Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles
Avignon, France, 27 juin au 1er juillet 2022

RésumeSVD : Un outil efficace et performant pour le résumé de texte non supervisé

Gabriel Shenouda[1] Christophe Rodrigues[1] Aurélien Bossard[2]
(1) Léonard De Vinci Pôle Universitaire, Research Center, 92 916 Paris La Défense, France
(2) Laboratoire d'Informatique Avancée de Saint-Denis, Université Paris 8 (EA4383), 93200 Saint-Denis, France
gabriel.shenouda@edu.devinci.fr, christophe.rodrigues@devinci.fr,
aurelien.bossard@iut.univ-paris8.fr

RÉSUMÉ

Cet article présente une nouvelle méthode, RésumeSVD, pour le résumé automatique extractif non supervisé. Cette méthode est fondée sur la décomposition en valeurs singulières afin de réduire la dimensionnalité des plongements de mots et de proposer une représentation de ces derniers sur un petit nombre de dimensions, chacune représentant un sujet latent. En effet, dans un contexte spécifique et restreint, de multiples dimensions des plongements de mots deviennent moins pertinentes puisqu'apprises dans des contextes plus larges. Elle utilise également le regroupement automatique de mots pour réduire la taille du vocabulaire, et est suivie d'une heuristique d'extraction de phrases. La méthode surpasse en efficacité les approches extractives les plus récentes tout en étant plus efficiente. De plus, RésumeSVD nécessite peu de ressources, en termes de données et de puissance de calcul. Elle peut donc être exécutée sur de longs documents, tels que des articles scientifiques, ainsi que sur de grands corpus à documents multiples. Notre méthode est suffisamment rapide pour être utilisée dans des systèmes de résumé en direct. Nous partageons publiquement le code source de notre approche permettant de reproduire tous nos résultats.

ABSTRACT

RésumeSVD : An efficient and effective tool for unsupervised text summarization

This paper introduces a new method, RésumeSVD, for automatic unsupervised extractive summarization. It is based on singular value decomposition in order to reduce the dimensionality of word embeddings and project words on hidden topics. Indeed, some word embedding dimensions can become less relevant in a specific and restreint context because they were learned in a larger context. Word clustering is also used to reduce the vocabulary size. It is followed by sentence extraction heuristics. We evaluate RésumeSVD using several corpora of different nature (news, scientific articles, social network). The method outperforms in effectiveness most recent extractive approaches. Moreover, RésumeSVD is not greedy, in terms of data and computing power. So it can be used on long single documents such as scientific papers as well as large multi-document corpora and is fast enough to be used in live summarization systems. The source code is publicly available allowing to reproduce all our results.

MOTS-CLÉS : résumé automatique de documents, plongement de mots, réduction de dimension.

KEYWORDS: automatic Summarization, word embedding, dimension reduction.

1 Introduction

La recherche sur le résumé automatique s'est récemment concentrée sur les approches supervisées. Depuis *Pointer Generator* (See *et al.*, 2017), des progrès considérables ont été réalisés dans le domaine du résumé génératif supervisé. Cependant, ces approches nécessitent des corpus d'apprentissage substantiels composés d'une grande quantité de paires de documents et résumés, et malgré les avancées récentes sur le réglage fin et l'apprentissage par transfert, elles sont limitées à des domaines spécifiques. La recherche sur les méthodes de résumé non supervisées ne peut donc pas être laissée de côté. Dans cet article, nous abordons le résumé extractif non supervisé, qui vise à sélectionner des phrases dans un ou plusieurs documents et à les assembler afin de construire un résumé. Cette sélection se fait souvent selon un compromis entre deux critères : la centralité d'une phrase au sein du ou des documents à résumer (c'est-à-dire à quel point le résumé restitue les informations centrales), et la diversité (la couverture par le résumé des différentes informations).

Inspirés par les travaux de Gong *et al.* (2018) sur le calcul de la similarité des textes longs, nous supposons que des thèmes cachés spécifiques à un texte peuvent émerger à partir des vecteurs de mots, calculés à partir d'un corpus général. Chaque sujet représente un aspect particulier de la sémantique du texte. Ces thèmes cachés permettent d'éliminer les informations inutiles des représentations de mots et peuvent être considérés comme une nouvelle représentation du texte. Les mots peuvent être mis en correspondance avec un sujet latent et de cette façon, nous pouvons dériver des scores de centralité de mots à partir d'un texte représenté à l'origine comme une matrice de vecteurs de mots. Compte tenu de ces scores de mots, des heuristiques d'extraction de phrases peuvent être appliquées pour générer un résumé extractif.

Nous proposons une nouvelle méthode de résumé extractif non supervisé, appelée RésumeSVD que nous évaluons sur des corpus de résumé hétérogènes afin de tester sa généralisation.

2 Travaux connexes

L'une des bases les plus connues du résumé automatique est probablement TextRank (Mihalcea & Tarau, 2004), une méthode qui extrait les phrases en fonction de leur centralité dans une représentation graphique du document.

À notre connaissance, Padmakumar & He (2021) est le plus récent résumeur extractif non supervisé. Dans une étude empirique, il surpasse les approches de pointe sur différents types de textes (actualités, médecine, réseaux sociaux). Le modèle est similaire au modèle de vraisemblance de requête décrit dans Manning *et al.* (2008) pour la recherche d'information où un modèle de langage est utilisé pour estimer la probabilité d'un document étant donné une requête. Ici, la requête est remplacée par une phrase candidate à l'extraction dans le résumé, et les phrases sont extraites au cours d'un processus glouton en fonction de l'estimation de probabilité du modèle de langage.

SummPip (Zhao *et al.*, 2020) est une méthode de résumé multi-documents non supervisée fondée sur la compression de graphes. Elle convertit les documents en un graphe de phrases où les nœuds sont les phrases, et les arêtes sont construites sur la base d'indices de surface, d'informations sémantiques exogènes et de similarités sémantiques. Afin d'obtenir un résumé de k phrases, une matrice laplacienne est créée sur la base de la représentation graphique des phrases de leur document, et les k premiers vecteurs propres de cette matrice sont calculés afin de vectoriser les phrases. Un *k-means* est utilisé

pour séparer ces phrases en k clusters avant d'utiliser une version de l'algorithme du plus court chemin pour sélectionner les phrases finales utilisées pour générer le résumé de sortie. Pour l'intégration, un modèle Word2Vec (Mikolov *et al.*, 2013) affiné sur chaque jeu de données est utilisé.

La décomposition en valeurs singulières (SVD) des textes a été utilisée à l'origine pour la comparaison de documents dans la technique d'analyse sémantique latente (LSA) introduite par Deerwester *et al.* (1990). Les documents sont représentés par une matrice document-terme remplie des occurrences des termes dans les documents, un terme par ligne et un document par colonne. La méthode SVD est donc utilisée pour réduire le nombre de termes tout en préservant la similarité entre les documents. Gong & Liu (2001) ont été les premiers à utiliser LSA pour le résumé automatique. LSA permet de détecter les thèmes principaux, puis les phrases les plus proches des thèmes sont extraites pour constituer un résumé. La méthode a été améliorée par Steinberger & Jezek (2004) en pondérant la probabilité de sélection des phrases par l'importance des sujets (proportionnellement à leur variance explicative). L'analyse latente Dirichlet a également été utilisée pour le résumé automatique (Blei *et al.*, 2003) : les résumés sont construits en maximisant la couverture des thèmes identifiés par l'analyse latente. Cependant, il a été montré dans Nyzam *et al.* (2018) que cette méthode était moins performante que les méthodes à base de graphes ou de centroïdes (Radev *et al.*, 2000).

Les méthodes de plongement de mots ont permis d'améliorer considérablement les méthodes de résumé supervisées, néanmoins ces modèles nécessitent des corpus alignés massifs et des ressources computationnelles conséquentes. Leur usage efficace dans un cadre non-supervisé présente des défis que nous nous proposons de relever au travers d'une méthode qui vise d'abord à réduire la dimensionnalité de ces modèles avant de les utiliser efficacement pour du résumé automatique.

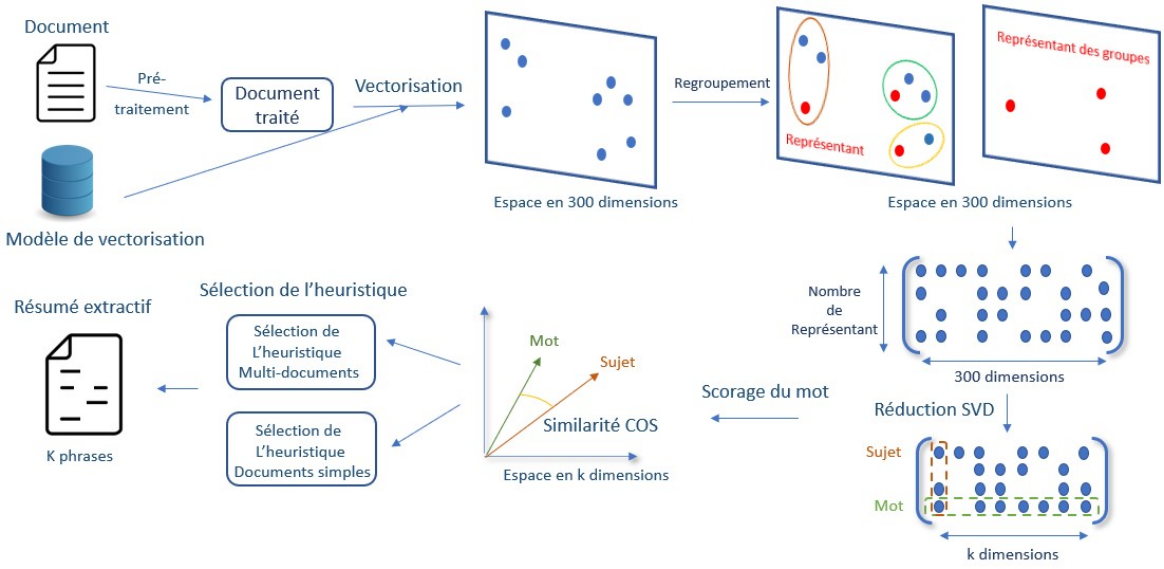

FIGURE 1 – Schéma global de la méthode RésumeSVD pour obtenir un résumé à partir d'un texte.

3 Notre méthode : RésumeSVD [1]

Modèle proposé : Les plongements de mots sont devenus incontournables pour la représentation sémantique. Cependant, dans un contexte restreint, par exemple un document ou plusieurs documents sur un même sujet, l'information sémantique pertinente peut être contenu sur un sous-ensemble restreint de dimensions. Même le calcul de la similarité sémantique entre deux mots d'après leur représentation vectorielle reste un défi (Farouk, 2018). Nous proposons d'adapter des méthodes de regroupement non supervisées afin d'exploiter ces vecteurs denses et d'identifier les phrases les plus centrales des textes. Nous pouvons représenter les textes dans une matrice où une ligne représente un mot et une colonne représente une dimension de vectorisation : $Matrice = \#Mot \times \#Dimension$. Comme un résumé peut être interprété comme une compression d'un texte, nous compressons cette matrice. Nous décrivons un processus en deux étapes où nous pouvons d'abord réduire le nombre de mots (lignes) par une méthode de regroupement, puis le nombre de dimensions (colonnes) par une décomposition en valeurs singulières. Une vue d'ensemble du modèle est donnée à la Figure 1.

Regroupement des mots : Afin de réduire le nombre de mots, et donc de vecteurs, nous utilisons une méthode de regroupement vectorielle non supervisée. De cette façon, les vecteurs les plus proches supposés partager les mêmes contextes seront regroupés. En fonction de la méthode de regroupement, il est possible de contrôler le nombre de groupes. Ainsi, plus le nombre de groupes est faible, plus le taux de compression est élevé. Au sein d'un même groupe les vecteurs des mots sont substitués par un vecteur unique, représentant l'ensemble. Ce vecteur est calculé comme le vecteur moyen de tous les vecteurs d'un groupe.

Décomposition : La décomposition en valeurs singulières (SVD) d'une matrice permet notamment de la compresser en réduisant son nombre de colonnes. Nous proposons d'utiliser la SVD pour réduire le nombre de dimensions des représentations vectorielles des mots. En effet, puisque les vecteurs ont une grande dimension (300 dans nos expériences), la SVD a la capacité d'identifier les dimensions portant la plupart des informations, ce qui nous permet de conserver les plus importantes. Comme dans LSA (Deerwester *et al.*, 1990), nous nommons les vecteurs propres comme des sujets.

Score des mots : Le score d'un mot sur un sujet donné (trouvé par la SVD) est défini par :

$$WordScore(w, t_i) = \frac{\overrightarrow{w} \cdot \overrightarrow{t_i}}{\|\overrightarrow{w}\|}$$

Où $\overrightarrow{w}$ est le vecteur d'intégration du mot w et t_i est un sujet trouvé par la SVD. Le score est une similarité cosinus entre le vecteur représentant le mot et un sujet. Intuitivement, plus un mot est proche d'un thème, plus il explique la variation de cet axe, donc plus il contient d'informations et devrait être sélectionné pour faire partie du résumé.

Extraction des phrases : Afin d'extraire les meilleures phrases de la matrice réduite obtenue par regroupement et décomposition, nous utilisons deux heuristiques : une pour le résumé mono et une pour le multi-document. Dans le cas mono-document, le score d'une phrase est défini par la somme des similarités cosinus entre les mots qui la composent et un axe sélectionné (selon le $WordScore$). Les k meilleures phrases constituent alors le résumé. En multi-document, plusieurs sujets différents pouvant être traités, les k premiers axes sont pris en compte. La meilleure phrase de chaque axe est sélectionnée (avec la même méthode que pour le cas mono-document) pour constituer le résumé.

1. https://github.com/anonymeTAL/ResumeSVD (anonymisée pour la soumission)

4 Expériences

Nous effectuons l'évaluation de notre méthode sur des corpus hétérogènes : par leur fonction (mono ou multi-document), le type des documents source (presse, réseaux sociaux, articles scientifiques) et la taille des documents et résumés. Ces corpus et leurs caractéristiques sont présentés en tableau 1. Tous les corpus présentés sont disponibles dans la plateforme *open source* tensorflow [2], à l'exception de DUC2004 qui est disponible sur demande auprès du NIST. Afin de produire une comparaison signifiante de notre méthode, nous comparons notre modèle à différentes approches de l'état de l'art :

TextRank : nous implémentons TextRank (Mihalcea & Tarau, 2004) largement répandue dans le domaine du résumé de texte. Cette méthode, décrite dans la section 2, est à ce jour, l'une des méthodes non supervisées les plus rapides pour produire des résumés.
LSA : nous utilisons LSA (Steinberger & Jezek, 2004), une méthode fondée sur la SVD comme décrit dans la section 2. Elle permet de mettre en évidence les avantages de notre approche.
BERT SVD : nous implémentons une approche complètement nouvelle fondée sur la vectorisation des phrases par BERT, suivie d'un processus de sélection similaire à notre approche principale RésumeSVD. L'étape finale de sélection des phrases est également directe, les phrases les plus proches des sujets sont considérées comme les meilleures et constituent donc notre résumé.
PMI : nous reproduisons (Padmakumar & He, 2021) en utilisant l'implémentation donnée par les auteurs. Notre exécution ne concerne que les ensembles de données de résumé de document simple, car PMI est une méthode de résumé de document simple.
SummPip : nous exécutons l'implémentation de SummPip publiée par Zhao *et al.* (2020). SummPip étant conçu pour le résumé multi-documents, nous l'exécutons uniquement sur ce type de données.
Supervisé : le modèle MatchSum (Zhong *et al.*, 2020). Il s'agit d'une approche extractive supervisée par apprentissage profond. C'est à notre connaissance une des meilleures approches existantes.

Afin de garder la méthode légère et réellement non supervisée, nous avons décidé empiriquement d'utiliser une méthode générique de vectorisation de mots : GloVe(300d) (Pennington *et al.*, 2014).

Nous avons testé trois méthodes de clustering : OPTICS (Ankerst *et al.*, 1999) ; une version améliorée de DBSCAN (Ester *et al.*, 1996), l'algorithme des K-moyennes (Forgy, 1965), et *Agglomerative Clustering*, toutes trois dans leur implémentation de la bibliothèque scikit-learn (Pedregosa *et al.*, 2011). L'utilisation d'*Agglomerative Clustering* induit une légère perte de score ROUGE, de l'ordre de 0.5% à 1.3% par rapport aux k-moyennes et de l'ordre de 1% à 1.9% par rapport à OPTICS, mais permet des gains de vitesse d'exécution respectivement de 40% à 700% et de 1100% à 2300% selon les corpus. L'algorithme *Agglomerative Clustering* est donc un bon compromis entre qualité et temps d'exécution, un aspect important pour le passage à l'échelle permis par la linéarité de notre méthode.

2. https://www.tensorflow.org/datasets/catalog/overview

Nom	doc nature	type	#docs	phrases/doc	mots/doc	phrases/résumé	mots/résumé	compression
CNN/DM	Art. presse	mono	11489	26.9	766.6	3.9	58.2	7.6%
XSum	Art. presse	mono	11331	23.2	424.9	1	18.6	4.4%
PubMed	Art. scient.	mono	6658	101.6	3142.9	7.6	208	6.6%
Reddit-Tifu	Rés. soc.	mono	42139	19.7	439.2	1.4	23.5	5.4%
Reddit	Rés. soc.	mono	48	12.1	234.5	1.2	25.2	10.7%
Multi-News	Art. presse	multi	5622	17.5	491	9.8	262.0	53.4%
DUC2004	Art. presse	multi	50	264.9	6583.14	31.12	422.26	6.4%

TABLE 1 – Caractéristiques des corpus utilisés dans l'évaluation.

Pour fixer les différents paramètres de la méthode, nous utilisons une grille de recherche automatique sur moins de 0.4% de chaque jeu de données. Le pourcentage de vocabulaire à conserver est défini en utilisant l'algorithme de regroupement entre 100% et 10% par pas de 10. Enfin, l'heuristique du sujet de notation (simple ou multi-documents) est sélectionnée en fonction des scores F1 de ROUGE.

Les caractéristiques de la machine que nous utilisons pour faire nos expériences sont les suivantes : Processeur AMD 3700X 8 coeurs, 64 GB de RAM, et 2 RTX 2080TI de 11 GB de mémoire chacune.

5 Résultats

Afin d'évaluer notre méthode, nous utilisons la mesure F1 ROUGE-2 (Lin, 2004) qui montre de fortes corrélations avec les jugements humains (Graham, 2015).

	Mono-document					Multi-document	
	CNN/DM	XSum	Reddit-Tifu	Reddit	PubMed	Multi-News	DUC2004
Supervisé	20.86	04.66	06.17	-	14.91	16.51	-
TextRank	13.90	**03.15**	**03.51**	08.64	13.60	10.86	08.32
LSA	10.47	02.60	01.94	07.74	00.90	09.22	08.09
PMI	15.49	02.89	02.38	08.51	10.85	-	-
SummPip	-	-	-	-	-	13.28	08.47
BERT SVD	7.60	02.44	02.36	05.60	09.43	13.42	03.76
RésumeSVD	**17.70**	02.77	03.33	**09.28**	**14.50**	**15.83**	**10.15**

TABLE 2 – Résultats F1 ROUGE-2 de chacunes des approches sur les différents corpus. La meilleure méthode non supervisée est indiquée en gras.

Le tableau 2 présente nos résultats, en utilisant la notation F1 ROUGE-2. Nous pouvons voir que RésumeSVD surpasse PMI, SummPip et TextRank dans la plupart des cas. Notre méthode n'est pas toujours la meilleure, mais elle est aussi efficace sur les tâches de résumé à un seul document que sur les tâches de résumé multi-documents, et ne semble pas être affectée par la longueur du document, ce qui est important pour le résumé d'articles scientifiques. Sur les deux corpus multi-documents que nous avons testés, notre méthode surpasse les autres méthodes non supervisées.

On peut voir dans la tableau 2 que la méthode supervisée MatchSum surpasse largement toutes les méthodes non supervisées sur les corpus qui partagent une caractéristique commune : de petits documents sources. Cependant, lorsqu'il s'agit de corpus contenant de plus gros documents (PubMed et Multi-News), l'écart entre MatchSum et notre méthode tend à diminuer.

En temps de calcul, même si notre approche est en moyenne 5 fois mois rapide que TextRank, elle permet de traiter un document en moins de 1/3 de seconde en moyenne. De plus, elle demeure 800 fois plus rapide que PMI et 1600 fois plus rapide que SummPip. Il est a noter également que la méthode état de l'art supervisée MatchSum (Zhong *et al.*, 2020) nécessite plus de 30 heures d'entraînement (pour un seul corpus).

Notre score F1 ROUGE 2 sur le jeu de données Reddit-Tifu est le deuxième derrière TextRank. Il s'agit d'un jeu de données de médias sociaux avec un vocabulaire très spécifique. Nous avons calculé le nombre de mots hors vocabulaire qui est 9 fois plus important que la moyenne du vocabulaire

manquant sur les autres jeux de données. Même si notre approche a montré de bons résultats, il peut être intéressant dans des travaux ultérieurs d'entraîner un modèle de vectorisation de mots sur des jeux de données spécifiques lorsque ces derniers (qui sont génériques) ne couvrent pas suffisamment le vocabulaire du jeu de données.

Notre méthode qui utilise BERT comme méthode de vectorisation à l'échelle des phrases pour le résumé est décevante. En effet, en utilisant la meilleure configuration des couches cachées de BERT pour le résumé de texte, on obtient les résultats présentés dans le tableau 2. Il y a plusieurs aspects à prendre en compte pour interpréter ces résultats. Premièrement, le nombre de phrase dans les documents. En effet, par rapport au nombre de mots, les phrases sont beaucoup moins représentés (voir 1). Cela influence directement la pertinence de l'utilisation de la SVD, censée faire ressortir les dimensions portant le plus d'information. Dans ce contexte, il devient délicat d'utiliser un outil de réduction de dimensionnalité. Deuxièmement, la représentation vectorielle d'une phrase noie l'information pertinente que certains mots apportent par leur présence, avec des mots beaucoup moins importants par rapport au contexte. Or nous montrons expérimentalement par la performance de notre modèle présenté précédemment, l'importance d'une heuristique mettant au centre de son fonctionnement, l'utilisation des mots portant le plus d'information. Après une première analyse des documents et de leurs résumés par BERT SVD, nous avons constaté que les phrases les plus proches des axes du SVD sont assez éloignées du thème principal du document, et échouent globalement à capturer les thèmes essentiels évoqués dans ce dernier. Il est donc intéressant d'en conclure que l'utilisation de la SVD sans autre contexte qu'une unique représentation vectorielle des phrases et de notre heuristique appliquée aux phrases pour le résumé de texte, ne permet pas d'obtenir de bons résumés.

6 Conclusion

Notre méthode, RésumeSVD, fondée sur la représentation vectorielle de mots et des méthodes non supervisées, permet d'obtenir des résumés rapides et fiables. Nous avons présenté deux heuristiques d'extraction capables d'exploiter la matrice de documents réduite. L'une traite les documents uniques, l'autre les documents multiples. L'étude empirique montre des résultats intéressants par rapport à l'état de l'art que ce soit en termes de scores ROUGE ou de temps de calcul. Par rapport aux approches les plus récentes, RésumeSVD est légèrement meilleur en termes de scores ROUGE moyens tout en étant environ 1000 fois plus rapide sur les jeux de données contenant les documents les plus longs. Ces résultats ont été obtenus sans aucune adaptation des modèles de vectorisation ; il est donc possible de les améliorer dans des domaines tels que le médical/scientifique ou les médias sociaux, car ils utilisent un vocabulaire spécifique qui pourrait être mieux traité. Sa polyvalence sur les documents, quel que soit leur type ou leur taille, ouvre la voie à une exploration beaucoup plus poussée des énormes ensembles de données multi-documents, comme Google ou TripAdvisor.

Références

ANKERST M., BREUNIG M. M., KRIEGEL H.-P. & SANDER J. (1999). Optics : Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, p. 49–60, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/304182.304187.

BLEI D. M., NG A. Y., JORDAN M. I. & LAFFERTY J. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, **3**, 2003.

DEERWESTER S., DUMAIS S. T., FURNAS G. W., LANDAUER T. K. & HARSHMAN R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, **41**(6), 391–407.

ESTER M., KRIEGEL H.-P., SANDER J. & XU X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231 : AAAI Press.

FAROUK M. (2018). Sentence semantic similarity based on word embedding and wordnet. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, p. 33–37. DOI : 10.1109/ICCES.2018.8639211.

FORGY E. W. (1965). Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, **21**, 768–769.

GONG H., SAKAKINI T., BHAT S. & XIONG J. (2018). Document similarity for texts of varying lengths via hidden topics. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 2341–2351, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1218.

GONG Y. & LIU X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, p. 19–25, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/383952.383955.

GRAHAM Y. (2015). Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 128–137, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/D15-1013.

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

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

MIHALCEA R. & TARAU P. (2004). TextRank : Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, p. 404–411, Barcelona, Spain : Association for Computational Linguistics.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In Y. BENGIO & Y. LECUN, Éds., *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

NYZAM V., RODRIGUES C. & BOSSARD A. (2018). MOTS : un outil modulaire pour le résumé automatique (MOTS : A modular framework for automatic summarization). In *CORIA-TALN-RJC (TALN)*, p. 101–114 : ATALA.

PADMAKUMAR V. & HE H. (2021). Unsupervised extractive summarization using pointwise mutual information. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 2505–2512, Online : Association for Computational Linguistics.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M. & DUCHESNAY E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

PENNINGTON J., SOCHER R. & MANNING C. (2014). GloVe : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1162.

RADEV D. R., JING H. & BUDZIKOWSKA M. (2000). Centroid-based summarization of multiple documents : sentence extraction utility-based evaluation, and user studies. *CoRR*, **cs.CL/0005020**.

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 (Volume 1 : Long Papers)*, p. 1073–1083, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1099.

STEINBERGER J. & JEZEK K. (2004). Text summarization and singular value decomposition. In T. M. YAKHNO, Éd., *Advances in Information Systems, Third International Conference, ADVIS 2004, Izmir, Turkey, October 20-22, 2004, Proceedings*, volume 3261 de *Lecture Notes in Computer Science*, p. 245–254 : Springer. DOI : 10.1007/978-3-540-30198-1_25.

ZHAO J., LIU M., GAO L., JIN Y., DU L., ZHAO H., ZHANG H. & HAFFARI G. (2020). Summpip : Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, p. 1949–1952, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3397271.3401327.

ZHONG M., LIU P., CHEN Y., WANG D., QIU X. & HUANG X. (2020). Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 6197–6208, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.552.