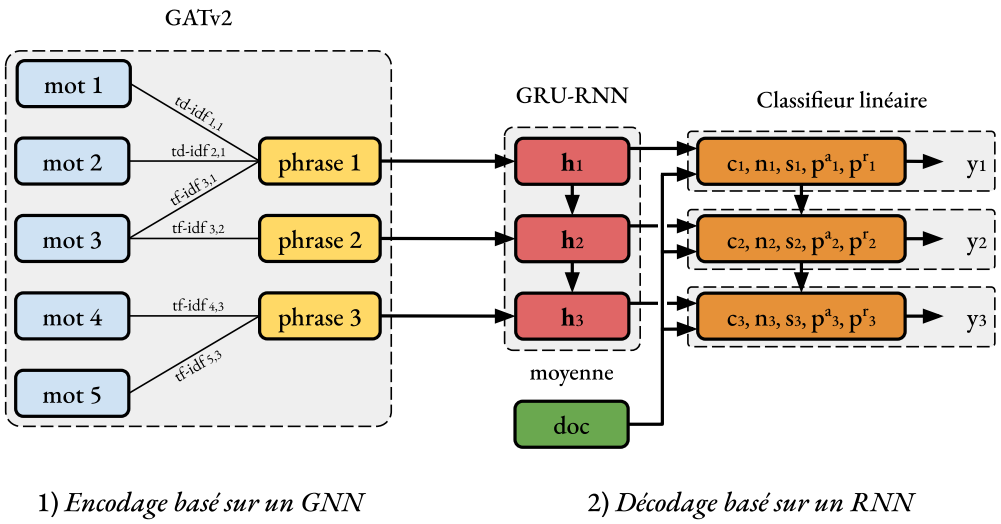

FIG. 1 – *Architecture générale de la méthode implémentée.*

ploiement de cette application est disponible ici : `https://github.com/baragouine/radsum_app/`. Par ailleurs, une vidéo illustrant l'utilisation de l'application est disponible à cette adresse : `https://youtu.be/vBenEaCIwkI`.

2 Méthode proposée

À l'image des méthodes récentes pour le résumé extractif, notre méthode repose sur un réseau de neurones profonds suivant une architecture encodeur-décodeur. Nous combinons deux méthodes existantes dans le but d'avoir à la fois un encodage expressif et un décodage permettant l'interaction avec l'utilisateur. Précisément, nous combinons (i) HeterSUMGraph (Wang et al., 2022) – que nous modifions légèrement et que nous n'utilisons que pour l'encodage du document et (ii) SummarRuNNer (Nallapati et al., 2017), que nous n'utilisons que pour le décodage du résumé. La figure 1 illustre l'architecture général du réseau de neurones implémenté.

2.1 Encodage du document

Nous convertissons le document à résumer en un graphe, suivant la démarche décrite par Wang et al. (2022). Il s'agit d'un graphe biparti comportant des sommets représentant les mots et des sommets représentant les phrases. Plus exactement, le graphe comporte un sommet par mot distinct dans le document et un sommet pour chaque phrase, connectés les uns avec les autres selon la composition des phrases. Les arêtes sont pondérées d'après les scores $tf \cdot idf$ mesurés au niveau des phrases : le terme tf correspond au nombre de fois qu'un mot apparaît dans une phrase, tandis que le terme idf est défini comme l'inverse du degré du sommet mot.

Des représentations vectorielles du sens des mots sont propagées à travers ce graphe par un GNN à deux couches, dans le but d'obtenir des représentations vectorielles des phrases. Alors qu'HeterSUMGraph utilise des couches GAT (Veličković et al., 2018), nous implémentons des couches GATv2 (Brody et al., 2022) plus expressives, avec un mécanisme d'attention prenant en compte la pondération des arêtes via des représentations vectorielles des pondérations $tf \cdot idf$ (discrétisées). Le poids d'attention entre deux sommets voisins v_i et v_j est mesuré à la couche ℓ (1 ou 2) selon la formule suivante :

$$\alpha_{ij}^{\ell} = \frac{\exp\left(\mathbf{w}^{\ell} \cdot \text{LeakyReLU}\left(\mathbf{W}_{\text{requête}}^{\ell}\mathbf{h}_i^{\ell-1} + \mathbf{W}_{\text{clé}}^{\ell}\mathbf{h}_j^{\ell-1} + \mathbf{W}_{\text{arête}}^{\ell}\mathbf{e}_{ij}\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathbf{w}^{\ell} \cdot \text{LeakyReLU}\left(\mathbf{W}_{\text{requête}}^{\ell}\mathbf{h}_i^{\ell-1} + \mathbf{W}_{\text{clé}}^{\ell}\mathbf{h}_k^{\ell-1} + \mathbf{W}_{\text{arête}}^{\ell}\mathbf{e}_{ik}\right)\right)}, \quad (1)$$

où $\mathcal{N}_i$ désigne les sommets voisins de v_i, plus le sommet v_i lui-même. Chaque couche calcule de nouvelles représentations en fonction des poids d'attention qu'elle calcule et des représentations reçues en entrée, comme suit :

$$\mathbf{h}_i^{\ell} = \sum_{j\in\mathcal{N}_i} \alpha_{ij}^{\ell}\mathbf{W}_{\text{clé}}^{\ell}\mathbf{h}_j^{\ell-1}. \quad (2)$$

Les vecteurs $\mathbf{h}^1$ et $\mathbf{h}^2$ désignent respectivement les représentations en sortie de la première et de la deuxième couche GATv2, alors que les vecteurs $\mathbf{h}^0$ désignent les représentations initiales (pré-entraînées pour les mots et aléatoires pour les phrases).

2.2 Décodage du résumé

Nous ne conservons que les représentations des phrases issues de la deuxième couche GATv2, les vecteurs $\{\mathbf{h}_1^2, \mathbf{h}_2^2, \ldots, \mathbf{h}_n^2\}$ (avec n le nombre de phrases), et les passons à un RNN basé sur une cellule GRU (Cho et al., 2014) pour les contextualiser selon leur position dans le document. Les nouvelles représentations ainsi obtenues (les états cachés du RNN), que nous notons par la suite $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$, sont traitées séquentiellement, dans l'ordre du document, comme le proposent Nallapati et al. (2017). La décision de conserver ou pas la i-ème phrase dans le résumé est prise linéairement en 5 scores :

- **score de contenu** (*i.e. content* dans l'interface) : fonction linéaire de la représentation de cette phrase, $\mathbf{W}_c\mathbf{h}_i$;
- **score de saillance** (*i.e. salience*) : fonction bilinéaire de la représentation de cette phrase et de la représentation du document entier, $\mathbf{h}_i^{\top}\mathbf{W}_s\mathbf{d}$;
- **score de nouveauté** (*i.e. novelty*) : fonction bilinéaire de la représentation de cette phrase et de la représentation du document jusqu'à la phrase $i-1$, $\mathbf{h}_i^{\top}\mathbf{W}_n \tanh(\mathbf{s}_i)$;
- **score de position absolue** (*i.e. absolute position*) : fonction linéaire de la représentation de la position absolue de cette phrase, $\mathbf{W}_{ap}\mathbf{p}_i^a$;
- **score de position relative** (*i.e. relative position*) : fonction linéaire de la représentation de la position relative de cette phrase, $\mathbf{W}_{rp}\mathbf{p}_i^r$.

La probabilité de conserver la i-ème phrase est calculée comme suit :

$$\text{p}(y_i = 1|\mathbf{h}_i, \mathbf{s}_i, \mathbf{d}) = \sigma\left(\mathbf{W}_c\mathbf{h}_i + \mathbf{h}_i^{\top}\mathbf{W}_s\mathbf{d} + \mathbf{h}_i^{\top}\mathbf{W}_n \tanh(\mathbf{s}_i) + \mathbf{W}_{ap}\mathbf{p}_i^a + \mathbf{W}_{rp}\mathbf{p}_i^r\right), \quad (3)$$

avec σ la fonction sigmoïde ; $\mathbf{d}$ la représentation du document entier, obtenue en moyennant tous les états cachés $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$; $\mathbf{s}_i$ la représentation partielle du document, jusqu'à la phrase $i-1$, calculée comme une moyenne pondérée des états cachés $\{\mathbf{h}_1, \ldots, \mathbf{h}_{i-1}\}$:

$$\mathbf{s}_i = \sum_{j=1}^{i-1} \mathbf{h}_j \mathrm{p}(y_j = 1 | \mathbf{h}_j, \mathbf{s}_j, \mathbf{d}). \tag{4}$$

3 Mise en œuvre et interface utilisateur

3.1 Entraînement et évaluation

Pour les besoins de cette démonstration, nous avons entraîné notre méthode – RadSum, sur le corpus NYT50, composé d'articles tirés du New York Times (Durrett et al., 2016). Nous listons dans la table 1 les scores ROUGE-1 et ROUGE-2 mesurés en comparant les résumés extraits automatiquement par RadSum avec les résumés de références. Nous y listons également les scores obtenus par SummarRuNNer et HeterSUMGraph, mettant en lumière la performance accrue de la méthode RadSum vis-à-vis des méthodes desquelles elle s'inspire.

3.2 Interface par défaut

La figure 2 montre l'interface utilisateur par défaut de l'application RadSum. Le côté gauche de la fenêtre permet de saisir le document à traiter et de spécifier la configuration générale (délimitation des phrases, longueur du résumé en caractères ou en phrases). Le côté droit affiche le résumé extrait de ce document. Pour chaque phrase retenue pour former le résumé, un diagramme en barres décrit la contribution de chacun des 5 scores et la phrase est colorée selon le score dominant (les scores étant ré-échelonnés dans l'intervalle [0 ;1] par la fonction sigmoïde pour en facilité la lecture). Dans cette capture d'écran, le document saisi correspond à un article à propos de la remise du prix Nobel de physique en octobre 2023 et le résumé extrait comporte 4 phrases, mettant l'accent sur les lauréats et les implications de leur découverte. On note que la première phrase du document a été retenue dans le résumé principalement en raison de sa position, tandis que les phrases 4 à 6 ont été choisies en raison des scores de saillance et de nouveauté.

3.3 Exemple d'interaction

La figure 3 montre comment l'utilisateur peut interagir avec la méthode via l'interface pour personnaliser le résumé. Dans cet exemple, à l'aide du bouton "filter", l'utilisateur a décidé d'ignorer tous les scores à l'exception du score de saillance, dont dépend donc exclusivement la probabilité d'appartenance au résumé. Cela a pour effet de produire un résumé différent, composé des phrases 19, 28, 29 et 31, se concentrant plutôt sur la découverte en elle-même. On voit aussi que l'utilisateur a manuellement retiré la phrase 31 du résumé, ce qui a entraîné l'ajout automatique de la prochaine phrase la plus probable, ici la 30ème, pour maintenir une longueur de 4 phrases.

	ROUGE-1	ROUGE-L
SummaRuNNer	45.3	34.65
HeterSUMGraph	46.76 +2.0%	35.21 +1.6%
RadSum	46.91 +2.4%	35.35 +2.0%

TAB. 1 – *Scores ROUGE sur le corpus NYT50. Le gain par rapport à SummaRuNNer est donné à droite dans chaque colonne.*

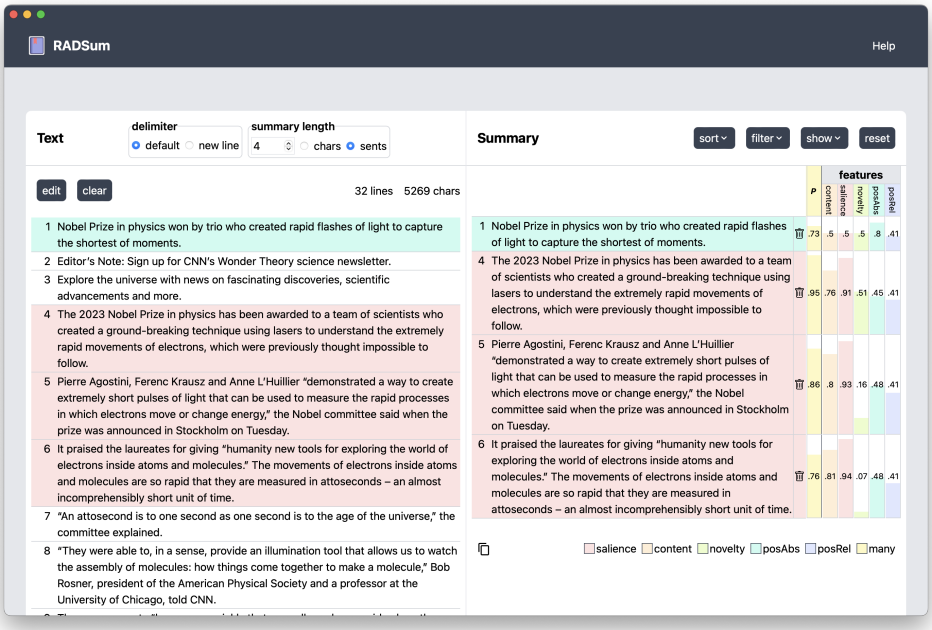

FIG. 2 – *Interface par défaut.*

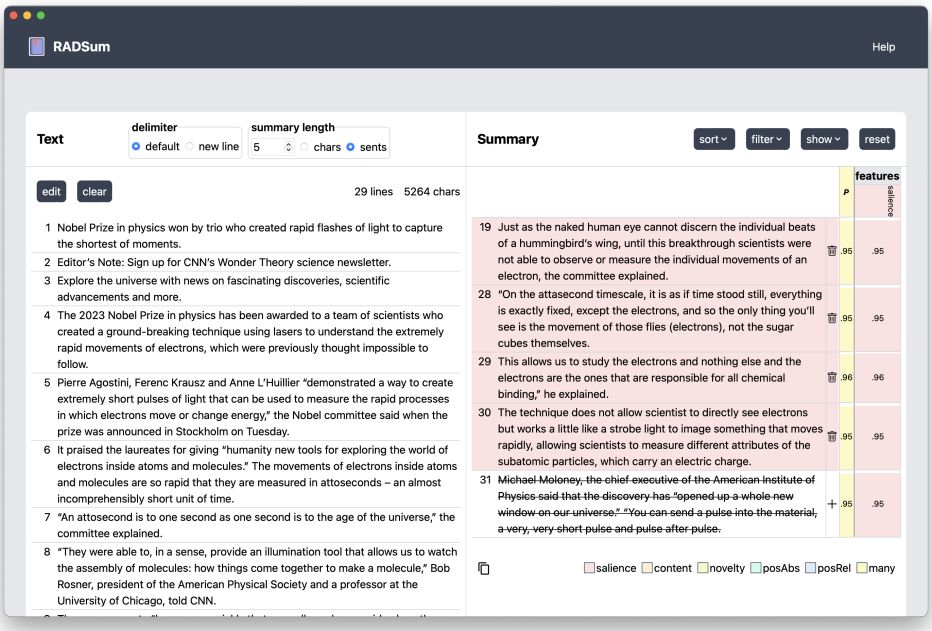

FIG. 3 – *Exemple d'interaction : l'utilisateur a sélectionné la saillance comme seul score et a retiré la phrase 31 du résumé.*

Références

Azaria, A., R. Azoulay, et S. Reches (2023). Chatgpt is a remarkable tool – for experts. *ArXiV Technical Report*.

Brody, S., U. Alon, et E. Yahav (2022). How attentive are graph attention networks ? ICLR.

Cho, K., B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. EMNLP.

Durrett, G., T. Berg-Kirkpatrick, et D. Klein (2016). Learning-based single-document summarization with compression and anaphoricity constraints. ACL.

Jentzsch, S. et C. Turan (2022). Gender bias in BERT - measuring and analysing biases through sentiment rating in a realistic downstream classification task. GeBNLP workshop @ ACL.

Ji, Z., N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, et P. Fung (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys 55*(12).

Nallapati, R., F. Zhai, et B. Zhou (2017). Summarunner : A recurrent neural network based sequence model for extractive summarization of documents. AAAI.

Ouyang, L., J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, et R. Lowe (2022). Training language models to follow instructions with human feedback. *ArXiV Technical Report*.

Rozado, D. (2023). The political biases of chatgpt. *Social Sciences 12*(3).

Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò, et Y. Bengio (2018). Graph Attention Networks. ICLR.

Wang, D., P. Liu, Y. Zheng, X. Qiu, et X. Huang (2022). Heterogeneous graph neural networks for extractive document summarization. ACL.

Summary

With the advent of modern chatbots, automatic summarization is becoming common practice to quicken access to information. However the summaries they generate can be biased, unhelpful or even untruthful. Hence, in sensitive scenarios (*e.g.* summarizing scientific articles or press articles), extractive summarization remains a more reliable approach. In this paper we present an original extractive method coupled with a user-friendly interface that allows for interactive summarization.