Université de Rouen Normandie - UFR Sciences et Techniques
Master 2 mention Bioinformatique – Parcours BIMS
2023 - 2024

Rapport de stage

Analyse textuelle d'article scientifiques evaluant l'impact des vers de terre sur l'environnement

Présenté et soutenu par

Antoine Malet

Campus Agro Paris Saclay, Unité MIA Paris-Saclay
Equipe SOLsTIS

Encadrants :

David Makowski
Sophie Donnet

Université de Rouen Normandie - UFR Sciences et Techniques
Master 2 mention Bioinformatique – Parcours BIMS
2023 - 2024

Rapport de stage

Analyse textuelle d'articles scientifiques evaluant l'impact des vers de terre sur l'environnement

Présenté et soutenu par

Antoine Malet

Campus Agro Paris Saclay, Unité MIA Paris-Saclay
Equipe SOLsTIS

Encadrant :

David Makowski
Sophie Donnet

Remerciements

En premier lieu, j'aimerais remercier mes encadrants pour ce stage, Mme Sophie DONNET et M. David MAKOWSKI, pour avoir accepté ma candidature et acceuilli au sein de leur équipe. Merci aussi pour leurs conseils et leurs effort d'accompagnement et de relecture de mes travaux ! Je tiens aussi à adresser un mot particulier à mes vaillants collègues de bureau Emré ANAKOK et Caroline COGNOT, pour leur compagnie perpétuelle et leurs très bons conseils.

Je remercie aussi Louis LACOSTE, pour ses excellents conseils en R et en cinématographie, ainsi que François VICTOR, pour ses généreuses explications en statistiques théoriques auxquelles je n'ai pas compris grand-chose. Merci aussi à Armand FAVROT, pour son acceuil et ses invitations aux Eventos des repas organisés par la cafet (les fraises maison étaient légendaires !). Enfin, chaleureuses salutations à tous mes collègues de pause café, qui sont toujours restés sympathiques et acceuillants même si je n'ai jamais bu la moindre goutte de café. Courage, peut-être qu'un jour vous me convertirez à votre religion !

Merci enfin à tous ceux que je n'ai pas nommés, particulièrement aux personnels qui prennent soin des locaux en silence, sans qui l'infrastructure de travail ne pourrait pas fonctionner correctement.

Table des matières

Table des figures

Liste des Abréviations

ASCII American Standard Code for Information Interchange
API Application Programming Interface
CSS Cascade Style Sheet
DOI Digital Object Identifier
HTML Hypertext Markup Language
IDE De l'anglais, Environnement de Développement Intégré
MA Métaanalyse
MIA Mathématiques et Informatique Appliquée
Rmd R Markdown
RG ResearchGate
RGS2 ResearchGateScrapper2.py
UMR Unité Mixte de Recherche

Glossaire

JSON : JSON (JavaScript Object Notation) est un format de fichier textuel conçu pour la structuration et l'échange de données (https ://www.hostinger.fr/tutoriels/quest-ce-que-json).

Markdown : Markdown est un langage de balisage léger qui permet de formater du texte de manière simple et rapide. Il utilise des caractères spéciaux pour indiquer les éléments de mise en forme, tels que les titres, les listes, les liens, etc. Les fichiers Markdown peuvent être convertis en HTML pour être affichés sur un site web ou dans un logiciel de traitement de texte (source : https ://bility.fr/definition-markdown/).

Métaanalyse : Article scientifique présentant la combinaison des résultats statistiques d'une série d'études indépendantes sur un problème donné.

N-gram : Suite de mots consécutifs de taille n (une des possibilités de tokenisation). Utile pour comprendre les relations logiques entre les mots. Les bigrammes sont un cas particulier de n-gram (n-gram de longueur 2).

Racinisation (linguistique) : Obtention du radical, par exemple par dépréfixation ou désuffixation (*Exemple :*"enhance", "enhances" et "enhancement" deviennent tous "enhanc").

Réseau de n-grams : Figure permettant de visualiser toutes les relations entre les différents tokens simultanément, plutôt que deux par deux. Cela permet d'aller plus loin que l'analyse de bigrammes séparés les uns des autres.

Sélecteurs CSS : Les sélecteurs définissent les éléments sur lesquelles s'applique un ensemble de règles CSS (langage de programmation employé pour mettre en forme une page Web). Ils peuvent être employés en Web scraping pour cibler et isoler certains éléments d'intérêt.

Token : Unité textuelle souvent réduite, voire ne comprenant qu'un seul mot, issue du processus de tokenisation.

Tokenisation : Processus consistant à découper un texte ou un corpus de textes en unités textuelles plus réduites, comme des mots, des n-grams ou des phrases.

Text-mining : Processus d'analyse textuelle consistant à transformer un texte non structuré en données structurées pour ensuite procéder à l'analyse. Cette pratique repose sur la technologie de « Natural Language Processing » (traitement du langage naturel), permettant aux machines de comprendre et de traiter le langage humain automatiquement (source : https ://datascientest.com/text-mining-definition).

Web scraping : Technique permettant d'extraire automatiquement de grandes quantités d'informations d'un site Web, sans intervention humaine directe, via un script informatique (source : https ://moncoachdata.com/blog/web-scraping-pratique/).

Chapitre 1

Introduction

1.1 Unité MIA, Campus Agro Paris Saclay

Mon stage s'est déroulé au sein de l'Unité MIA (Mathématique et Informatique Appliqués), à l'INRAE du Campus Agro Paris Saclay, sur le plateau de Saclay. L'UMR MIA Paris-Saclay, associée aux tutelles AgroParisTech, INRAE et Université Paris Saclay, regroupe des statisticiens et des informaticiens spécialisés dans la modélisation et l'apprentissage statistique et informatique pour la biologie, l'écologie, l'environnement, l'agronomie et l'agro-alimentaire.

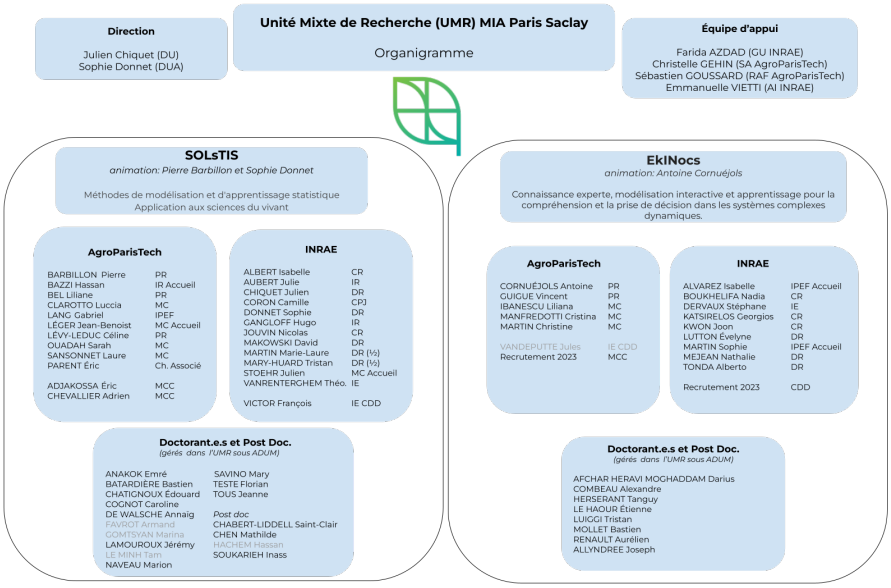

Figure 1.1 – Organigramme de l'UMR MIA Paris-Saclay

Les compétences mises en oeuvre portent sur les méthodes d'inférences statistiques et algorithmiques. L'unité développe des méthodes statistiques et informatiques originales, génériques ou motivées par des problèmes précis en science du vivant. Ses activités s'appuient sur une bonne culture dans les disciplines destinatrices : écologie, environnement, agro-alimentaire, biologie moléculaire et biologie des systèmes. L'unité MIA est dirigée par Julien CHIQUET et Sophie DONNET, et comprends deux équipes distinctes : l'équipe SOLsTIS (Statistical mOdeling and Learning for environnemenT and Life Science) dirigée par Sophie

DONNET et Pierre BARBILLON, et l'équipe EkiNocs (Expert Knowledge, INteractive modelINg for understandINg and decisiOn makINg in dINamic Complexe Systems), dirigée par Antoine CORNUÉJOLS.

En tant que stagiaire, j'ai ainsi pu intégrer SOLsTIS (figure 1.1) pour mettre au point des méthodes informatiques et statistiques pour l'analyse textuelle d'abstracts d'articles scientifiques. Dans ce cadre, j'ai aussi pu participer aux interventions de divers spécialistes du domaine, lors des séminaires hebdomadaires tenus le jeudi.

1.2 Le vers de terre dans la littérature scientifique

Pour caractériser le rôle écologique des vers de terre d'après la littérature scientifique disponible, les abstracts (courts résumés **standardisés**) de 116 articles scientifiques issus de 4 métaanalyses distinctes ont été fournis sous forme de fichier CSV pour être analysés informatiquement. Selon la perception actuelle des spécialistes du sujet en Europe, les services écosystémiques rendus par les vers de terre sont multiples (figure 1.2)[7] :

- Ils permettent une meilleure aération des sols ([6]).
- Ils favorisent le recyclage des éléments nutritifs, du carbone, et du phosphore ([8]).
- Ils jouent un rôle important dans le recyclage de la matière organique des sols ([3]).
- Leur activité de décomposeurs (minéralisation de la matière organique) favorise la croissance d'autres organismes de l'environment, augmentant de ce fait la productivité des plantes ([1]).
- Ils sont essentiels pour la structuration, l'entretien et la productivité des sols, forestiers, prairiaux et agricoles ([11]).

Cependant, la position argumentative défendue par les spécialistes d'Amérique du Nord est très différente, pour ne pas dire radicalement opposée. On peut résumer la perception nord-américaine comme suis :

- Les vers de terre exotiques (venus d'Europe et d'Asie) sont des espèces invasives suceptibles de perturber les écosystèmes natifs ([9]). Voir figure 1.3.
- Leur présence dans les sols modifie durablement les propriétés physico-chimiques des écosystèmes sousterrains ([2]).
- Les modifications physico-chimiques induites par les espèces exotiques causent une baisse globale de la biodiversité chez les espèces natives du milieu ([4]).
- L'impact de ces espèces exotiques sur les émissions de gaz à effet de serre est également sujet à controverse ([10, 5]).

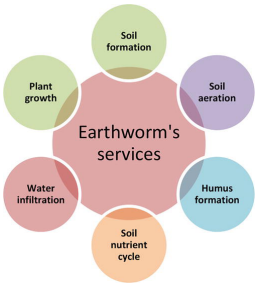

Figure 1.2 – Les services écosystémiques rendus par les vers de terre.

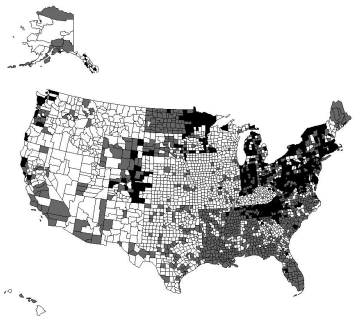

Figure 1.3 – Distribution régionale de Lumbricus rubelus (espèce exotique) aux USA. Les régions représentent les stations où : L. rubelus a été retrouvé (**noir**), une autre espèce que L. rubelus a été retrouvée (**gris**), aucune donnée n'est disponible (**blanc**). *Source :* `https://daphnia.ecology.uga.edu/drakelab/?p=318`.

1.3 Objectifs de mon travail

L'analyse infomatique des abstracts des articles issus des 4 métaanalyses a été conduite pour répondre à la question de recherche suivante :

Les méthodes statistiques de Text-mining permettent-elles d'identifier des groupes distincts d'articles scientifiques défendant une conception opposée du rôle écologique du vers de terre au sein du corpus de texte fourni pour l'analyse ?

Pour répondre à cette question, nous procéderons en deux étapes :

1. Création d'une base de métadonnées (comprenant notamment les **abstracts** de chaque étude indépendante formant les 4 métaanalyses) via un script Python de Web-scraping, sous forme d'un fichier CSV brut destiné à être analysé.
2. Analyse textuelle des abstracts d'articles scientifiques présents dans le fichier CSV issu de l'étape précédante via un script R.

Une fois ces étapes achevées, nous discuterons les résultats obtenus et les méthodes employées en les comparant avec ceux d'autres articles de la littérature scientifique spécialisés dans ce domaine.

Chapitre 2

Ressources : pratiques professionnelles, environnement informatique, outils informatiques et statistiques, données

2.1 Environnement informatique

Le matériel qui m'a été fourni par le laboratoire est un ordinateur fixe HP Elite SFF 800 G9 Desktop PC, numéro de série CZC2479ZXS - 4G087AV, avec 31 Go de mémoire vive et un processeur 12th Gen Intel® Core™ i7-12700 × 20, ainsi qu'un processeur graphique Mesa Intel® UHD Graphics 770. Il est géré par un système d'exploitation Ubuntu 22.04 64-bits (distribution Linux), sous la version 42.9 de GNOME (GNU Network Object Model Environment) fournissant une interface utilisateur ergonomique pour interagir avec le système d'exploitation GNU (*GNU's Not Unix*). Il est ainsi composé d'un noyeau Linux et de GNU, qui ensemble forment le système d'exploitation communément connu sous le nom "Linux". Le système de fenêtrage est "Wayland", un système chargé de l'affichage et du placement des fenêtres durant l'usage du système d'exploitation par l'utiisateur.

2.2 Pratique Professionnelle

2.2.1 Veille bibliographique et technologique

Pour la veille bibliographique, je me suis aidé du script de Web-scraping développé dans le cadre de mon stage, qui, à partir d'un fichier comprenant les titres de nombreux articles cible, m'a permis de récupérer les métadonnées associées de maniere semi-automatique. Les journaux consultés (*Applied Soil Ecology, Ecosystems, Soil Biology and Biochemistry*, etc.) étaient majoritairement orientés sur les études écologiques. Un travail par recherche MESH a aussi été réalisé, en retenant les mots-clés **Oligochaetas, Earthworm, Lumbricus terrestris, Ecosystems, Introduced species, Soil, Data mining** et **Meta-Analysis**. Pour gérer cette bibliographie, j'ai employé le package LaTeX *cleveref* conçu dans ce but.

2.2.2 Bonne pratique de programmation informatique et de developpement logiciel

Pour l'écriture du code Python, j'ai utilisé l'éditeur de code Visual Studio Code (VScode). Pour la conception du script Rmd, j'ai travaillé sous l'IDE RStudio. Conformément aux bonnes pratiques, l'intégralité des scripts (Python et R) ont été commentés en anglais, pour

faciliter la relecture du code. Pour tester le fonctionnement de chaque script, deux approches différentes ont été mises en place :

Pour le script de Web-scraping en Python, les tests de fonctionnement effectués durant le développement ont été réalisées sur des jeux de données réduits (comprenant le plus souvent seulement les dix premiers articles) afin de pouvoir détecter rapidement si la sortie renvoyée est correcte par rapport à la tâche initiale. Pour la récupération de données via Crossref, des tests ont aussi été réalisées, notamment pour l'exploration des structures JSON renvoyées par la fonction works() de l'API. Une fois les données recherchées obtenues en sortie de tests ponctuels, il a été possible de généraliser la méthode, en l'appliquant dans l'ensemble du script principal. Enfin, afin d'éviter la régression du code (perte de fonctionnalité), j'ai souvent travaillé sur deux scripts identiques mais distincts, le premier servant de script principal, tandis que le deuxième était davantage un support pour le développement de nouvelles fonctionnalités. De cette façon, le script principal n'était mis à jour que lorsque le script secondaire était considéré comme fonctionnel. Un dépôt GitHub personnel aurait aussi pu jouer ce rôle, mais comme je travaillais seul sur cette partie, je n'en ai pas ressenti l'utlité.

Pour le script de Text-mining en Rmd sous RStudio, la plupart des tests de fonctionnement au cours du dévéloppement ont été réalisés dans la console R directement, afin d'éviter d'ajouter de nouveaux objets inutiles "test" à l'environnement R. Les processus n'ont été intégrés au script Rmd proprement dit qu'une fois leur fonctionnement testé et validé dans la console. Dans cette démarche, le fonctionnement de l'environnement R a été très utile, car cela a permis de réemployer certains objets stockés en mémoire sans avoir à les redéfinir seulement en vue d' effectuer des tests.

2.2.3 Communication des travaux

En concertation avec mes encadrants, j'ai aussi utilisé un dépôt GitHub spécialement mis en place pour le projet entre eux et moi, où mon travail de chaque jour a pu être sauvegardé grâce à un mécanisme de Push/Pull. De cette façon, mes encadrants ont pu facilement suivre l'évolution de mon travail et évaluer la qualité des solutions proposées. Pour faciliter l'utilisation de cet outil, le logiciel GitHub Desktop m'a été présenté, une interface utilisateur graphique facilitant grandement la visualisation et l'usage du dépôt GitHub mis en place pour le projet. Grâce à la fonctionnalité Knitr de Rmd, j'ai pu rendre compte de ma progression quotidienne en produisant automatiquement un fichier rapport au format HTML, directement issu de mon code (figures produites sous R, titres et intérprétation rédigées en Markdown ou HTML). Les réunions avec mes encadrants, le plus souvent hebdomadaires, ont été fixées par échange de mails ou bien organisées de vive voix sur site. Elles m'ont permis de rester bien focalisé sur la mission en me fournissant des objectifs hebdomadaires clairs et précis, tout en permettant à mes encadrants d'être régulièrement informés de ma progression par rapport aux objectifs fixés.

2.3 Outils informatiques et statistiques pour les différentes phases de vos travaux

2.3.1 Récupération des abstracts et des métadonnées avec Python

Pour l'élaboration du script Python, j'ai surtout utilisé le package Python Habanero Crossref pour requêter (en utilisant une API) la base de données Crossref, qui contient une grande quantité de métadonnées (donnés sur les articles en tant que tel, comme l'abstract, les auteurs, etc.), afin de récupérer les informations voulue à propos d'un ensemble de titres

d'articles scientifiques défini au préalable sur le sujet des vers de terre. Pour compléter les données récupérées (la base de Crossref comprenant de nombreuses données manquantes), j'ai aussi développé en parallèle un module pour récupérer les données d'intérêt dans le code source de ResearchGate (web scraping), comme le *DOI (Digital Object Identifier)* de chaque publication, la date de chargement sur la base de RG ou encore le lien vers la page RG correspondante. Pour parvenir à cette solution, voici la liste des modules qui ont été utilisés :

Pandas : Pandas est un module qui permet de manipuler facilement des tableaux de données avec des étiquettes de variables (colonnes) et d'individus (lignes). Il est notamment utilisé dans le script pour exporter les résulats issus du code Python vers un fichier CSV(coma separated values), lisible et modifiable à l'aide d'outils de bureautique courants comme LibreOffice Calc ou Excel.

Numpy : Package conçu pour le calcul scientifique avec Python. Il est très utile pour l'algèbre (comme par exemple pour la manipulation de matrices), et son implémentation en C, C++ et Fortran en fait un outil de calcul rapide et efficace pour l'analyse de données et le calcul scientifique. Dans le code, cela dit, il sert simplement à l'indexage lors de la création du DataFrame de résultats.

Itertools : Module implémentant des outils Python pour maîtriser plus subtilement les itérations. La méthode employée dans le code est *zip_longest*, qui permet de créer un DataFrame à partir d'une liste de listes de tailles potentiellement différentes. La plus longue sera employée en référence (longest), et toutes les autres seront ajustées à cette longueur par l'ajout d'une *fillvalue* ("null", dans le code). Dans mon travail, elle a servi à créer le DataFrame requis à partir de listes de données de tailles pas forcément égales (à cause des valeurs manquantes).

Habanero : Module client de bas niveau pour interroger l'API Crossref, une base de données contenant les métadonnées des articles de tous les membres (des informations comme le titre, le nom d'auteur, le DOI etc.). Dans le code Python, elle est employée surtout pour rechercher les noms d'auteurs, les autres champs testés n'étant pas assez fiables pour automatiser complètement la récupération d'informations. Crossref est codé comme une **classe** du module Habanero, comprenant les méthodes works(), members(), prefixes(), funders(), journal(), type() et licence(). Dans le code Python, seule la méthode works() a été employée pour envoyer une requête à partir du titre de chaque article.

Unidecode : Module contenant entre autres la fonction éponyme *unidecode* (employée dans le script) conçue pour transformer les chaînes de caractères contenant des caractères non-ASCII (comme par exemple des idéogrammes chinois) pour les traduire en chaînes de caractères contenant uniquement des caractères ASCII. Dans le code, la fonction *unidecode* est employée pour rendre l'affichage des noms d'auteur contenant des caractères non-ASCII. Certaines corrections sont imparfaites et retirent quelques lettres.

RGS2 : Sous-module codé localement à partir d'un exemple trouvé en ligne [1] , par la suite adapté pour récupérer directement les informations voulues dans le code source du site scientifique ResearchGate, les autres options potentielles (comme Google Scholar) ayant souvent un système de détection et de blocage des bots. Seule la deuxième version du sous-module a été retenue dans le projet final. Il dépend des modules suivants :

1. Module **Parsel**, fonction *Selector* : Module facilitant l'extraction des données pour les formats HTML, JSON et XML. Dans le code, il est utilisé pour trouver les données recherchées directement dans le code source de la page (web scraping) en s'appuyant sur des sélecteurs CSS.

1. citation url à mettre ici (scrape publications from RG)

2. Module **playwright.sync_api**, fonction *sync_playwright* : Module permettant de lancer une session navigateur depuis un script Python. Dans le code, il est utilisé pour se rendre sur le site de ResearchGate via une session Chromium, un navigateur libre développé par Google.
3. Module **re** : Module fournissant des opérations sur les expressions rationnelles utilisable dans un code Python. Dans le code Python, il est utilisé pour filtrer les résultats HTML bruts issus du Web scraping.
4. Module **time**, fonction *sleep* : Module fournissant différentes fonctions liées au temps. Dans le script, la fonction "sleep(t)" est utilisée pour forcer le système à ne rien faire pendant t secondes, évitant de cette façon de surcharcher le serveur cible de requêtes trop rapides et trop nombreuses.

2.3.2 Text Mining avec R

Pour réaliser le text-mining, j'ai utilisé un script R (développé sous Rstudio en Rmd) pour traiter les textes et l'analyser sous forme de figures. Mon travail a donc consisté à adapter les codes R montrés en exemple sur des livres à mes propres données (abstracts d'articles scientifiques), structurées différemment. Les analyses portaient par exemple sur la fréquence des mots, au global et pour chaque MA, ou encore sur l'analyse de sentiment reposant sur l'attribution d'un score positif (+1) ou négatif (-1) à chaque mot du corpus. Cette attribution de sentiment a pu être réalisée grâce à un dictionnaire R conçu pour relier un token donné à la valence (positive ou négative) qui lui correspond. Afin de mieux visualiser les résultats, différentes figures ont été réalisées, comme des diagrammes en barre ou des nuages de mots. Le script R développé repose sur les librairies suivantes :

Librairie dplyr : Librairie R conçue pour faciliter la manipulation de larges jeux de données (DataFrame et Tibble), avec des fonctions spacialisées comme *mutate* (ajout de variables), *select* (sélectionner les variables à partir de leurs noms), *filter* (filtrer des cellules selon leur valeur), *summarise* (résumer les informations d'un tibble dans un format très synthétique) et *arrange* (pour réordonner les lignes selon l'ordre / la variable voulue.)

Librairie ggplot2 : Librairie R pour créer déclarativement des graphiques divers et variés (barplots, histogrammes, scatter plots, etc.)

Librairie tidytext : Librairie R conçue pour faciliter l'analyse de texte (Silge, Julia, and David Robinson. 2016. "tidytext : Text Mining and Analysis Using Tidy Data Principles in R." [2]), se fondant sur le paradigme d'analyse de données "tidy", où chaque variable est une colonne, chaque observation une ligne et chaque ensemble d'observations est un tableau. La fonction la plus utilisée dans le cadre de cette analyse est *unnest_tokens*, qui permet de transformer un texte donné en une tables d'unités textuelles plus réduites (tokens), comme des phrases ou des mots.

Librairie Knitr : Librairie R conçue pour récupérer automatiquement l'output d'un code R (par exemple, pour produire une figure) afin de l'inclure dans un autre document (par exemple, format Word, HTML ou PDF) qui contiendra aussi la prose écrite par l'auteur du document, souvent pour interpréter ou commenter des résultats. Cette librairie permet notamment de moduler plus finement l'affichage des résultats, en permettant par exemple de ne pas afficher certaines figures dans un format de sortie donné (pour masquer une figure interactive au format HTML que l'on ne souhaite pas forcément voir apparaître dans un document PDF, par exemple).

2. ref Silge/Robinson à mettre à la place

Librairie SnowballC : Librairie R implémentant Snowball, un langage conçu pour gérer les chaînes de caractères, les nombres entiers et les booléens. Dans le script R de Text Mining, il a servi à transformer les tokens pour ne garder que la racine de chaque mot (processus de *racination*), afin d'éviter les erreurs de comptage (si, pour un humain, "enhance" et "enhancement" sont deux mots ayant approximativement le même sens, informatiquement ce sont deux chaînes de caractères distinctes).

Librairie grid : Librarie R implémentant les fonctions graphiques primitivesqui sous-tendent le package ggplot2. Elles permettent de modifier certains détails des graphes produits grâce à ggplot2, offrant ainsi un meilleur contrôle du rendu visuel du résulat obtenu. Dans le script, il est employé, par exemple, pour spécifier exactement l'aspect des flèches composant le réseau de bigrammes.

Librairie ggraph : Librairie R formant une extension de ggplot2, conçue pour permettre de supporter les structures de données relationnelles comme les réseaux, les graphes et les arbres. Dans le script, elle est notamment employée pour produire les réseaux de bigrammes.

Librairie igraph : fonction *graph_from_data_frame() :* Cette fonction appartient à la librairie igraph. Elle crée un objet graphe à partir d'un data frame, où le data frame représente les arêtes entre les nœuds.

Librairie egg : fonction *ggarrange() :* Librairie servant à organiser différents graphes sur une seule et même figure. Dans le script, c'est l'une des librairies employées pour comparer les quatre MA entre elles.

Librairie tidyr : Librairie R implémentant des méthodes utiles pour manipuler des objets de type "tidy". Elle comprend des fonction telles que *pivot_wider* et *pivot_longer* (pour convertir le dataFrame d'un format à un autre), ou encore *bind_rows()*, pour combiner des DataFrames par lignes.

2.4 Données

L'approche "tidy" choisie repose sur la *tokenisation* du corpus de texte en unitées textuelles (appelées *"tokens"*) plus petites, comme des mots, des phrases ou des *n-grams*. J'ai pour cela pu m'inspirer du livre rédigé par Julia Slige (data scientist) et David Robinson (Directeur de Data Scientist de la plateforme Heap). [3] Afin de filtrer les mots d'intérêt seulement, deux stratégies ont été employées : Premièrement, un filtrage brut de tous les mots de liaisons sans rapport direct avec le sujet (nommés "stop words" en text-mining, des mots tels que "the", "and", "is", "of" etc. en anglais) pour ne conserver que les mots sur lesquels les analyses pourront donner des résultats scientifiques significatifs (savoir que le mot "le" est le plus fréquent dans un corpus de texte français ne signifie rien sur le plan biologique). Deuxièmement, une autre approche a été de considérer la fréquence de chaque mot par rapport au nombre total de mots présents dans le corpus (n mot / N mots). De cette façon, les mots les plus fréquents perdent une partie de leur poids statistique, tandis que les mots plus rares en gagnent. On peut ainsi visualiser facilement les mots les plus importants d'un texte, sans même avoir besoin de modifier les données au préalable avec une liste de stop words. Cela permet de conserver le texte dans son ensemble, évitant ainsi un potentiel biais pouvant perturber l'analyse.

3. disponible en ligne à l'adresse : https ://www.tidytextmining.com/.

Chapitre 3

Résultats

3.1 Web-scraping et obtention d'une base de métadonnées en format CSV

En sortie du code de Web-scraping dévéloppé en Python (et après recherche manuelle des données manquantes), un fichier de métadonnées au format CSV, contenant la métaanalyse d'origine, le titre, les auteurs, l'abstract, la date de publication sur ResearchGate, le DOI et l'URL vers la page RG correspondante a été obtenue (tableau 3.1).

MA	**Title**	**First author**	**Last author**	**Abstract**	**Date**	**DOI**	**URL**
MA1	Influence of exotic earthworm invasion on soil organic matter, microbial biomass and denitrification potential in forest soils of the northeastern United States.	Amy E Burtelow	Peter M Groffman	Formerly glaciated regions of the northeastern United States have few native earthworm species and the region is dominated by exotic earthworms from Europe and Asia [...].	Sep 1998	10.1016/S0929-1393(98)00075-4	URL

Table 3.1 – Tableau présentant un exemple de la structure type du fichier CSV issu du Web-scrapping et employé pour l'analyse. Le fichier originel comprend 168 lignes. Les données manquantes (non représentées ici), sont notées "NA".

3.2 Analyse globale des textes bruts

Pour détecter informatiquement les racines les plus fréquentes du corpus, chaque token distinct (dans cette section, des mots uniques) présent dans les quatre jeux de données (abstracts d'articles) a été compté. Le graphique ci-dessous présente les racines les plus fréquemment retrouvées dans l'ensemble du corpus. Les trois plus imporantes sont **earthworm**, **soil** et **plant** (figure 3.1).

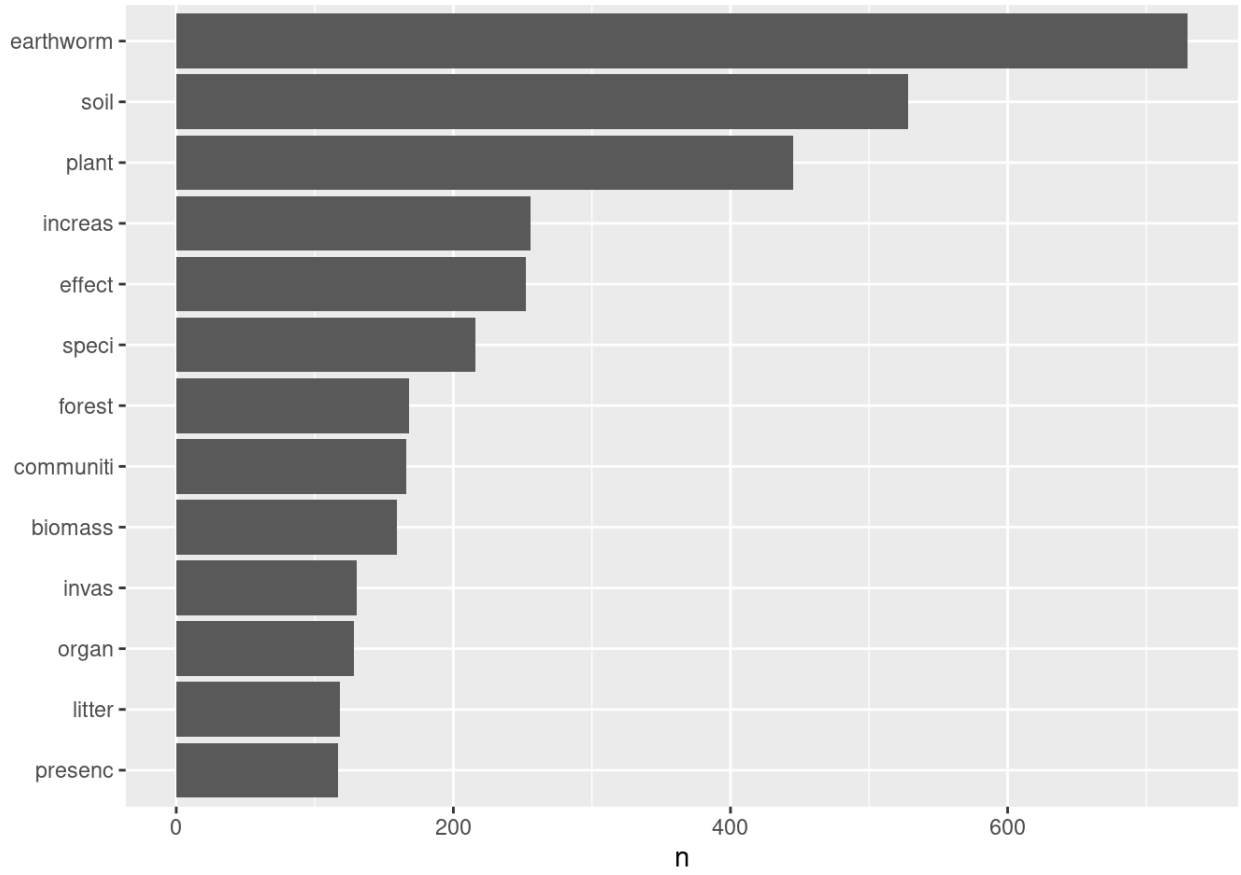

Figure 3.1 – Sur l'axe des abscisses, les valeurs numériques représentent le nombre total d'occurences des mots (après *racination*) écrits en ordonnées. On remarque que les trois mots les plus fréquents dans les articles des quatres métaanalyses confondues sont **earthworm**, **soil** et **plant**.

Dans un second temps, on a conduit la même analyse, mais en considérant cette fois-ci chaque métaanalyse comme un sous jeu de données à part entière (figure 3.2). Les résultats observés pour chaque métaanalyse sont différents des résulats globaux, avec **earthworm**, **soil** et **forest** en premier pour la MA1, **plant**, **earthworm** et **increase** pour la MA2, **earthworm**, **species** et **plant** pour la MA3 et **earthworm**, **plant** et **soil** pour la MA4. Si tous ces groupes ne sont certes pas identiques, on remarque la prévalence de "earthworm" et de "plant", ainsi que la mention de certains écosystèmes ("soil", "forest"). La racine "invas" est aussi présente dans les graphes de fréquence des MA1 et 3.

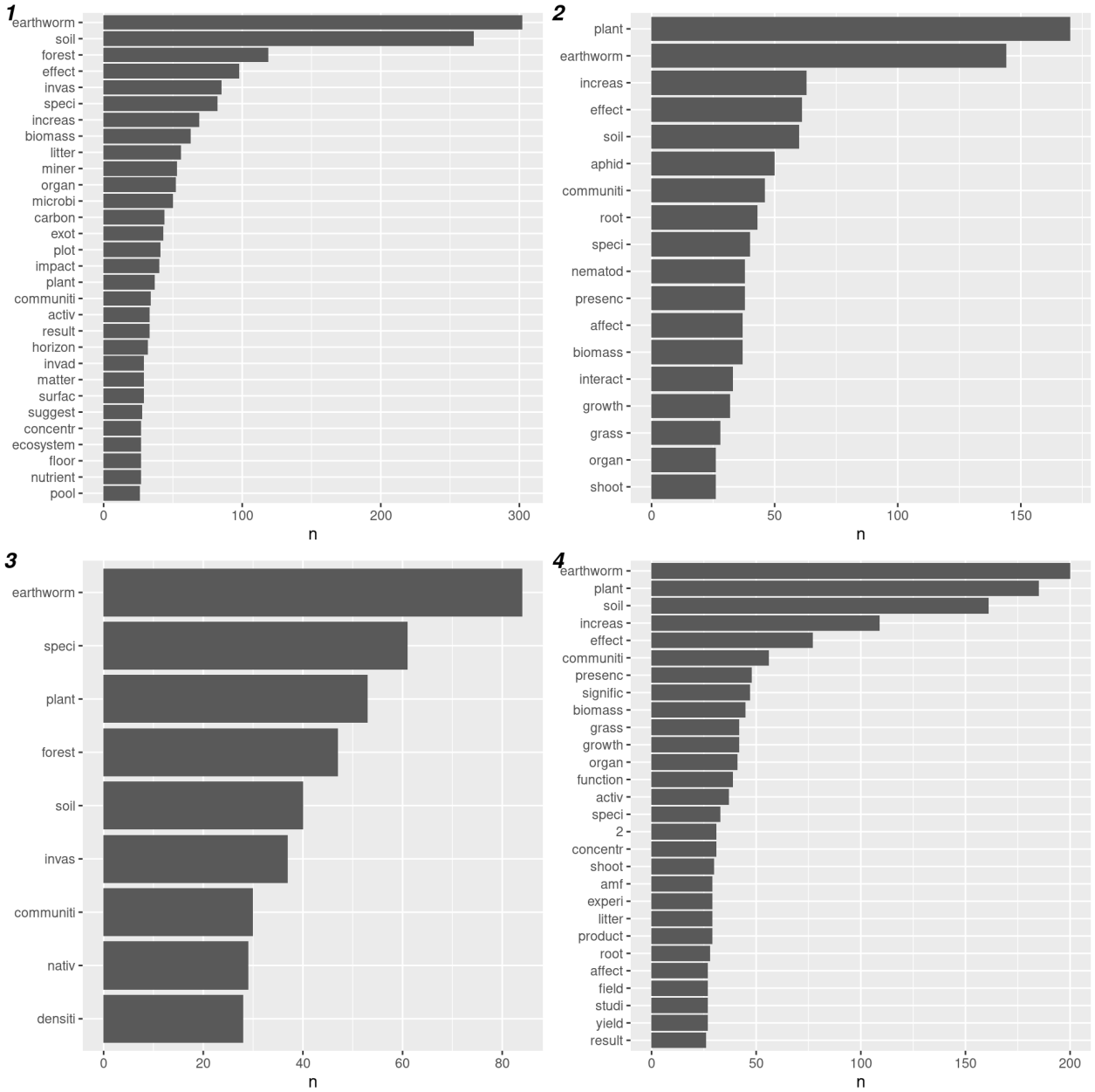

FIGURE 3.2 – Sur l'axe des abscisses, les valeurs numériques représentent le nombre total d'occurences des mots (après *racination*) écrits en ordonnées. On remarque que les résultats individuels sont très différents de ceux obtenus lors de l'analyse collective.

3.3 Approche tf-idf et loi de Zipf

En analyse comparative de texte, il est possible de calculer le **rang** de chaque mot. Dans les tableaux ci-dessous (figure 3.3), on remarque que trois premiers rangs sont toujours occupés par des mots-outils, c'est à dire non sémantiquement pleins. Le mot "earthworm" fait partie des dix premiers de toutes les MA. Le mot "soil" est dans les dix premiers pour les MA1, 3 et 4, là où "plant" est dans les dix premiers pour les MA2, 3 et 4. Pour la MA1, le champ lexical des végétaux est réprésenté par le mot "forest".

```
> freq_and_rank %>% filter(MA=='MA1')
# A tibble: 1,348 × 6
   MA    word          n Total word_frequency  rank
   <chr> <chr>     <int> <int>          <dbl> <int>
 1 MA1   the         488  5616         0.0869     1
 2 MA1   and         456  5616         0.0812     2
 3 MA1   of          425  5616         0.0757     3
 4 MA1   in          310  5616         0.0552     4
 5 MA1   earthworm   302  5616         0.0538     5
 6 MA1   soil        267  5616         0.0475     6
 7 MA1   to          129  5616         0.0230     7
 8 MA1   a           121  5616         0.0215     8
 9 MA1   forest      119  5616         0.0212     9
10 MA1   effect       98  5616         0.0175    10
# i 1,338 more rows
```

```
> freq_and_rank %>% filter(MA=='MA2')
# A tibble: 911 × 6
   MA    word          n Total word_frequency  rank
   <chr> <chr>     <int> <int>          <dbl> <int>
 1 MA2   the         261  2959         0.0882     1
 2 MA2   of          248  2959         0.0838     2
 3 MA2   and         225  2959         0.0760     3
 4 MA2   plant       170  2959         0.0575     4
 5 MA2   in          158  2959         0.0534     5
 6 MA2   earthworm   144  2959         0.0487     6
 7 MA2   by           84  2959         0.0284     7
 8 MA2   a            78  2959         0.0264     8
 9 MA2   on           63  2959         0.0213     9
10 MA2   increas      63  2959         0.0213    10
# i 901 more rows
```

```
> freq_and_rank %>% filter(MA=='MA3')
# A tibble: 474 × 6
   MA    word          n Total word_frequency  rank
   <chr> <chr>     <int> <int>          <dbl> <int>
 1 MA3   the         495  1860          0.266     1
 2 MA3   of          442  1860          0.238     2
 3 MA3   and         362  1860          0.195     3
 4 MA3   in          282  1860          0.152     4
 5 MA3   earthworm   200  1860          0.108     5
 6 MA3   plant       185  1860          0.0995    6
 7 MA3   a           171  1860          0.0919    7
 8 MA3   soil        161  1860          0.0866    8
 9 MA3   to          133  1860          0.0715    9
10 MA3   increas     109  1860          0.0586   10
# i 464 more rows
```

```
> freq_and_rank %>% filter(MA=='MA4')
# A tibble: 1,447 × 6
   MA    word          n Total word_frequency  rank
   <chr> <chr>     <int> <int>          <dbl> <int>
 1 MA4   the         495  4985         0.0993     1
 2 MA4   of          442  4985         0.0887     2
 3 MA4   and         362  4985         0.0726     3
 4 MA4   in          282  4985         0.0566     4
 5 MA4   earthworm   200  4985         0.0401     5
 6 MA4   plant       185  4985         0.0371     6
 7 MA4   a           171  4985         0.0343     7
 8 MA4   soil        161  4985         0.0323     8
 9 MA4   to          133  4985         0.0267     9
10 MA4   increas     109  4985         0.0219    10
# i 1,437 more rows
```

Figure 3.3 – Impression écran issue de Rstudio, montrant les dix mots les plus représentés dans chaque MA. Ces résultats ont été obtenu par filtration d'un tibble R plus large nommé "freq_and_rank". On remarque que les premiers rangs sont souvent occupés par des mots fonctionnels, comme "the", "of", "and" et "in".

L'objectif de l'approche tf-idf est d'évaluer l'importance d'un terme contenu dans un document, relativement à l'ensemble du document. La fréquence brute du terme est nommée tf, et la fréquence inverse de document (une mesure de l'importance du terme dans l'ensemble du corpus) est nommée idf. Le poids ajusté de chaque terme s'obtient en multipliant ces deux mesures, et il augmente proportionnellement au nombre d'occurrences du mot dans le document. Grâce à l'approche tf-idf (figure 3.4), on peut voir que chaque mot présent sur l'axe Oy est un mot utilisé plus souvent que les autres au sein d'une métaanalyse. Les racines **earthworm, soil, forest** sont plus fréquentes dans la MA1, **aphid, nematod, herbivor** plus fréquentes dans la MA2, **eastern, canopy, arthropod** plus fréquentes dans la MA3 et **plfa, pb, cu** plus féquentes dans la MA4.

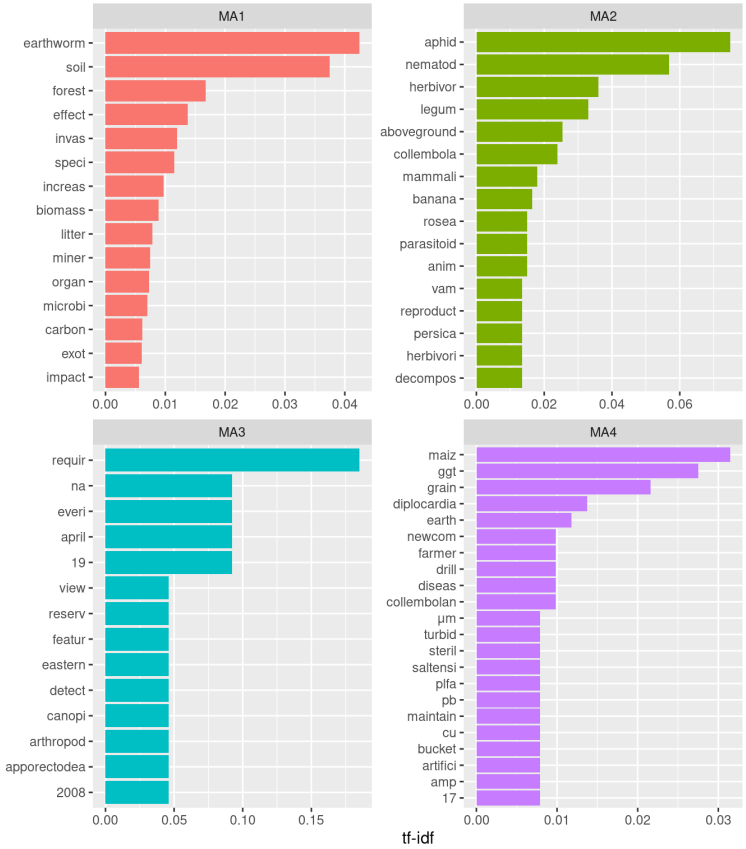

FIGURE 3.4 – Mesures de tf-idf pour chacune des métaanalyses du corpus. Les mots écrits sur l'axe Oy sont plus importants dans leurs MA respectives que dans les autres.

En linguistique, la loi de Zipf (figure 3.5), stipule que $tf \propto \frac{1}{rang}$. La MA3 (courbe grise) contient davantage de mots "rares" que la valeur prédite par le modèle linéaire (quart supérieur gauche), alors que toutes les autres en contiennent moins. La MA3 est donc celle qui contient la plus haute fréquence de mots rares. Pour ce qui est des mots très fréquents (quart inférieur droit sous 0.001), on peut constater que toutes les métaanalyses en contiennent moins que la valeur prédite.

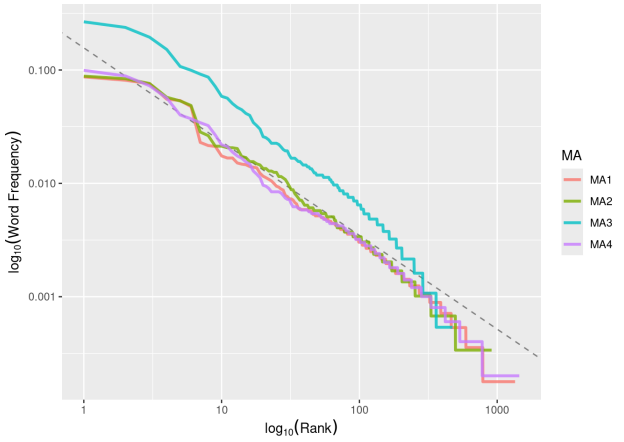

FIGURE 3.5 – La courbe grise en pointillés représente la valeur attendue selon un modèle linéaire sur l'intervalle [11,99]. Sa pente est de -0.8272.

3.4 Analyses de bigrammes seuls et réseaux de bigrammes

Grâce à l'approche tf-idf (figure 3.6), on peut voir que chaque *bigramme* présent sur l'axe Oy est particulièrement important pour la métaanalyse cconcernée. Ainsi, **mineral soil, earthworm invasion, exotic earthworms** sont plus importants que les autres dans la MA1, **soil organisms, plant responses, plant mediated** plus importants que les autres dans la MA2, **earthworm invasions, native earthworm, species richness** plus importants que les autres dans la MA3 et **soil organisms, soil fertility, dry weight** plus importants que les autres dans la MA4.

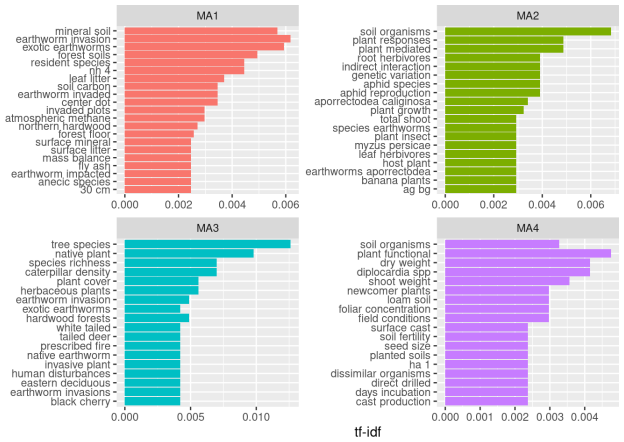

FIGURE 3.6 – Mesures de tf-idf pour chacune des métaanalyses du corpus. Les **groupes de mots** écrits sur l'axe Oy sont plus importants dans leurs MA respectives que dans les autres.

Grâce à l'approche en réseau de bigrammes, il est possible de relever des associations de mots au-delà de deux mots seulement. Le réseau ci-dessous (figure 3.7) montre que les trois mots les plus importants du corpus sont **"plant", "soil"** et **"earthworm"**. Pour le premier, les bigrammes rencontrés sont *"plant community", "plant growth", "plant communities", "native plant"*, pour le second *"mineral soil", "soil microbial"* / *"microbial biomass" (forte association), "soil organisms"* et pour le troisième *"invasion", "species", "activity"*. D'autres bigrammes notables sont : "forest floor", "exotic earthworm", "organic matter", "leaf litter", "northern hardwood" et "lumbricus terretris".

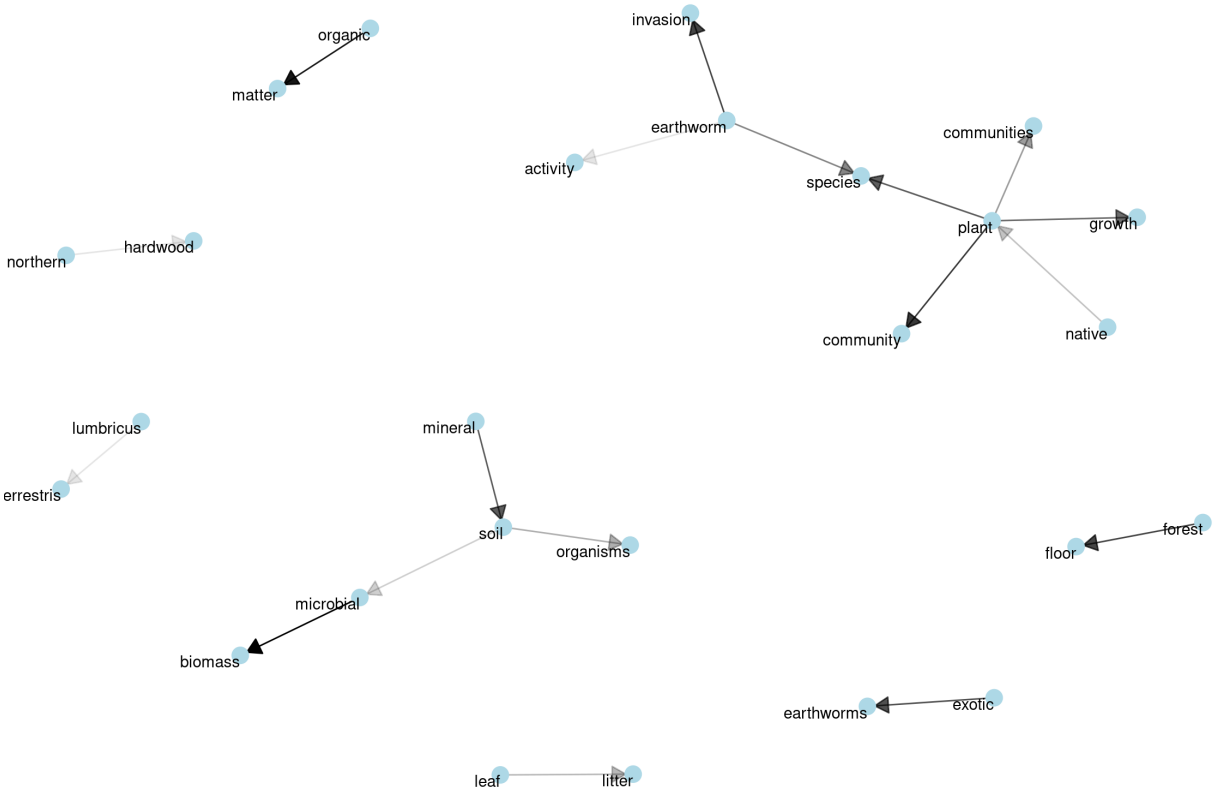

FIGURE 3.7 – Réseau de bigrammmes montrant les connexions les plus fréquentes reliant les mots du corpus. Chaque mot est représenté par un noeud du réseau (**bleu clair**) et les connexions entre les mots sont orientées. L'opacité des flèches est proportionnelle à la fréquence d'occurence du bigramme.

Il peut aussi être intéressant d'analyser l'aspect des réseaux obtenus pour chaucne des métaanalyses individuellement (figures 3.8 à 3.11). Pour la MA1 (figure 3.8), le seul centre retrouvé est **"earthworm"**, impliqué dans les connexions *"earthworm species"* et *"earthworm invasion"*. D'autres bigrammes notables sont : "mineral soil" –> "soil microbial" –> "microbial biomass", ou encore "exotic earthworms", "organic matter", "forest floor", "leaf litter", "northern hardwood", "lumbricus terrestris"

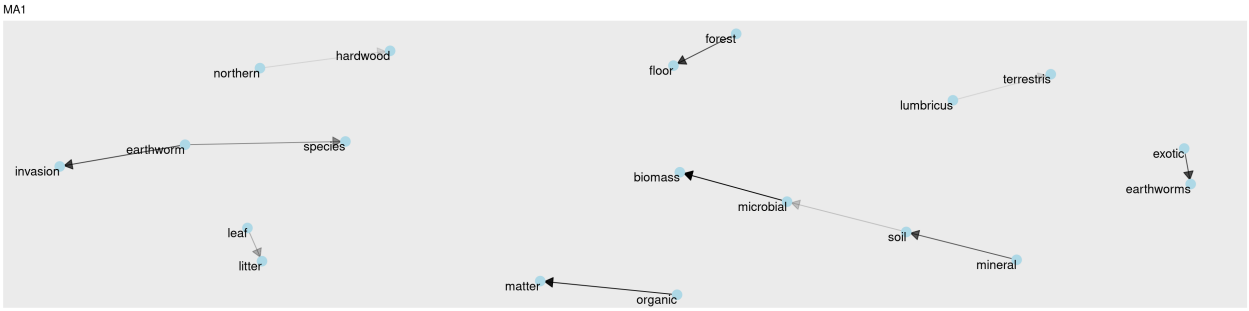

Figure 3.8 – Réseau de bigrammmes montrant les connexions les plus fréquentes reliant les mots de la MA1. Chaque mot est représenté par un noeud du réseau (**bleu clair**) et les connexions entre les mots sont orientées. L'opacité des flèches est proportionnelle à la fréquence d'occurence du bigramme.

Pour la MA2 (figure 3.9), le seul centre retrouvé est **"plant"**, impliqué dans les connexions *"plant species"*, *"plant growth"* et *"plant communities"*. "Soil organisms" est aussi un bigramme notable.

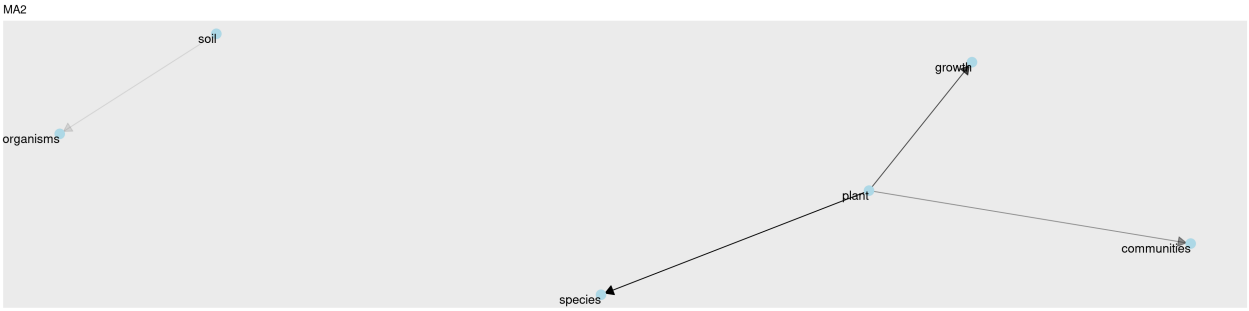

Figure 3.9 – Réseau de bigrammmes montrant les connexions les plus fréquentes reliant les mots de la MA2. Chaque mot est représenté par un noeud du réseau (**bleu clair**) et les connexions entre les mots sont orientées. L'opacité des flèches est proportionnelle à la fréquence d'occurence du bigramme.

Pour la MA3 (figure 3.10), aucun centre n'est retrouvé. Par contre, ce réseau est constitué d'une chaîne de mots qui se suivent : “native plants” –> “plant species” –> “earthworm species”.

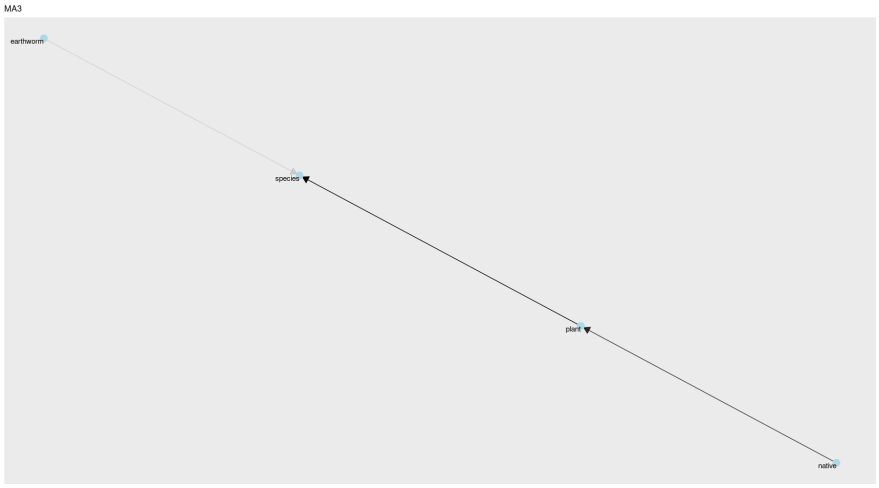

Figure 3.10 – Réseau de bigrammmes montrant les connexions les plus fréquentes reliant les mots de la MA3. Chaque mot est représenté par un noeud du réseau (**bleu clair**) et les connexions entre les mots sont orientées. L'opacité des flèches est proportionnelle à la fréquence d'occurence du bigramme.

Pour la MA4 (figure 3.11), le seul centre retrouvé est **"plant"**, impliqué dans les connexions *“plant species”, “plant growth”* et *“plant communities”*. "Soil organisms" et "earthworm activity" sont aussi des bigrammes notables.

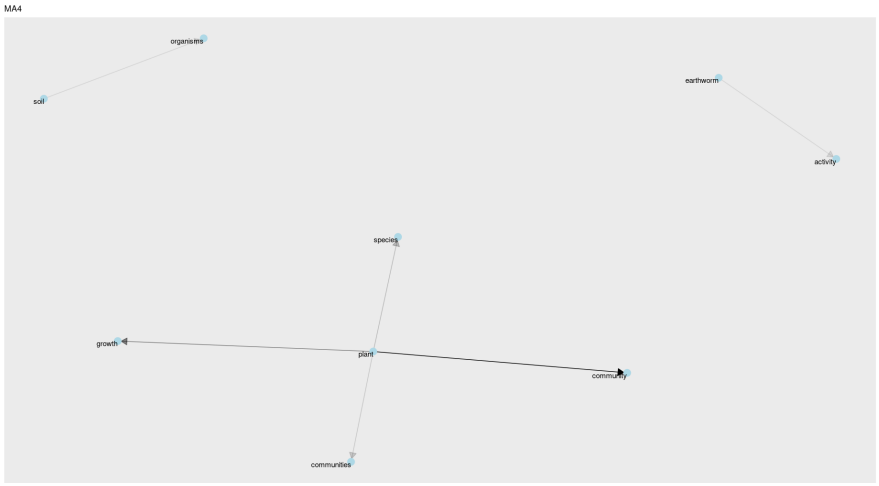

Figure 3.11 – Réseau de bigrammmes montrant les connexions les plus fréquentes reliant les mots de la MA4. Chaque mot est représenté par un noeud du réseau (**bleu clair**) et les connexions entre les mots sont orientées. L'opacité des flèches est proportionnelle à la fréquence d'occurence du bigramme.

3.5 Analyse de sentiment

L'objectif de cette série d'analyses est de détecter informatiquement l'avis émis par les abstracts composant le corpus de texte à propos des vers de terre. Pour cela, on s'appuie sur des dictionnaires R prédéfinis faisant correspondre à chaque mot un score chiffré, +1

si le mot est positif, -1 sinon. La visualisation sous forme de nuage de mots (figure 3.12) permet de constater que les racines "invasiv", "invad" et "effect", "affect" ont été les plus importantes contributrices au score de sentiment global. Les racines "loss" et "negat" ont aussi une grande contribution.

FIGURE 3.12 – Nuage de mots de l'ensemble des racines de mots trouvées dans la base de données. Les mots exprimant un sentiment négatif sont en **rouge**, les mots exprimant un sentiment positif en **bleu**.

Une analyse plus détaillée de chaque contributeur (en ordonnée) a pu être mise en place pour chaque métaanalyse (figure 3.13). On peut voir que les métaanalyses 1 et 3 semblent globalement négatives, avec des racines comme "invasive", "invade" ou "negative" (encadrés 1 et 3), tandis que les métaanalyses 2 et 4 semblent globalement positives avec des mots comme "enhance" ou "competitive". (encadrés 2 et 4).

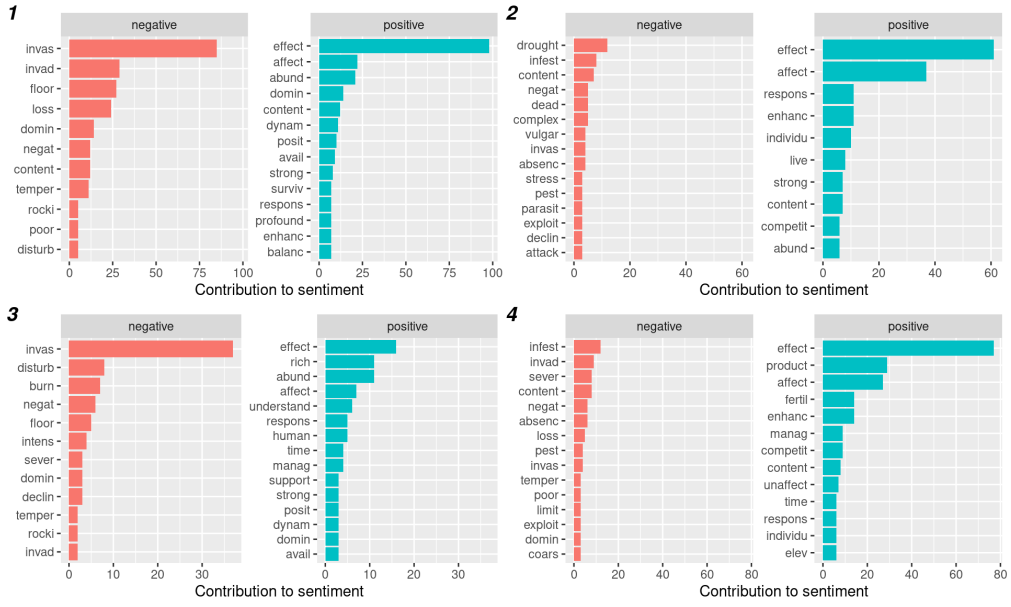

FIGURE 3.13 – Mots contribuant au score de sentiment pour chacune des quatre métaanalyses. Les mots exprimant un sentiment négatif sont en **rouge**, les mots exprimant un sentiment positif en **bleu**. Le nombre de contributeurs différents est visible sur l'axe Oy de chaque graphique.

Chapitre 4

Discussion

Pour la constitution de la base de données de travail (le fichier CSV contenant les abstracts des articles à analyser), une approche semi-automatique a pu être implémentée, mais il serait encore nécessaire d'améliorer l'outil pour pouvoir complètement automatiser le processus. Cependant, de nombreux sites internet combattant activement le Web-scraping, il pourrait s'avérer difficile d'obtenir un script parfaitement fonctionnel et stable dans le temps. Les abstracts d'articles se sont avérés étonnament difficiles à récupérer automatiquement, c'est pourquoi une intervention manuelle a souvent été nécessaire.

Une fois les abstracts récupérés, une première analyse de la fréquence brute des termes à montré que les mots les plus retrouvés dans le corpus entier étaient "earthworm", "soil" et "plant", ce qui semble confirmer que tous ces textes traitaient bel et bien de l'influence des vers de terre sur le sol et les plantes. (chapitre 3, section 3.2). Individuellement, il ressort que la MA1 étudie davantage l'impact des vers de terre invasifs sur les sols forestiers, là où la MA3 se focalise sur la canopée des forêts et l'influence des espèces venues de l'est. La MA2 s'intéresse à l'influence des organismes du sol sur la physiologie des plantes, là où la MA4 semble davantage étudier leur influence la productivité des plantes.

Chapitre 5

Conclusion

Elle vise, à reformuler les objectifs visés, énoncer les résultats essentiels obtenus, à replacer le travail dans son contexte scientifique et à faire ressortir leur importance théorique, pratique, tech- nique ou économique. Elle peut ouvrir de nouvelles perspectives ou hypothèses qui seront le point de départ de nouveaux travaux. Il n'y a pas a priori d'appel à des références.

Bibliographie

[1] Michel Bertrand, Sébastien Barot, Manuel Blouin, Joann Whalen, Tatiana de Oliveira, and Jean Roger-Estrade. Earthworm services for cropping systems. a review. *Agronomy for Sustainable Development*, 35(2) :553–567, January 2015.

[2] Patrick J. Bohlen, Stefan Scheu, Cindy M. Hale, Mary Ann McLean, Sonja Migge, Peter M. Groffman, and Dennis Parkinson. Non-native invasive earthworms as agents of change in northern temperate forests. *Frontiers in Ecology and the Environment*, 2(8) :427–435, October 2004.

[3] Clive A. Edwards and Norman Q. Arancon. *The Role of Earthworms in Organic Matter and Nutrient Cycles*, page 233–274. Springer US, 2022.

[4] Olga Ferlian, Nico Eisenhauer, Martin Aguirrebengoa, Mariama Camara, Irene Ramirez-Rojas, Fábio Santos, Krizler Tanalgo, and Madhav P. Thakur. Invasive earthworms erode soil biodiversity : A meta-analysis. *Journal of Animal Ecology*, 87(1) :162–172, September 2017.

[5] Oswaldo Forey, Joana Sauze, Clément Piel, Emmanuel S. Gritti, Sébastien Devidal, Abdelaziz Faez, Olivier Ravel, Johanne Nahmani, Laly Rouch, Manuel Blouin, Guénola Pérès, Yvan Capowiez, Jacques Roy, and Alexandru Milcu. Earthworms do not increase greenhouse gas emissions (co2 and n2o) in an ecotron experiment simulating a three-crop rotation system. *Scientific Reports*, 13(1), December 2023.

[6] Young-Nam Kim, Brett Robinson, Keum-Ah Lee, Stephane Boyer, and Nicholas Dickinson. Interactions between earthworm burrowing, growth of a leguminous shrub and nitrogen cycling in a former agricultural soil. *Applied Soil Ecology*, 110 :79–87, February 2017.

[7] Rahul Kumar, Renu Yadav, Rajender Kumar Gupta, Kiran Yodha, Sudhir Kumar Kataria, Pooja Kadyan, Pooja Sharma, and Simran Kaur. The earthworms : Charles darwin's ecosystem engineer. In Khalid Rehman Hakeem, editor, *Organic Fertilizers*, chapter 13. IntechOpen, Rijeka, 2023.

[8] Aboulkacem Lemtiri, Gilles Colinet, Taofic Alabi, Daniel Cluzeau, Lara Zirbes, Eric Haubruge, and Frédéric Francis. Impacts of earthworms on soil components and dynamics. a review. *Biotechnologie, Agronomie, Société et Environnement / Biotechnology, Agronomy, Society and Environment*, November 2014.

[9] Scott R. Loss, Ryan M. Hueffmeier, Cindy M. Hale, George E. Host, Gerald Sjerven, and Lee E. Frelich. Earthworm invasions in northern hardwood forests : a rapid assessment method. *Natural Areas Journal*, 33(1) :21–30, January 2013.

[10] Ingrid M Lubbers, Kees Jan Van Groenigen, Steven J Fonte, Johan Six, Lijbert Brussaard, and Jan Willem Van Groenigen. Greenhouse-gas emissions from soils increased by earthworms. *Nature Climate Change*, 3(3) :187–194, 2013.

[11] D. K. Sharma, S. Tomar, and D. Chakraborty. Role of earthworm in improving soil structure and functioning. *Current Science*, 113(6) :1064–1071, 2017.

Résumé

à faire à la fin

Mots-clés de référencement type MESH : Lumbricus terrestris, Soil, Ecosystem, Introduced species, Meta-Analysis.

Mots-clés des acquis techniques : Web-scraping, Text-mining, dplyr, R Markdown, GitHub.