Modèles graphiques causaux interactifs pour les données textuelles

Amine Ferdjaoui[*,**], Séverine Affeldt[*], Mohamed Nadif[*]

[*] Centre Borelli UMR 9010, Université Paris Cité, 75006 Paris, France.
[**] SogetiLabs, 147 Quai du Président Roosevelt, 92130, Issy-les-Moulineaux.
<prénom.nom>@u-paris.fr

Résumé. Nous proposons de reconstruire des modèles graphiques causaux à partir de données textuelles via un nouveau package Python appelé `WordGraph`. Ce package facilite l'exploration de grands corpus de documents par des visualisations interactives sous forme de modèles graphiques de mots. Le pipeline WordGraph exploite à la fois les *widgets jupyter* et le *notebook jupyter* pour aider les utilisateurs sans expérience en Python à prendre rapidement en main le pipeline `WordGraph`, qui est entièrement personnalisable. `WordGraph` est disponible via un dépôt GitHub, qui fournit également une courte vidéo présentant l'utilisation de notre système.

1 Introduction

1.1 Co-clustering et données textuelles

Soit un ensemble d'objets décrits par un ensemble de caractéristiques organisés sous forme d'une matrice de données. La tâche consistant à partitionner simultanément les deux ensembles est communément appelée *co-clustering* ou *biclustering*. Elle vise à révéler des blocs/co-clusters d'intérêt dans une matrice de données et aboutit généralement à des clusters de lignes et de colonnes plus pertinents et interprétables qu'avec le clustering qui ne s'appuie que sur l'une des deux dimensions. Depuis les travaux de Hartigan (Hartigan, 1972), le co-clustering a trouvé des applications dans de nombreux domaines tels que l'analyse textuelle et la bio-informatique (Affeldt et al., 2020, 2021). Ainsi, il est particulièrement bien adapté aux matrices de données documents×termes, qui sont par essence de grande dimension, *sparses* et de nature directionnelle. Les algorithmes pour de telles matrices de co-occurrences peuvent être dérivés de différentes approches. Les méthodes de type spectral, qui traitent la matrice d'entrée comme un graphe bipartite entre les documents et les mots, approximent la coupe normalisée de ce graphe à l'aide d'une relaxation réelle (Dhillon, 2001). Les méthodes basées sur un modèle dérivé des modèles de blocs latents (LBM) appropriés (Govaert et Nadif, 2013, 2018), reposent sur des algorithmes d'*expectation-maximization* (Salah et Nadif, 2019). Le co-clustering peut également utiliser des méthodes basées sur la factorisation matricielle telles que la factorisation matricielle non négative (NMF) (Febrissy et al., 2022) ou la tri-factorisation (NMTF) (Salah et al., 2018). Les méthodes basées sur la théorie de l'information,

utilisées avec des tableaux de contingence à deux voies, minimisent une perte d'information mutuelle en regroupant les lignes et les colonnes de la matrice en fonction du co-clustering (Govaert et Nadif, 2018). Notons aussi que la tâche de co-clustering peut être également réalisée à partir d'une version adaptée de la mesure de modularité habituellement utilisée pour les réseaux (Labiod et Nadif, 2011). Le package `CoClust` est bien connu parmi les packages populaires dédiés au co-clustering pour les matrices de co-occurrences, en particulier pour les matrices document-terme dans les applications de text mining (Role et al., 2019). Il fournit les implémentations de trois algorithmes qui ont prouvé leur efficacité dans le traitement de telles matrices [1]. `WordGraph` est le premier package Python qui facilite l'exploration visuelle des thèmes de grands corpus basés sur le co-clustering. Dans les sections suivantes, nous montrons comment le package Python `WordGraph` tire parti de `CoClust` pour permettre l'exploration de grands corpus et du vocabulaire associé à l'aide de modèles graphiques.

1.2 Inférence de réseaux causaux

La reconstruction de modèles graphiques est une méthode connue dans de nombreux domaines. Ces modèles peuvent être appris à partir de données de séries temporelles, d'expériences de perturbation contrôlée ou de données d'observation, comme c'est le cas en biologie. Les méthodes traditionnelles qui exploitent les données non perturbées comprennent le *search-and-score* bayésien, l'entropie maximale ou la carte de diffusion. Au-delà de la reconstruction de modèles graphiques, la découverte de la causalité entre plusieurs variables est d'un grand intérêt. Par exemple, elle peut conduire à l'identification de facteurs de transcription, régulateurs clés pour des maladies complexes. Appliqué à des données textuelles, elle pourrait fournir des *résumés graphiques* des thèmes par le biais de réseaux dirigés. Les méthodes basées sur les contraintes (Pearl et Verma, 1995) peuvent identifier les contraintes structurelles correspondant aux arêtes inutiles d'un graphique tout en découvrant la causalité à partir de données non perturbées. L'algorithme de causalité inductive basée sur l'information multivariée, `MIIC` [2], repose sur une méthode issue de la théorie de l'information qui combine l'apprentissage basé sur les contraintes et le cadre du maximum de vraisemblance (Affeldt et Isambert, 2015 ; Affeldt et al., 2016 ; Verny et al., 2017). `MIIC` est basé sur l'analyse d'information multivariée, qui étend l'information mutuelle à plus de deux variables. L'intégration des méthodes basées sur les contraintes dans un cadre de théorie de l'information améliore considérablement la précision de la prédiction, le temps d'exécution et les capacités de mise à l'échelle en termes de taille d'échantillon et de taille de réseau par rapport aux approches traditionnelles. Ainsi, alors que `MIIC` a démontré son efficacité sur une variété d'ensembles de données génomiques (Verny et al., 2017), il semble également être particulièrement bien adapté pour aborder les données textuelles de haute dimension, facilitant ainsi l'exploration des thèmes de grands corpus. `WordGraph` est le premier package Python qui applique `MIIC` aux données textuelles. C'est aussi le premier package qui combine la reconstruction de graphe et le co-clustering pour découvrir les thématiques de corpus et leurs interactions. Dans ce qui suit, nous détaillons l'interface de programmation proposée par `WordGraph` et montrons comment elle combine naturellement les approches de co-clustering, qui facilitent la découverte des thé-

1. https ://coclust.readthedocs.io/en/v0.2.1/install.html
2. Voir https ://github.com/miicTeam/miic_R_package pour l'implémentation R de `MIIC`

matiques d'un corpus, et la reconstruction de modèles graphiques causaux, qui identifient les interactions dirigées entre les termes.

2 Description générale du pipeline

Notre pipeline se décompose en trois parties. Tout d'abord, il propose la création d'un corpus à partir de documents chargés localement et dont la thématique globale est donc choisie par l'utilisateur ; extraits par exemple de la base de données en ligne PubMed [3] (Fig. 1, *a*), qui comprend plus de 35 millions de citations pour la littérature biomédicale. La création du corpus inclut le prétraitement des données textuelles (par exemple, le *stemming* ou l'élimination des mots peu informatifs) et la construction d'une matrice documents×termes, avec une procédure de pondération TF-IDF (Term Frequency-Inverse Document Frequency). Ensuite, `WordGraph` permet le co-clustering de la matrice documents×termes pondérée pour explorer les thématiques sous-jacentes. Cette partie s'appuie sur le package `Coclust` et fournit le regroupement simultané des mots et des documents (Fig. 1, *b*, haut). Dans cette étape, on peut également obtenir un extrait de la matrice documents×termes avec les colonnes correspondant aux termes les plus similaires à un mot donné, sur la base du score de *similarité cosine* (Fig. 1, *b*, bas). Enfin, `WordGraph` fournit une interface de programmation Python du package R `MIIC` qui permet la reconstruction des réseaux causaux à partir de données d'observation. Un modèle graphique peut être obtenu à partir des termes les plus représentatifs - typiquement, les mots les plus fréquents - d'un co-cluster (Fig. 1, *c*, haut). Les visualisations de graphes, qui utilisent la bibliothèque Python `ipycytoscape` [4], sont interactives. L'utilisateur peut réorganiser les nœuds, obtenir des vues agrandies et cliquer sur un sommet pour construire un graphe secondaire, *One word similarity*, basé sur un extrait de la matrice document×term (Fig. 1, *c*, bas). On peut également obtenir ce dernier type de graphe en entrant un mot du vocabulaire dans le champ de saisie *widget*. Dans ce qui suit, nous fournissons les détails du code source de `WordGraph` et expliquons les exploitations possibles à plusieurs niveaux. En effet, le package proposé peut être utilisé tel quel, dans une source Python ordinaire, ou bien l'utilisateur peut personnaliser un *notebook jupyter* interactif. Enfin, une application web peut être instantanément générée à partir du notebook pour faciliter le partage en ligne.

3 Un package à plusieurs niveaux

Interface de programmation Python pour les réseaux causaux `WordGraph` est le premier package Python qui propose une interface de programmation au package R `MIIC`. La figure 2 présente les principales fonctionnalités de `WordGraph` et de ses modules. Le package peut charger localement un corpus autour d'une thématique librement choisie ou, par exemple, récupérer des documents de PubMed (module *io*). `WordGraph` intègre MIIC dans la méthode *build_graph*, et s'appuie sur les fonctions *create_miic* et *preproc_graphML* (module *utils*). *utils* permet également le prétraitement du corpus. Un exemple d'utilisation simple de `WordGraph`, depuis la recherche dans PubMed au calcul de l'objet graphique `MIIC`, se

3. https ://pubmed.ncbi.nlm.nih.gov
4. https ://ipycytoscape.readthedocs.io/en/latest/

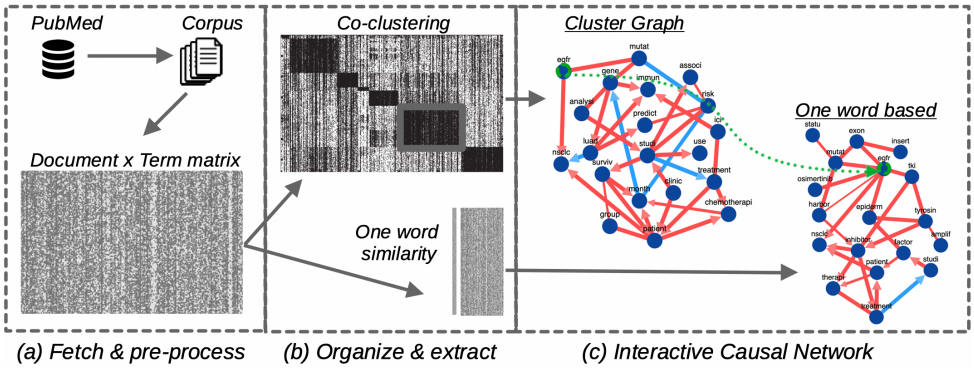

FIG. 1 – WordGraph : de la recherche de documents aux réseaux causaux interactifs de mots.

trouve dans la section *Simple usage* de la présentation GitHub en ligne du package [5]. Nous fournissons dans `WordGraph` un *jupyter notebook* qui intègre cette utilisation simple (voir *WordGraph utilisation simple.ipynb*).

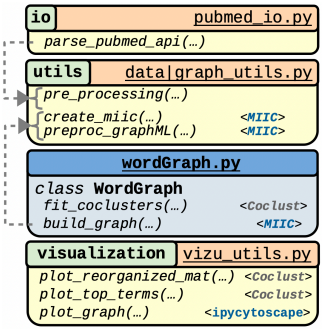

FIG. 2 – Functionnalités principales de WordGraph.

Notebook interactif d'exploration `WordGraph` dispose d'un module *visualization* qui utilise le package `ipycytoscape`. En particulier, *plot_graph* affiche des réseaux interactifs dans un *notebook jupyter*, où la disposition des graphes peut être ajustée à l'aide de la souris. Il fournit des histogrammes de fréquence pour les principaux termes des co-clusters et permet ainsi de résumer facilement les thématiques (Fig. 3, à gauche). Un clic sur un nœud du modèle graphique de mots permet l'affichage d'un réseau interactif secondaire, dans lequel les sommets sont choisis en fonction de leur similarité avec le nœud précédemment sélectionné (Fig. 3, droite). De plus, `WordGraph` propose un *notebook jupyter* facilement personnalisable (*WordGraph Custom.ipynb*), dans lequel l'ensemble du pipeline peut être incorporé dans une seule cellule. Le notebook intègre plusieurs éléments `ipywidgets` (par exemple, un curseur, un bouton, des entrées textuelles et numériques). La figure 4 montre comment le pipeline s'organise dans une seule cellule.

5. https ://github.com/AmineFrj/WordGraph#1-simple-usage

La modification ou l'ajout d'un *widget* ne nécessite que quelques lignes de code. La section *Customizable notebook pipeline* de la présentation GitHub en ligne du package [6] fournit l'implémentation pour la mise en place d'un bouton *confirm* et l'appel de la fonction correspondante lors du clic.

Application web interactive Le notebook *WordGraph Custom.ipynb* a également été conçu pour être automatiquement transformé en application web, avec le package *voilá* [7]. Ainsi, l'utilisateur peut exploiter le *WordGraph Custom.ipynb* tel quel ou le personnaliser, puis obtenir facilement l'application web correspondante (Fig. 5), prête à l'emploi pour les utilisateurs qui ne sont pas familiers avec la programmation.

4 Application biomédicale

Nous avons généré trois corpus en lien avec le cancer et d'une taille de $10,000$ documents, et avons exploré leurs thématiques avec `WordGraph`. La Figure 1 se focalise sur le **cancer du poumon**. L'opposition (lien bleu) entre *NSCLC* (Non-Small Cell Lung Cancer) et *LUAD* (LUng ADenocarcinoma) suggère que les documents traitant de *LUAD* ignorent le sujet *NSCLC*. En effet, même si les cancers de type *LUAD* appartiennent à la famille *NSCLC*, leur pronostic et leur expression génétique sont très distincts et ils sont considérés comme des entités séparées. Le cancer *EGFR*-positive NSCLC résulte d'une mutation du gène EGFR (récepteur du facteur de croissance épidermique). Notre réseau de mots établit un lien causal entre la présence dans les documents du terme *EGFR* et de l'acronyme *NSCLC*. Le terme *ICI* (Immune Checkpoint Inhibitor) *treatment* [8] a également une relation de cause à effet avec le nœud *immun*. Le réseau secondaire, basé sur *egfr*, se concentre sur le *treatment*, qui vise généralement à prévenir la croissance anormale de la tumeur en ciblant la *Tyrosin* Kinase *Inhibitor* (*TKI*), comme le permet le traitement par *osimertinib*. Les modèles graphiques de la Figure 3 correspondent à un corpus centré sur le **glioblastoma cancer**. Le glioblastome multiforme (GBM) est le type

6. https ://github.com/AmineFrj/WordGraph#2-customizable-notebook-pipeline
7. https ://voila.readthedocs.io/en/stable/using.html
8. Les documents des corpus étant en anglais pour cette application, nous rendons compte des termes également en anglais, tels qu'il apparaissent dans les réseaux. La procédure de *stemming* explique la troncature de certains mots.

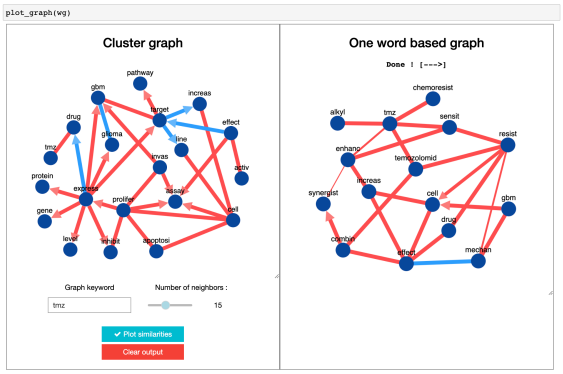

FIG. 3 – Graphes interactifs dans le notebook jupyter.

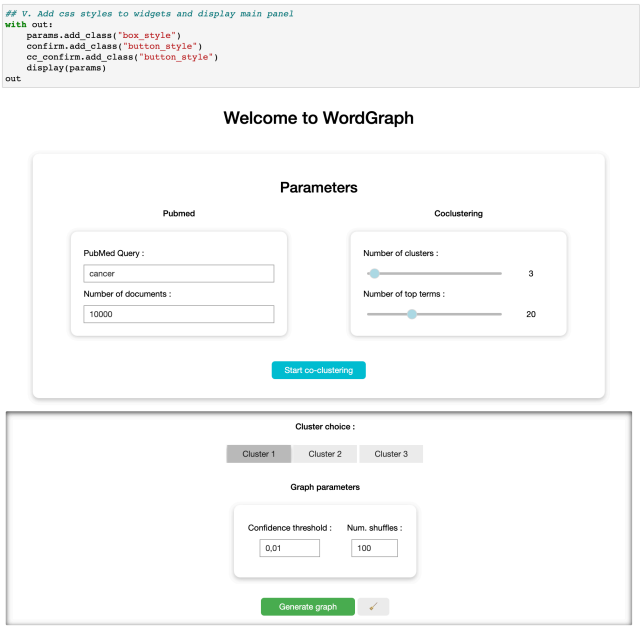

FIG. 4 – Pipeline interactif dans un jupyter notebook.

de cancer du cerveau le plus aggressif, avec une augmentation de l'*invas*ion, de la *prolifer*ation et une diminution de l'*apoptosi*s, comme le détaille notre graphe. Nous remarquons également le nœud *TMZ*, qui est le médicament (*drug*) chimiothérapeutique prédominant dans le GBM. Le graphe secondaire (Figure 3, droite), qui se concentre sur *TMZ*, apporte d'autres informations. En particulier, nous comprenons que la tumeur *GBM* a tendance à développer une *TMZ* (*temozolomid*) *resist*ance ou *chimioresist*ance. Enfin, la Figure 5 propose une exploration d'un corpus qui concerne le **cancer du sein**. En particulier, le graphe de cluster met l'accent sur la possibilité de *drug resist*ance pour ce cancer. Ceci est particulièrement vrai pour le cancer du sein de type Triple Négatif *TNBC*. Séparé du graphe de cluster (Figure 5, gauche), le *TNBC* s'impose comme une sous-thématique, pour laquelle le graphe secondaire (Figure 5, droite) fournit des informations supplémentaires. Plus précisément, il s'agit d'un *subtype agressive*, dont le *prognosi*s est généralement *poor*.

5 Conclusion

Nous avons créé `WordGraph` afin de fournir un package Python facile à exploiter pour les utilisateurs souhaitant extraire un maximum d'informations autour d'un sujet pas nécessairement du domaine biomédical. `WordGraph` est une application web autonome pour les utilisateurs n'ayant aucune connaissance de Python ou de *jupyter notebook*, tout en permettant la personnalisation de l'interface. Nos applications d'exploration sur divers corpus autour du cancer démontrent l'intérêt de notre package, qui propose la première interface de programmation Python du package R `MIIC`, pour la reconstruction de réseaux causaux.

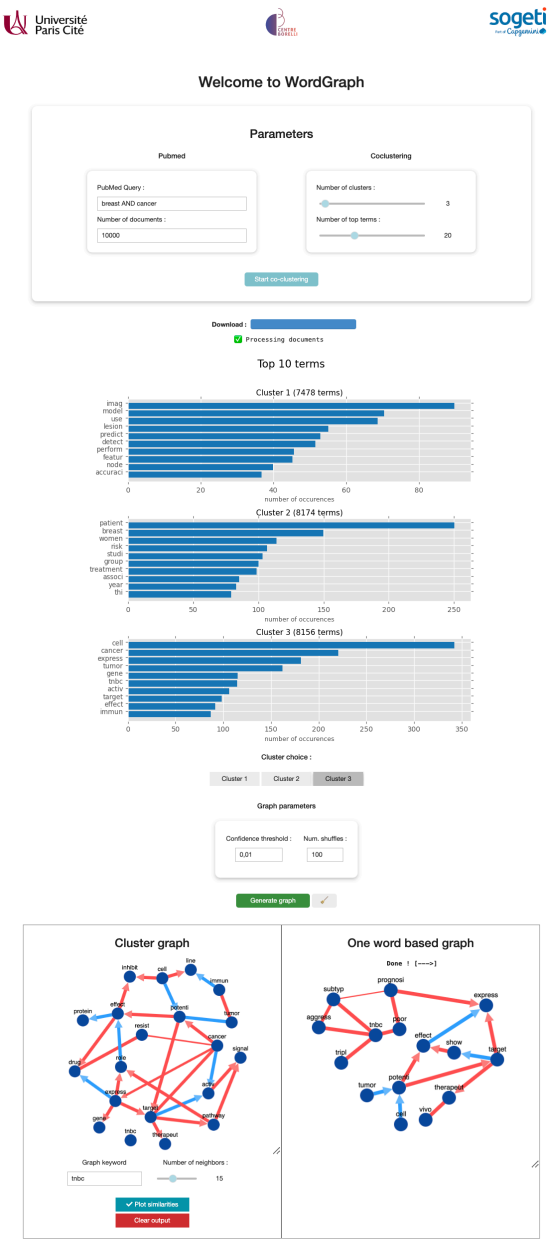

FIG. 5 – Application web automatiquement générée.

Remerciements Ce travail est soutenu par une subvention de l'ANR (ANR-19-CE23-0002). Il a reçu la labellisation des pôles de compétitivité *Cap Digital* et *EuroBiomed*. Nous remercions le Isambert Lab pour son support concernant le package R `MIIC`.

Références

Affeldt, S. et H. Isambert (2015). Robust reconstruction of causal graphical models based on conditional 2-point and 3-point information. In *ACI@ UAI*, pp. 1–29.

Affeldt, S., L. Labiod, et M. Nadif (2020). Ensemble block co-clustering : a unified framework for text data. In *the 29th ACM CIKM*, pp. 5–14.

Affeldt, S., L. Labiod, et M. Nadif (2021). Regularized bi-directional co-clustering. *Statistics and Computing 31*(3), 32.

Affeldt, S., L. Verny, et H. Isambert (2016). 3off2 : A network reconstruction algorithm based on 2-point and 3-point information statistics. In *BMC bioinformatics*, Volume 17, pp. 149–165.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *the 7^{th} SIGKDD*, pp. 269–274.

Febrissy, M., A. Salah, M. Ailem, et M. Nadif (2022). Improving NMF clustering by leveraging contextual relationships among words. *Neurocomputing 495*, 105–117.

Govaert, G. et M. Nadif (2013). *Co-clustering : models, algorithms and applications*. John Wiley & Sons.

Govaert, G. et M. Nadif (2018). Mutual information, phi-squared and model-based co-clustering for contingency tables. *Advances in data analysis and classification 12*, 455–488.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the american statistical association 67*(337), 123–129.

Labiod, L. et M. Nadif (2011). Co-clustering for binary and categorical data with maximum modularity. In *the 11^{th} International Conference on Data Mining*, pp. 1140–1145. IEEE.

Pearl, J. et T. S. Verma (1995). A theory of inferred causation. In *Studies in Logic and the Foundations of Mathematics*, Volume 134, pp. 789–811. Elsevier.

Role, F., S. Morbieu, et M. Nadif (2019). Coclust : a python package for co-clustering. *Journal of Statistical Software 88*, 1–29.

Salah, A., M. Ailem, et M. Nadif (2018). Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In *AAAI*, Volume 32.

Salah, A. et M. Nadif (2019). Directional co-clustering. *Advances in Data Analysis and Classification 13*, 591–620.

Verny, L., N. Sella, S. Affeldt, P. P. Singh, et H. Isambert (2017). Learning causal networks with latent variables from multivariate information in genomic data. *PLoS computational biology 13*(10), e1005662.

Summary

We propose to reconstruct causal graphical models from textual data via a new Python package, `WordGraph`. This package facilitates the exploration of large corpora of documents through interactive visualizations in the form of graphical word models, which is available via a GitHub repository.