جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITE SIDI MOHAMED BEN ABDELLAH DE FES
كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MAHRAZ FES

UNIVERSITÉ SORBONNE PARIS NORD

Mémoire présenté à
La Faculté des Sciences Dhar El Mahraz Fès
Pour l'obtention du Diplôme de Master

Web Intelligence et Science des Données (WISD)

Master en double diplomation avec l'Université Sorbonne Paris Nord

Spécialité : Informatique

Normalisation automatique de variables issues de bases de données en agroécologie

---

Réalisé par :
Oussama MECHHOUR

Encadré par :
Mme. Sandrine AUZOUX
M. Mathieu ROCHE
M. Benjamin HEUCLIN
M. Ismail EL BATTEOUI

Soutenu le 13/07/2023, *Devant le jury composé de :*

Mme. Sandrine AUZOUX, CIRAD
M. Ismail EL BATTEOUI, Faculté des Sciences Dhar El Mahraz
M. Ali YAHYAOUY, Faculté des Sciences Dhar El Mahraz
M. Badraddine Aghoutane, Faculté des Sciences de Meknès
M. Abdelouahed Sabri, Faculté des Sciences Dhar El Mahraz
M. Mohammed Adnane Mahrez, Faculté des Sciences Dhar El Mahraz

Année universitaire 2022-2023

Table des matières

Liste des tableaux

Table des figures

Liste des abréviations

#DigitAg : Digital Agriculture Convergence Lab

AEGIS : Agroecological Global Information System

AIDA : Agroécologie et Intensification Durable des cultures Annuelles

BERT : Bidirectional Encoder Representations from Transformers

CIRAD : Centre de coopération internationale en recherche agronomique pour le développement

CSV : Comma-Separated Values

EID² : Exploration Informatique des Données et Décisionnel

IRD : Institut de Recherche pour le Développement

Irstea : Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture

LSTM : Long Short-Term Memory

MLM : Masked Language Model

NSP : Next Sentence Prediction

PDF : Portable Document Format

RNN : Recurrent Neural Network

RoBERTa : Robustly Optimized BERT Pretraining Approach

STICS : Simulateur mulTIdisciplinaire pour les Cultures Standard

TETIS : Technologies, Environnement, Territoires et Sociétés

TF-IDF : Term Frequency-Inverse Document Frequency

UMR : Unité Mixte de Recherche

UP13 : Université Paris 13

USMBA : Université Sidi Mohamed Ben Abdellah

WISD : Web Intelligence et Science des Données

XLNet : eXtreme Language understanding Network

Glossaire

Agroécologie : Une approche holistique de l'agriculture qui intègre des principes écologiques, économiques et sociaux pour développer des systèmes agricoles durables. 12–14

Analyse de données : Le processus d'examiner, d'interpréter et d'extraire des informations significatives à partir de jeux de données afin de prendre des décisions éclairées. 12

Apprentissage automatique (machine learning) : est une branche de l'Intelligence artificielle qui se concentre sur le développement de méthodes et d'algorithmes permettant aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sur une tâche spécifique, sans être explicitement programmés. L'apprentissage automatique est largement utilisé en agroécologie pour analyser et interpréter les données agricoles, telles que les rendements des cultures, les conditions météorologiques, les données du sol, etc. Il existe différentes approches d'apprentissage automatique, dont l'apprentissage supervisé, non supervisé et par renforcement. L'apprentissage automatique est souvent combiné avec l'Apprentissage profond (deep learning). 8

Apprentissage profond (deep learning) : est une sous-catégorie de l'apprentissage automatique qui utilise des réseaux de neurones profonds pour apprendre et modéliser des données complexes. Les réseaux de neurones profonds sont constitués de plusieurs couches de neurones interconnectés, qui permettent d'apprendre des représentations hiérarchiques des données. L'apprentissage profond est particulièrement efficace pour traiter des données non structurées telles que des images, des textes ou des signaux audio. En agroécologie, l'apprentissage profond est utilisé pour diverses tâches, telles que la classification d'images de cultures, la prédiction des rendements agricoles, l'analyse des maladies des plantes, etc. L'apprentissage profond a montré des performances impressionnantes dans de nombreux domaines et continue d'être un sujet de recherche actif dans le domaine de l'agroécologie. 7

Base de données : Une collection organisée de données structurées et interconnectées, stockées de manière persistante pour faciliter l'accès, la gestion et la manipulation des informations. Les bases de données sont largement utilisées dans les applications informatiques pour stocker, gérer et récupérer des données de manière efficace et sécurisée. Elles sont essentielles dans de nombreux domaines tels que les systèmes d'information, les applications web, les systèmes de gestion de contenu, etc.. 16

BERT : Un modèle de traitement du langage naturel basé sur des Transformers, largement utilisé pour des tâches telles que la compréhension du langage, la traduction automatique et la génération de texte. 14, 28

Biostatistiques : Une branche des statistiques qui s'applique spécifiquement à l'analyse de données biologiques et médicales. Les biostatistiques combinent les principes statistiques avec les connaissances du domaine de la biologie ou de la médecine pour interpréter et tirer des conclusions à partir des données collectées dans ces domaines. Elles jouent un rôle essentiel dans la conception d'études cliniques, l'analyse de données génétiques, l'évaluation des risques sanitaires, la modélisation épidémiologique, etc. Les méthodes biostatistiques permettent de prendre des décisions éclairées basées sur des preuves statistiques solides dans le domaine des sciences de la vie. 16

Canonisation : Le processus de normalisation ou d'uniformisation des variables consiste à les rendre conformes à une forme standardisée afin de faciliter leur correspondance. Dans notre cas, cela signifie transformer des données non canoniques (non formelles et/ou non structurées) en données canoniques (formelles et/ou structurées). 14

Corpus : Une collection de textes ou de données utilisée comme base de référence pour l'analyse et la recherche dans le domaine de l'agroécologie. 14

Correspondance : Le Processus d'alignement et de mise en correspondance des variables utilisées par les chercheurs en agroécologie. 14

Cosinus : Une mesure de similarité entre deux vecteurs utilisée pour évaluer la proximité entre les variables dans le contexte de correspondance des variables. 14

Distance de Levenshtein : Une mesure qui calcule le nombre minimum d'opérations nécessaires pour transformer une chaîne de caractères en une autre. 14

Données agricoles : Les informations recueillies sur les pratiques agricoles, les cultures, les rendements, les ressources naturelles, etc., utilisées dans l'étude et l'analyse des systèmes agricoles. 14

Données multilingues et hétérogènes : Un ensemble de données qui comprend des documents dans plusieurs langues différentes, ainsi que des données provenant de sources variées et présentant des caractéristiques différentes. Ces données peuvent inclure des textes, des images, des vidéos, etc.. 14

Développement de systèmes agricoles durables : L'objectif de concevoir et de mettre en place des systèmes agricoles qui préservent l'environnement, soutiennent la productivité à long terme, améliorent les conditions de vie des agriculteurs et contribuent au bien-être des communautés agricoles. 14

Fine-tuning (ajustement fin) : est une technique largement utilisée dans les domaines de l'Apprentissage automatique (machine learning) et du traitement du langage naturel. Son principe consiste à prendre un modèle de langue pré-entraîné, tel que BERT, et à le ré-entraîner en utilisant des données spécifiques à une tâche donnée, comme l'agroécologie. Grâce au fine-tuning, le modèle pré-entraîné peut être adapté aux caractéristiques et aux spécificités de la tâche cible en ajustant ses poids et ses paramètres. Cette approche permet d'améliorer les performances du modèle dans la tâche spécifique en exploitant l'expertise et les connaissances préalablement acquises lors de l'entraînement initial. Le fine-tuning joue un rôle essentiel dans de nombreux projets d'apprentissage automatique, car il permet de tirer parti des avantages des modèles pré-entraînés tout en les adaptant aux besoins spécifiques de la tâche à accomplir. 9

Informatique : La science du traitement de l'information par des moyens automatiques, notamment par des ordinateurs. 16

Intelligence artificielle : est une branche de l'informatique qui vise à développer des systèmes capables de simuler des comportements intelligents. L'IA se concentre sur la création de machines capables de percevoir leur environnement, de raisonner, d'apprendre et de prendre des décisions de manière autonome. Elle englobe un large éventail de techniques, telles que l'apprentissage automatique, le traitement automatique du langage naturel, la vision par ordinateur, les réseaux de neurones artificiels, etc. En agroécologie, l'intelligence artificielle est utilisée pour analyser et interpréter les données agricoles, optimiser les systèmes de culture, prédire les rendements des cultures, identifier les maladies des plantes, etc. L'intelligence artificielle continue de progresser rapidement et offre des possibilités prometteuses pour améliorer l'efficacité et la durabilité des systèmes agricoles. 7

Masquage de mots : est une tâche d'entraînement utilisée dans le modèle BERT. Dans le MLM, certaines parties du texte d'entrée sont masquées, et le modèle doit prédire les mots manquants en se basant sur le contexte environnant. Cela permet à BERT d'apprendre des représentations de mots contextualisées qui capturent les relations entre les mots dans une phrase ou un document[17]. 32

Mesure en agroécologie : fait référence à une variable, un indicateur ou un paramètre utilisé pour évaluer et quantifier un aspect environnemental, social ou économique d'un système agricole. Cette mesure vise à évaluer la durabilité, la performance et les impacts des pratiques agricoles sur l'écosystème, la biodiversité, la santé humaine, la qualité des sols, l'utilisation des ressources, etc. La mesure en agroécologie peut inclure des indicateurs tels que la consommation d'eau, l'émission de gaz à effet de serre, la diversité des cultures, l'efficacité énergétique, l'utilisation d'intrants chimiques, la productivité agricole, les revenus des agriculteurs, etc. Cette mesure est essentielle pour évaluer les performances des systèmes agricoles durables et orienter les décisions et les actions visant à promouvoir une agriculture plus respectueuse de l'environnement et socialement équitable. 10

Modèle de langues pré-entraîné : est un modèle de traitement automatique du langage naturel (TALN) qui a été entraîné sur de grandes quantités de données textuelles non étiquetées. Il apprend les relations et les structures linguistiques à partir des données brutes, ce qui lui permet de capturer des informations sémantiques et syntaxiques utiles pour des tâches spécifiques, telles que la traduction automatique, la génération de texte ou la compréhension de la langue naturelle. Les modèles de langage pré-entraînés servent souvent de base pour des tâches de TALN ultérieures en Fine-tuning (ajustement fin). 30

Modèle STICS : est un modèle dynamique, générique et robuste permettant de simuler le système sol-atmosphère-culture. 10

Méthodes de plongement de mots : Des techniques utilisées dans le traitement automatique du langage naturel pour représenter les mots sous forme de vecteurs numériques denses. Ces vecteurs capturent les similarités sémantiques et syntaxiques entre les mots, ce qui permet d'effectuer des opérations mathématiques sur les mots et d'effectuer des tâches telles que la recherche de similarité, la classification de texte, etc. Des exemples de méthodes de plongement de mots populaires sont Word2Vec, GloVe et FastText. 14

Ontologies : Des structures de connaissances formelles qui représentent les concepts, les relations et les propriétés d'un domaine spécifique. Elles facilitent la compréhension et l'interopérabilité des données dans le domaine de l'agroécologie. 14

Permutations de mots : sont une technique utilisée dans le modèle XLNet pour capturer les relations de dépendance entre les mots d'une phrase. Au lieu de fixer un ordre linéaire des mots, XLNet considère toutes les permutations possibles et les traite comme des instances d'entraînement distinctes. Cela permet au modèle d'apprendre des représentations qui capturent les dépendances contextuelles entre les mots, même en présence d'ordres différents. Cette approche permet à XLNet d'obtenir de bonnes performances sur des tâches de compréhension du langage naturel [19]. 32

Processus d'alignement : La mise en correspondance de deux éléments ou plus afin de les rendre cohérents ou de les associer d'une manière spécifique. 8

Précision à la position 10 : Dans ce rapport, elle représente la probabilité que la solution pertinente, c'est-à-dire la variable candidate qui correspond réellement à la variable source, soit parmi les **10 premières** variables candidates proposées. 41

Similarité : Une Mesure en agroécologie qui évalue à quel point deux variables ou documents sont similaires ou proches les uns des autres. 14

Text Mining : L'exploration et l'analyse automatique de grands ensembles de données textuelles afin d'en extraire des informations significatives et utiles. 12, 13

TF-IDF : Une mesure utilisée pour évaluer l'importance d'un terme dans un document en tenant compte de sa fréquence et de sa rareté dans un corpus de documents. 14

Traitement automatique du langage naturel : Un domaine de l'informatique qui vise à permettre aux ordinateurs de comprendre, d'analyser et de générer le langage humain de manière automatique. 14

Transformers : sont une architecture de réseau de neurones révolutionnaire utilisée dans le domaine du traitement automatique du langage naturel (TALN). Ils ont été introduits par Vaswani et al. en 2017 et ont connu un grand succès dans de nombreuses tâches de TALN, notamment la traduction automatique, la génération de texte, la classification de texte, etc. Les Transformers se distinguent des architectures précédentes par leur utilisation de l'attention pour capturer les relations entre les mots ou les parties du texte, permettant ainsi de modéliser des dépendances à longue distance et d'obtenir de meilleures performances. Ils ont ouvert la voie à des modèles de langues pré-entraînés tels que BERT et RoBERTa, qui ont révolutionné le domaine du TALN. 7

Variables : Ce sont des entités qui représentent des caractéristiques, des attributs ou des paramètres mesurables dans un contexte spécifique. Elles peuvent prendre la forme de valeurs numériques, de catégories, d'états ou de chaînes de caractères. Dans le contexte de ce rapport, ces variables sont représentées sous forme de chaînes de caractères. 10, 11

Variables candidates : Ces Variables, composées de termes sémantiques issus de connaissances expertes et d'ontologies de référence, ont été spécifiquement définies dans le but de faciliter la comparaison et l'analyse des données, ainsi que d'établir des liens avec des modèles de culture tels que **Modèle STICS**. 16

Variables sources : Ce sont des Variables issues des travaux de recherche dans le domaine de l'agroécologie, et pour ce stage, le travail s'est focalisé sur des données spécifiques liées à la culture de la canne à sucre. 16

Dédicace

Je dédie ce rapport de stage à mes parents et à mon frère, pour leur soutien, leur encouragement et leur aide inconditionnels. Leurs présences et leur soutien indéfectible ont été d'une importance capitale tout au long de ce parcours. Je suis infiniment reconnaissant envers eux et je souhaite que Dieu les préserve et les entoure de sa protection pour qu'ils puissent assister à ma soutenance de thèse et me voir évoluer vers de meilleurs postes professionnels.

Je tiens à exprimer ma profonde gratitude envers mes encadrants, **M. Mathieu ROCHE**, **M. Benjamin HEUCLIN**, **Mme. Sandrine AUZOUX** et **M. Ismail EL BATTEOUI**, pour leur soutien constant, leurs encouragements, leurs précieuses remarques et leur aide précieuse. Leurs connaissances approfondies dans les domaines de l'Agroécologie, de l'Analyse de données et du Text Mining m'ont permis d'approfondir mes connaissances et de progresser dans ma formation. Leur accompagnement a été essentiel pour la réussite de ce stage.

Je souhaite également exprimer ma gratitude envers **USMBA** et **UP13** pour les programmes de masters **WISD** et **EID²** qui m'ont offert des opportunités exceptionnelles pour réaliser mes rêves. Je remercie particulièrement les coordinateurs des deux masters, **M. Ali YAHYAOUY** et **M. Younès BENNANI MEZIANE**, ainsi que tous les enseignants de notre master et de notre université, pour leur enseignement de qualité et leur contribution à ma formation.

Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont joué un rôle crucial dans la réussite de mon stage M2. Leur soutien inestimable, leurs conseils avisés et leurs encouragements constants ont été des facteurs déterminants tout au long de ce parcours.

En premier lieu, un grand merci à **Mme. Sandrine AUZOUX** pour sa précieuse assistance dans la compréhension des données d'Agroécologie. Ses connaissances approfondies et son engagement ont considérablement enrichi mon expérience de stage.

Je tiens également à remercier chaleureusement **M. Benjamin HEUCLIN** pour ses remarques pertinentes qui m'ont permis d'améliorer la qualité de mon travail. Ses suggestions précieuses ont été une source d'inspiration et m'ont encouragé à repousser mes limites.

Je souhaite exprimer ma reconnaissance à **M. Mathieu ROCHE** pour son soutien indéfectible et ses conseils avisés tout au long de ce stage. Sa guidance éclairée dans le domaine du Text Mining a été essentielle pour la réalisation de mes travaux.

Je suis également reconnaissant envers **M. Ismail El BATTEOUI**, qui a généreusement consacré son temps et ses efforts pour m'aider à comprendre les subtilités de mon stage et me prodiguer des conseils avisés afin d'obtenir des résultats probants.

J'aimerais exprimer ma gratitude envers le **CIRAD** (*Qui sommes-nous. [En ligne]. Disponible :* CIRAD) et #**DigitAg** (*À propos. [En ligne]. Disponible :* #DigitAg) pour avoir financé mon stage. Leur soutien financier a joué un rôle essentiel dans la concrétisation de mes idées et de mes objectifs.

Enfin, mes sincères remerciements vont à tous les chercheurs, doctorants et post-doctorants que j'ai eu le plaisir de rencontrer lors des présentations, réunions et événements organisés par ces institutions. Leurs échanges enrichissants et les connaissances partagées ont contribué à élargir ma compréhension et ma vision dans le domaine de la recherche.

En conclusion, je tiens à exprimer ma profonde reconnaissance envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce stage. Leur soutien inestimable et leur engagement indéfectible ont été essentiels pour atteindre les objectifs que je m'étais fixés.

Résumé

Mots-clés : Agroécologie, Correspondance, Traitement automatique du langage naturel, Ontologies, TF-IDF, BERT, Similarité, Distance de Levenshtein, Cosinus, Canonisation, Corpus, Données agricoles, Développement de systèmes agricoles durables.

Ce rapport de stage présente une étude réalisée au sein de l'**UMR TETIS**, située à la Maison De la Télédétection sur le campus Agropolis de Montpellier, en collaboration avec l'**UR AIDA**. Le stage s'est focalisé sur l'importance de la correspondance des variables **sources** et **candidates** en agroécologie.

L'objectif principal de ce stage était de résoudre la problématique liée à l'hétérogénéité des variables utilisées par les chercheurs en agroécologie. Cependant, chaque chercheur a sa propre méthode de **nomination** et de **description** des variables sources, ce qui rend la correspondance complexe et sujette à des erreurs.

Pour aborder cette problématique, différentes méthodes de représentation des données textuelles ont été explorées, telles que **TF-IDF** [1] et des approches basées sur des modèles de langues tels que **BERT-base** (section **3.3.2**), **BERT-large** (section **3.3.2)**, **RoBERTa** (section **3.3.4**) et **XLNet** (section **3.3.3**), pour la vectorisation des noms et des descriptions des variables. Des mesures de similarité, telles que la distance de **Levenshtein** [2] et le **cosinus** [3], ont été appliquées pour évaluer la proximité entre les variables.

Les résultats obtenus ont démontré des améliorations significatives par rapport aux approches précédentes [5]. Cependant, certaines limites ont été identifiées, notamment le nombre limité de variables en anglais, la formulation non canonique des variables, les descriptions courtes et l'absence de prise en compte des ontologies associées. Des recommandations ont été formulées pour surmonter ces limites, telles que la traduction des variables dans la même langue que les ontologies, la canonisation des variables non canoniques, l'extension du corpus avec des Données multilingues et hétérogènes, et l'utilisation de Méthodes de plongement de mots et de mesure de similarité.

Ce rapport met en évidence l'importance de la correspondance des variables en agroécologie. Les résultats obtenus offrent de nouvelles perspectives pour une meilleure utilisation et compréhension des données agricoles.

Abstract

Keywords : Agroecology, matching, Natural Language Processing, Ontologies, TF-IDF, BERT, Similarity, Levenshtein Distance, Cosine, Canonization, Corpus, Agricultural Data, Sustainable Agricultural Systems Development.

This internship report presents a study conducted at the **MRU TETIS** located at the Maison De la Télédétection on the Agropolis campus in Montpellier, in collaboration with the **AIDA research unit**. The focus of the internship was on the importance of **matching** between **source** and **candidate** variables in agroecology.

The main objective of this internship was to address the issue related to the heterogeneity of variables used by researchers in agroecology. However, each researcher has their own method of naming and describing source variables, which makes the matching complex and prone to errors.

To address this issue, various methods of textual data representation were explored, such as **TF-IDF** [1] and approaches based on language models like **BERT-base** (section **3.3.2**), **BERT-large** (section **3.3.2**), **RoBERTa** (section **3.3.4**), and **XLNet** (section **3.3.3**), for the vectorization of variable names and descriptions. Similarity measures, such as **Levenshtein** [2] distance and **cosine** [3], were applied to evaluate the proximity between variables.

The results demonstrated significant improvements compared to previous approaches [5]. However, certain limites were identified, including the limited number of variables in English, non-canonical formulation of variables, short descriptions, and the lack of consideration for associated ontologies. Recommendations were made to overcome these limites, such as translating variables into the same language as the ontologies, canonizing non-canonical variables, expanding the corpus with multilingual and heterogeneous data, and utilizing word embedding and similarity measurement methods.

This report highlights the importance of variable matching in agroecology. The obtained results offer new perspectives for better utilization and understanding of agricultural data.

Chapitre 1

Introduction

1.1 Contexte du stage

Pendant une période de 4 mois, j'ai effectué mon stage au sein de l'**UMR TETIS** [1], située à la Maison De la Télédétection [2] sur le campus Agropolis de Montpellier. Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence ANR-16-CONV-0004. Durant cette période, j'ai eu le privilège d'être encadré par une équipe multidisciplinaire comprenant **Mme. Sandrine AUZOUX**, spécialisée en Informatique scientifique et en Base de données, **M. Mathieu ROCHE**, expert en informatique spécialisé dans la fouille de texte, ainsi que **M. Benjamin HEUCLIN**, chercheur en Biostatistiques.

Au cours de ce stage, mon étude a porté sur l'importance de la correspondance des variables dans le domaine de l'agroécologie, qu'elles soient **candidates** [3] ou **sources** [4], afin d'assurer une interprétation cohérente des résultats.

1.2 Problématique et objectifs

Au cours de ce stage, j'ai utilisé des **Variables candidates** stockées dans le système d'information **AEGIS** [5].

L'objectif principal de ce stage est de résoudre la problématique de l'hétérogénéité des variables utilisées par les chercheurs, appelée **Variables sources**. En effet, chaque chercheur a sa propre méthode de **nomination** et de **description** de ces variables, ce qui

1. L'UMR TETIS est un laboratoire de recherche interdisciplinaire centré sur le développement de l'usage de l'information spatiale pour la compréhension de la complexité territoriale, des agro-éco systèmes et l'accompagnement des acteurs.

2. La Maison de la Télédétection en Languedoc-Roussillon rassemble principalement des équipes appartenant aux organismes de recherche et de formation, AgroParisTech, le Cirad, l'IRD et l'Irstea, installés à Agropolis. Ces équipes sont dédiées à la télédétection et plus largement à l'information spatialisée. Elles sont organisées en deux unités mixtes de recherche, l'UMR TETIS et l'UMR Espace-Dev.

3. Variables candidates ou communes ou du dictionnaire.

4. Variables chercheurs ou sources.

5. AEGIS, développé par le **CIRAD** (Auzoux et al., 2019), est une plateforme en ligne qui permet de stocker et exploiter des données provenant d'expérimentations en agroécologie menées dans les pays du Sud.

rend la tâche de correspondance des variables complexe.

Dans le cadre du stage, des efforts ont été faits pour établir une correspondance entre les variables **sources** et **candidates**. Dans un premier temps, cette correspondance sera basée uniquement sur les **noms** (par exemple, **Yield_CAS_t.ha-1**) et les **descriptions** (par exemple, **Cane yield (in fresh machinable stem)**), sans prendre en compte le **contexte** (section **4.1.1**). Par la suite, il est envisagé d'enrichir cette correspondance en incluant également d'autres **articles** pertinents (section **4.1.2**). Cette approche vise à faciliter l'utilisation et la compréhension des variables, en les reliant de manière précise et cohérente.

La **Figure 1.1** illustre la correspondance entre les variables sources et candidates en utilisant des articles comme informations supplémentaires. Pour la correspondance sans contexte, le principe reste le même, mais les articles sont exclus.

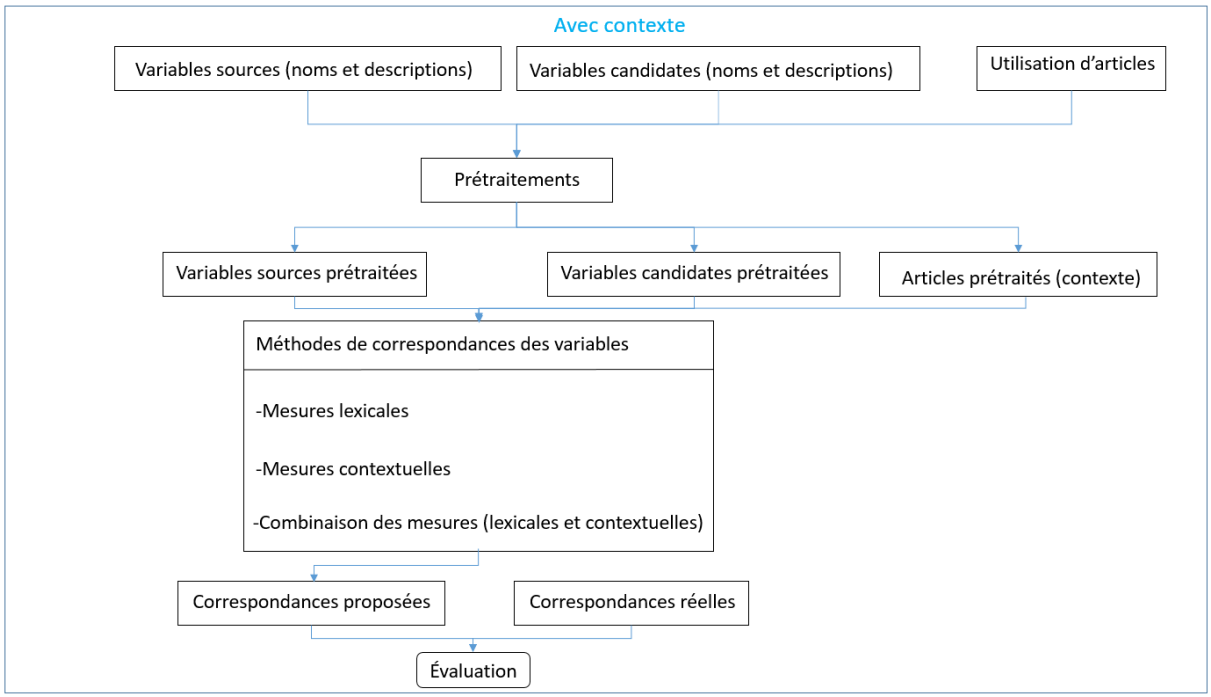

FIGURE 1.1 – Correspondance des variables avec enrichissement contextuel

Une autre tâche importante a été le développement d'une **interface web** destinée aux chercheurs. Cette interface a été conçue dans le but d'appliquer, évaluer et visualiser les résultats du travail réalisé lors du stage précédent [5] et actuel (section **3.4**).

1.3 Méthodologies employées

Dans le cadre de la problématique abordée dans la **section 1.2**, différentes méthodes de fouille de texte sont utilisées pour proposer les variables candidates qui sont les mieux adaptées aux variables sources. Les méthodes suivantes sont mobilisées :

— Des mesures lexicales (section **3.2.1**).
— Des mesures contextuelles (section **3.2.2**).

— La combinaison des mesures lexicales et contextuelles (section **3.2.3**).

Chapitre 2

État de l'art

La similarité entre les variables joue un rôle crucial dans de nombreux domaines de recherche, permettant d'évaluer la proximité ou la ressemblance entre des objets, des concepts ou des données. Elle est largement utilisée dans des domaines tels que le traitement du langage naturel, la classification de données, la recherche d'informations, et bien d'autres. Dans le contexte spécifique de l'agroécologie, la similarité entre les variables revêt une importance particulière pour comprendre les interrelations complexes qui existent entre les différentes composantes des systèmes agroécologiques.

La mise en correspondance des variables en agroécologie est une problématique clé qui consiste à identifier et à établir des relations significatives entre eux. Cela permet de mieux comprendre les liens et les interdépendances entre ces variables, et de mettre en évidence les mécanismes sous-jacents qui régissent les performances des systèmes agroécologiques. Une mise en correspondance précise des variables permet de prendre des décisions éclairées en matière de gestion agricole durable et de maximiser l'efficacité des pratiques agroécologiques.

Dans cette revue de littérature, nous abordons la problématique de la similarité, qui a été étudiée dans plusieurs travaux de recherche. Ces travaux se divisent généralement en deux approches distinctes :

1. Approche prenant en compte le contexte.
2. Approche ne prenant pas en compte le contexte.

2.1 Approche prenant en compte le contexte

Cette approche utilise des techniques qui prennent en considération le contexte. Dans ce chapitre, nous résumons l'article [10], qui présente une étude explorant l'utilisation de BERT pour améliorer la correspondance entre les produits. L'article décrit le modèle **eComBERT**, développé sur la base de BERT, dans le but d'apprendre une représentation de similarité facilitant la mise en correspondance des produits.

L'approche décrite dans [10] exploite les caractéristiques linguistiques et sémantiques des descriptions de produits pour calculer leur similarité. Le modèle BERT a été entraîné sur un ensemble de données comprenant des paires de produits, accompagnées de leurs scores de similarité respectifs.

Les résultats présentés dans [10] démontrent que le modèle **eComBERT** surpasse les approches traditionnelles en termes de précision et de performance pour la mise en correspondance des produits. Il est capable de capturer les similitudes subtiles entre les produits, permettant ainsi de générer des correspondances plus précises.

Bien que l'article cité précédemment ne traite pas directement de la mise en correspondance des variables en agroécologie, il fournit des informations sur les avancées récentes dans le domaine des modèles de langues et leur utilisation dans le traitement du langage naturel. Le modèle BERT est reconnu pour sa capacité à capturer les relations sémantiques et le contexte dans le langage [17], ce qui en fait une approche prometteuse pour évaluer la similarité entre les variables en agroécologie.

2.2 Approche ne prenant pas en compte le contexte

Dans cette approche, les chercheurs se concentrent principalement sur des mesures de similarité générales qui ne prennent pas en compte le contexte spécifique de l'agroécologie. Ils utilisent des méthodes telles que **TF-IDF** [1] et la distance de **Levenshtein** [2] pour évaluer la similarité entre les variables. Cependant, il convient de noter que ces approches peuvent présenter certaines limites pour capturer la similarité sémantique et de prendre en compte le contexte spécifique de l'agroécologie.

Cette approche a été abordée dans un travail précédent menée en 2022 [5], où la distance de **Levenshtein** (partie **A** de la section **4.1.1**) a été utilisée comme mesure de similarité entre les **noms** des deux types de variables. De plus, le **TF-IDF** (section **3.2.2**) a été employé pour la vectorisation des **descriptions** et le calcul du **cosinus** a été utilisé comme mesure de similarité entre les vecteurs de ces descriptions. Ces deux techniques ont ensuite été combinées dans une méthode appelée **combinaison** (section **3.2.3**). Cependant, il est important de noter que l'utilisation du **TF-IDF** dans cette approche ne tient pas compte du contexte spécifique de l'agroécologie, ce qui peut limiter les résultats obtenus.

Pour illustrer les limites du **TF-IDF** et de la distance de **Levenshtein**, examinons l'exemple de deux variables dans le domaine de l'agroécologie : ***Rendement des cultures*** et ***Utilisation d'engrais***. Selon le **TF-IDF** et la distance de **Levenshtein**, ces deux

variables pourraient sembler assez différentes en termes de **noms** et de **descriptions**.

Cependant, il est important de noter que le **TF-IDF** est une mesure de la fréquence et de l'importance des mots dans un document, mais il ne prend pas en compte le sens global du texte. Par conséquent, il peut ne pas capturer la similarité sémantique entre les variables.

De même, la distance de **Levenshtein** mesure la différence entre deux chaînes de caractères en termes de modifications nécessaires pour les transformer l'une en l'autre. Bien qu'elle puisse être utile pour comparer des **noms** de variables similaires, elle ne prend pas en compte le contexte ou la signification des mots.

Dans l'ensemble, l'étude [5] a permis de progresser dans la recherche sur **la correspondance des variables** dans le domaine de l'agroécologie. Cependant, il reste des lacunes à combler, notamment en prenant en compte le contexte spécifique. Cela peut être réalisé en utilisant des modèles de langues plus avancés tels que BERT, qui sont capables de capturer la sémantique et le contexte des mots. Ces modèles sont détaillés en section **3.3.2** (chapitre 3).

En conclusion, la mise en correspondance des variables en agroécologie est une problématique active qui suscite de nombreuses recherches. Dans le cadre de ce rapport de stage, notre objectif est d'améliorer la correspondance entre les variables agroécologiques, en particulier entre les variables sources et candidates. Nous proposons une approche novatrice en combinant des méthodes prenant en compte le contexte avec d'autres qui ne le prennent pas en compte.

Notre approche consiste à tirer parti des avantages des deux approches en intégrant le contexte spécifique de l'agroécologie tout en utilisant des mesures de similarité générales. En combinant de manière judicieuse ces approches, nous avons réussi à obtenir des évaluations de similarité plus précises et pertinentes pour les variables agroécologiques.

À notre connaissance, il n'existe actuellement aucune recherche dans le domaine de l'agroécologie qui ait mis en œuvre cette approche combinée. Par conséquent, notre travail de stage vise à combler cette lacune en explorant la faisabilité et l'efficacité de cette méthode dans le contexte spécifique de l'agroécologie.

Chapitre 3

Méthodes et outils

3.1 Description et préparation du jeu de données

3.1.1 Description du jeu de données

Dans le cadre de ce rapport de stage M2, le jeu de données utilisé se compose de quatre fichiers texte distincts. Le premier fichier contient les **noms** des variables **sources** [12], le deuxième fichier contient les **descriptions** de ces variables [13], le troisième fichier contient les **noms** des variables **candidates** [14], et enfin, le quatrième fichier contient les **descriptions** de ces variables [15].

Noms des variables sources	**Descriptions des variables sources**
Yield_CAS_t.ha-1	Cane yield (in fresh machinable stem)
Sugar_CAS_%	Sugar content of fresh stem mass
Rec_globale_plein_%	full weed and service plant coverage
...	...

TABLE 3.1 – Exemples de noms et de descriptions des variables sources

Noms des variables candidates	**Descriptions des variables candidates**
root_crop_yield_dm_t.ha-1	measurement of root dry biomass at plot level
root_crop_yield_fm_t.ha-1	measurement of root fresh biomass at plot level
stem_crop_yield_dm_t.ha-1	measurement of dry stem biomass at plot level
...	...

TABLE 3.2 – Exemples de noms et de descriptions des variables candidates

De plus, dans le cadre de la **validation**, un fichier de correspondances [16] a été utilisé. Ce fichier contient les véritables correspondances entre les variables **sources** et **candidates**. Il a été utilisé pour évaluer la performance des approches proposées en termes de correspondance des variables.

Variable source	Variable correspondante
Yield CAS t.ha-1	stem crop yield fm t.ha-1
Sugar CAS %	stem sugar fm content %
Rec globale plein %	plant ground cover %
...	...

TABLE 3.3 – Exemples de correspondances réelles des variables

*Note 1 : Les **noms** des variables (sources et candidates) sont une combinaison de leurs **noms** et de leurs **unités de mesure**, l'unité se trouve après le dernier _ du nom de la variable, prenons l'exemple suivant : le nom de la variable **root_ crop_ yield_ dm_ t.ha-1** se compose de son nom, **root_ crop_ yield_ dm**, et de son unité de mesure qui est **t.ha-1**.*

*Note 2 : Dans le cadre de la **validation**, les underscores (_) sont supprimés des **noms** des variables **sources** et **candidates** afin de pouvoir les comparer aux noms des variables du fichier de correspondances qui ne les contiennent pas.*

*Note 3 : La similarité a été calculée entre chaque variable **source** et toutes les autres variables **candidates**, triées du plus similaire au moins similaire, en utilisant les mesures lexicales, contextuelles et de combinaison. Ensuite, nous avons utilisé la précision à la position n (P@n) pour évaluer les résultats. P@n représente la probabilité qu'une variable candidate correcte soit incluse parmi les n premières variables candidates sélectionnées pour chaque variable source spécifique.*

3.1.2 Préparation du jeu de données

Dans le contexte de l'agroécologie, l'une des principales problématiques est la correspondance entre les variables **sources** et **candidates**. Pour résoudre cette problématique, la préparation des données textuelles joue un rôle essentiel. Elle vise à transformer les **noms** et les **descriptions** des variables en une forme appropriée pour faciliter leur comparaison et leur correspondance.

La préparation des données textuelles (noms et descriptions des variables) qui a été appliquée comprend plusieurs étapes :

1. **clean_text()** : Cette fonction est utilisée pour nettoyer les variables sources et candidates. Elle élimine les nombres, les parenthèses [1] et leur contenu, et convertit les noms et les descriptions en minuscules. En normalisant les variables, cette étape facilite la comparaison et l'alignement ultérieur.

   Référence : Voir l'annexe **4.1.3** pour le code de la fonction clean_text() et les exemples de données avant et après le prétraitement.

1. La suppression des parenthèses et de leur contenu a amélioré les performances. Cependant, pour les travaux futurs, nous explorerons comment tirer parti de ces informations.

2. **remove_stopwords()** : Les stopwords sont des mots couramment utilisés dans la langue qui n'apportent pas de signification particulière dans le contexte de l'agroécologie. Cette fonction supprime ces mots fonctionnels, tels que les prépositions et les conjonctions. En éliminant les stopwords, nous nous concentrons sur les termes clés qui sont plus significatifs pour la correspondance entre les variables.

   Référence : Voir l'annexe **4.1.3** pour le code de la fonction remove_stopwords() et les exemples de données avant et après le prétraitement.

3. **lemmatize()** : La lemmatisation est une technique linguistique qui consiste à ramener les termes à leur forme canonique ou à leur lemme. Elle permet de transformer les noms du pluriel au singulier et les verbes à leur forme infinitive. Par exemple, elle permet de ramener les termes ***rédige***, ***rédiges*** et ***rédigé*** à leur forme de base ***rédiger***. La lemmatisation facilite la correspondance entre les termes similaires, améliorant ainsi la précision de l'alignement des variables.

   Référence : Voir l'annexe **4.1.3** pour le code de la fonction lemmatize() et les exemples de données avant et après le prétraitement.

4. **remove_punctuation()** : Cette fonction supprime la ponctuation des descriptions des variables. En éliminant les caractères spéciaux tels que les points, les virgules et les guillemets, nous évitons les interférences indésirables lors de la correspondance entre les variables.

5. **replace_synonyms()** : Pour faciliter la correspondance entre les variables, cette technique permet de remplacer certains mots par leurs synonymes. Par exemple, le mot ***degré*** peut être remplacé par ***niveau***. En utilisant des termes équivalents, cette étape a amélioré la cohérence et la précision de l'alignement des variables.

   Référence : Voir l'annexe **4.1.3** pour le code de la fonction replace_synonyms() et les exemples de données avant et après le prétraitement.

Note : toutes les fonctions ont été appliquées aux ***descriptions*** *des variables* ***sources*** *et* ***candidates****, tandis que seules les trois premières fonctions ont été utilisées pour les* ***noms*** *des variables.*

3.2 Mise en place des méthodes de mise en correspondance de variables

3.2.1 Mesure lexicale

L'objectif de cette approche est de comparer les **noms** de variables **sources** et **candidates** en se basant sur leur chaîne de caractères. Dans le cadre de cette approche, la

distance de **Levenshtein** [2] a été utilisée. Cette distance permet de calculer le nombre de changements nécessaires entre deux chaînes de caractères.

La distance de **Levenshtein** peut être formulée mathématiquement comme suit :

$$\text{lev}(a,b) = \begin{cases} \max(\text{len}(a), \text{len}(b)) & \text{si } \min(\text{len}(a), \text{len}(b)) = 0 \\ \min \begin{cases} \text{lev}(\text{tail}(a), b) + 1 \\ \text{lev}(a, \text{tail}(b)) + 1 \\ \text{lev}(\text{tail}(a), \text{tail}(b)) + \text{diff}(a[0], b[0]) \end{cases} & \text{sinon} \end{cases} \tag{3.1}$$

où $\text{len}(a)$ représente la longueur de la chaîne a, $\text{tail}(a)$ représente la chaîne a sans son premier caractère, et $\text{diff}(x, y)$ est égal à 0 si $x = y$, et à 1 sinon.

Cependant, la distance de **Levenshtein** présente une faiblesse en agroécologie, comme l'illustre l'exemple suivant : supposons que nous ayons une variable **source** appelée ***Crop_ Yield*** (rendement des cultures) et une variable **candidate** appelée ***Crop_ Weed*** (présence de mauvaises herbes dans les cultures). Si nous utilisons la distance de **Levenshtein** pour mesurer la similarité entre ces deux chaînes de caractères, nous obtiendrons une distance relativement faible. Cependant, il est évident que ces deux variables sont distinctes et n'ont pas de correspondance directe. La différence entre ***Yield*** (rendement) et ***Weed*** (mauvaises herbes) est significative et ne peut pas être compensée par la similitude de caractères entre les mots ***Crop***. Cela montre la faiblesse de la distance de **Levenshtein** dans ce contexte, car elle ne prend pas en compte le sens ou le contexte des mots, mais se concentre uniquement sur la similarité de caractères.

Note : $\textbf{\textit{lev}}(a, b) \in [0, \infty[$. *Plus la valeur de Levenshtein est petite, plus il y a une forte similarité.*

Pour mieux appréhender le fonctionnement de la distance de **Levenshtein**, examinons les exemples suivants :

- **Exemple de deux mots similaires :** Considérons les mots ***agro-écologie*** et ***agroécologie***. La distance de **Levenshtein** entre ces deux mots est de 1, car il suffit de supprimer le tiret pour les transformer l'un en l'autre.
- **Exemple de deux mots non similaires :** Prenons maintenant les mots ***agro-écologie*** et ***biodynamie***. Dans ce cas, la distance de **Levenshtein** entre ces deux mots est de 10. Pour transformer *agro-écologie* en *biodynamie*, plusieurs opérations sont nécessaires : 2 suppressions pour éliminer les tirets, 2 substitutions pour remplacer le *'a'* par *'i'* et le *'é'* par *'y'*, et 6 insertions pour ajouter les lettres *'b'*, *'i'*, *'d'*, *'n'*, *'a'* et *'m'*.

3.2.2 Mesure contextuelle

Le but de l'approche contextuelle est de comparer les variables à travers leurs **descriptions**. Dans cette approche, nous utilisons deux types de méthodes : **sans contexte**[2], comme **TF-IDF** [1], et **avec contexte**[3], en utilisant des modèles de langues tels que **BERT-base** [17], **BERT-large [17]**, **XLNet** [19] et **RoBERTa** [20].

La mesure **TF-IDF** est une méthode classique pour évaluer l'importance des termes dans un document par rapport à une collection de documents.

La mesure **TF** représente la fréquence du terme dans le document. Elle est calculée en divisant le nombre d'occurrences du terme par le nombre total de termes dans le document. La formule mathématique de **TF** pour un terme t dans un document d est donnée par :

$$\mathrm{TF}(t, d) = \frac{\text{nombre d'occurrences de } t \text{ dans } d}{\text{nombre total de termes dans } d}$$

L'**IDF** mesure l'importance globale d'un terme dans la collection de documents. Elle est calculée en prenant le logarithme inverse de la proportion du nombre total de documents sur le nombre de documents contenant le terme. La formule mathématique de **IDF** pour un terme t dans une collection de documents est donnée par :

$$\mathrm{IDF}(t) = \log\left(\frac{\text{nombre total de documents}}{\text{nombre de documents contenant } t}\right)$$

La mesure **TF-IDF** est obtenue en multipliant la valeur de **TF** par la valeur de **IDF** pour chaque terme. Ainsi, les termes qui sont fréquents dans un document particulier tout en étant rares dans l'ensemble de la collection auront une valeur **TF-IDF** plus élevée.

Cependant, en agroécologie, la mesure **TF-IDF** peut présenter certaines faiblesses. Par exemple, si un terme spécifique à l'agriculture est fréquent dans tous les documents de la collection, sa valeur **TF-IDF** sera faible, ce qui peut diminuer sa pertinence dans la comparaison des variables. Par conséquent, il est important de prendre en compte ces limites lors de l'application de la mesure **TF-IDF** en agroécologie.

Pour comprendre le fonctionnement de **TF-IDF**, supposons que nous ayons une collection de documents sur l'agroécologie, et nous voulons calculer les scores **TF-IDF** pour certains termes clés. Voici un exemple de trois documents de la collection :

- **Document 1 :** L'agroécologie favorise la biodiversité dans les pratiques agricoles.
- **Document 2 :** Les techniques agroécologiques réduisent l'utilisation de pesticides.
- **Document 3 :** L'agriculture conventionnelle utilise intensivement des produits chimiques.

2. Les méthodes qui ne prennent pas en considération le contexte se basent principalement sur des mesures de similarité ou de fréquence de termes.

3. Les méthodes qui prennent en compte le contexte utilisent souvent des modèles linguistiques avancés, tels que **BERT**.

Terme	Document 1	Document 2	Document 3
agroécologie	0.287	0.287	0
biodiversité	0.287	0	0
pesticides	0	0.287	0.693

TABLE 3.4 – Scores TF-IDF pour les termes clés

*Note : **TF-IDF** $\in [0, 1]$. Plus la valeur de **TF-IDF** est proche de 1, plus le terme est important.*

La distance **cosinus** [3] est une mesure de similarité largement utilisée pour comparer la similarité entre deux vecteurs. Elle est souvent utilisée dans des tâches de classification de texte, de recommandation et de correspondance. La distance **cosinus** mesure l'angle entre deux vecteurs dans un espace multidimensionnel.

La formule mathématique pour calculer la distance **cosinus** entre deux vecteurs **A** et **B** est donnée par :

$$\text{cosine\_similarity}(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}$$

où $\cdot$ représente le produit scalaire des vecteurs, θ l'angle entre les vecteurs **A** et **B**, et $|\mathbf{A}|$ et $|\mathbf{B}|$ sont les normes des vecteurs **A** et **B** respectivement.

Voici un exemple illustratif de calcul de la distance **cosinus** entre deux vecteurs **A** et **B** :

$$\mathbf{A} = [2, 3, 1, 5] \quad \text{et} \quad \mathbf{B} = [4, 1, 3, 2]$$

$$\begin{aligned}\text{cosine\_similarity}(\theta) &= \frac{2 \cdot 4 + 3 \cdot 1 + 1 \cdot 3 + 5 \cdot 2}{\sqrt{2^2 + 3^2 + 1^2 + 5^2} \cdot \sqrt{4^2 + 1^2 + 3^2 + 2^2}} \\ &= \frac{8 + 3 + 3 + 10}{\sqrt{4 + 9 + 1 + 25} \cdot \sqrt{16 + 1 + 9 + 4}} \\ &= \frac{24}{\sqrt{39} \cdot \sqrt{30}} \approx 0.7\end{aligned}$$

*Note : **cosine_similarity**$(\theta) \in [0, 1]$. Plus la valeur de **cosine_similarity**(θ) est proche de 1, plus les vecteurs sont similaires.*

3.2.3 Combinaison

La méthode de **combinaison** permet de combiner les mesures lexicales et contextuelles afin d'améliorer la précision de la correspondance des variables.

Cette méthode est définie par la formule mathématique suivante :

$$\textbf{combinaison} = \alpha \cdot \mathrm{X} + (1 - \alpha) \cdot Y [5]$$

où $\alpha \in]0, 1[$ est un facteur de pondération attribué à X dans cette équation. Il est utilisé pour contrôler l'influence respective de X et Y dans la combinaison finale.

L'un des avantages de l'utilisation du paramètre α est sa capacité à régler le degré d'importance accordé à X et Y dans la combinaison. En ajustant la valeur de α, on peut privilégier davantage X (lorsque α est proche de 1) ou Y (lorsque α est proche de 0).

D'autre part, un inconvénient du paramètre α est sa sensibilité aux variations. De petites modifications dans la valeur de α peuvent entraîner des changements significatifs dans le résultat final. Cela nécessite donc une expérimentation et une évaluation rigoureuses afin de trouver la valeur optimale pour α.

Dans ce contexte, les valeurs de α ont été choisies entre 0.1 et 0.9, avec un pas de 0.01. Les meilleurs résultats obtenus **sans contexte** ont été obtenus pour α=0.79, tandis que les meilleurs résultats **avec contexte** ont été obtenus pour α=0.25.

*Note : **combinaison*** $\in [0, 1]$.

3.3 Extension des méthodes de mise en correspondance de variables

Dans cette section, nous explorons l'extension des méthodes de mise en correspondance de variables en utilisant des modèles de langues basés sur **BERT**.

3.3.1 Modèles de langues

Les modèles de langues jouent un rôle essentiel dans le domaine du traitement automatique du langage naturel [7]. Ils visent à capturer les structures et les relations linguistiques dans un corpus de texte pour générer des prédictions précises. Dans cette sous-section, nous aborderons les différents types de modèles de langues, en commençant par les **n-grammes** [7], puis en discutant des modèles basés sur les **RNN** [8] et les **LSTM** [9] , pour finalement présenter les modèles de langues basés sur les transformers, tels que **BERT** [17].

A. n-grammes

Les modèles de langues basés sur les n-grammes sont parmi les plus simples. Ils reposent sur l'idée d'exploiter les fréquences d'apparition des n-grammes (séquences de mots) dans un corpus pour prédire le mot suivant. Par exemple, un modèle de langue basé sur les trigrammes considère les deux mots précédents pour prédire le prochain mot.

Bien que les n-grammes soient faciles à mettre en œuvre et puissent fournir des résultats acceptables pour des tâches de langage simples, ils présentent des limites majeures. L'une des principales faiblesses des n-grammes est leur incapacité à capturer les dépendances à long terme entre les mots, ce qui limite leur capacité à générer des séquences de mots cohérentes.

B. RNN et LSTM

Pour remédier aux limites des modèles de langues basés sur les n-grammes, les RNN ont été introduits. Les RNN sont conçus pour traiter des séquences de données, ce qui les rend bien adaptés pour modéliser les séquences de mots dans un texte. Ils peuvent prendre en compte les dépendances à long terme en propageant l'information d'un pas de temps à l'autre. Cependant, les RNN traditionnels souffrent du problème du *vanishing gradient* et ont du mal à capturer des dépendances à long terme.

Les LSTM sont une extension des RNN qui permettent de mieux gérer les dépendances à long terme. Grâce à l'utilisation de portes de mémoire, les LSTM peuvent décider de conserver ou d'oublier certaines informations, ce qui améliore la capacité du modèle à capturer des dépendances à plus long terme. Cependant, même les LSTM ont leurs limites. Ils peuvent avoir du mal à capturer des relations complexes entre les mots et peuvent souffrir de problèmes de surapprentissage lorsqu'ils sont confrontés à de grands ensembles de données.

C. Modèles de langues basés sur les transformers (BERT)

Les modèles de langues basés sur les transformers, tels que BERT, ont révolutionné le domaine du TALN ces dernières années. Les transformers exploitent une architecture d'attention pour capturer les relations entre tous les mots d'une séquence, à la fois dans le contexte avant et après. Cette approche bidirectionnelle permet au modèle de comprendre plus efficacement les relations sémantiques complexes dans le texte.

BERT a été largement reconnu pour ses performances exceptionnelles dans de nombreuses tâches de TALN [17], y compris la compréhension de texte, la traduction automatique et le résumé automatique. Son architecture transformer lui permet de capturer de manière plus précise les dépendances à long terme par rapport aux modèles précédents. De plus, BERT peut être pré-entraîné sur de vastes corpus non supervisés, ce qui lui permet d'acquérir une connaissance linguistique riche et de s'adapter à diverses tâches spécifiques.

D. Conclusion

En conclusion, les modèles de langues ont évolué au fil des années, passant des n-grammes aux RNN/LSTM, pour finalement aboutir aux transformers, tels que BERT. Chaque approche présente ses avantages et ses faiblesses spécifiques. les modèles basés sur les transformers, notamment BERT, se sont révélés être parmi les plus performants, en capturant de manière plus précise les dépendances à long terme dans le texte [4].

3.3.2 BERT

BERT [17] est un **Modèle de langues pré-entraîné** sur de vastes corpus, tels que Wikipedia, qui utilise le mécanisme d'attention [4] pour comprendre le contexte des mots. Il prend en compte à la fois le contexte à gauche et à droite de chaque mot, ce qui lui permet de capturer les informations contextuelles de manière précise. Cette capacité à saisir le contexte permet à BERT de générer des représentations vectorielles de mots riches en sémantique. Il convient de noter qu'il existe deux types de modèles BERT couramment utilisés : **BERT-base** [17] et **BERT-large** [17].

A. Pré-entraînement de BERT

BERT est pré-entraîné sur deux tâches non supervisées : **MLM** et **NSP**.

Pour la tâche **MLM**, 15% des jetons WordPiece dans chaque séquence d'entré sont masqués au hasard. Dans 80% des cas, le jeton masqué est remplacé par le jeton [MASK], dans 10% des cas par un jeton aléatoire et dans 10% des cas, le jeton reste inchangé. BERT prédit uniquement les mots masqués plutôt que de reconstruire l'ensemble de l'entrée.

Pour la tâche **NSP**, BERT prédit si une phrase suit logiquement une autre dans une paire de phrases. Plus précisément, lors du choix des phrases A et B pour chaque exemple de pré-entraînement, 50% du temps, B est la phrase qui suit logiquement A (étiquetée **IsNext**), et 50% du temps, il s'agit d'une phrase aléatoire sélectionnée du corpus (étiquetée **NotNext**) dans la tâche NSP. Cette approche permet à BERT de capturer les relations de séquence et d'apprendre à comprendre le contexte entre les paires de phrases.

En combinant ces deux tâches non supervisées, BERT parvient à créer des représentations pré-entraînées profondes bidirectionnelles qui capturent efficacement les relations contextuelles entre les mots et les phrases.

B. Représentation d'entrée de BERT

La représentation d'entrée de BERT est obtenue en sommant les embeddings de jetons, les embeddings de segmentation et les embeddings de position. Chacun de ces types d'embeddings joue un rôle spécifique dans la capture des informations textuelles (Figure **3.1**).

— Les embeddings de jetons représentent individuellement chaque jeton d'entrée et encodent les informations sémantiques des mots. Ils permettent à BERT de comprendre le sens des mots dans le contexte global du texte.

— Les embeddings de segmentation sont utilisés pour différencier différentes parties de l'entrée, comme les phrases ou les segments de phrases. Ils permettent à BERT de saisir les relations et les dépendances entre ces parties du texte.

— Les embeddings de position indiquent la position relative de chaque jeton dans la séquence. Ils aident BERT à comprendre l'ordre des mots et les relations temporelles présentes dans le texte.

En combinant ces embeddings, BERT crée une représentation d'entrée qui intègre à

4. Le mécanisme d'attention permet au modèle de se concentrer sur les parties importantes du texte en donnant plus de poids aux mots pertinents.

la fois le sens des mots, les relations entre les parties du texte et l'ordre des mots. Cette représentation permet à BERT de prendre en compte les informations contextuelles lors du traitement du langage naturel.

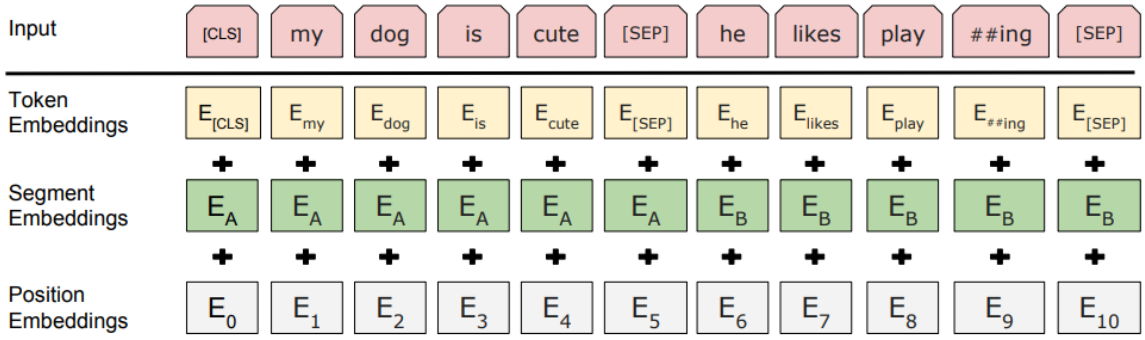

FIGURE 3.1 – Représentation d'entrée de BERT. Les embeddings d'entrée sont la somme des embeddings de jetons, des embeddings de segmentation et des embeddings de position [17].

C. Différences entre BERT-base et BERT-large

Les principales différences entre **BERT-base** et **BERT-large** sont résumées dans la Table suivante :

Caractéristiques	**BERT-base**	**BERT-large**
Nombre de couches de transformer	12	24
Taille du modèle (nombre de paramètres)	110 millions	340 millions
Têtes d'attention	12	16
Dimensions (taille de sortie)	768	1024
Capacité de traitement	Moins élevée	Plus élevée
Complexité de l'information contextuelle	Moins complexe	Plus complexe
Temps d'entraînement	Moins long	Plus long
Temps d'inférence	Moins long	Plus long
Utilisation de mémoire	Moins importante	Plus importante

TABLE 3.5 – Différences entre BERT-base et BERT-large [18]

Il convient de noter que **BERT-large** a une capacité de traitement et une complexité de l'information contextuelle supérieures grâce à sa plus grande taille et son nombre de couches de transformer plus élevé. Cependant, cela se traduit par un temps d'entraînement et d'inférence plus long, ainsi qu'une utilisation de mémoire plus importante par rapport à **BERT-base**.

D. Exemple illustrant la puissance de BERT

Pour illustrer la puissance de BERT en agroécologie, prenons l'exemple suivant : supposons que nous ayons deux variables : ***rendement des cultures*** et ***production agricole par hectare***. Ces termes sont lexicalement différents, mais ils partagent une signification similaire dans le contexte de l'agroécologie. Utilisant la méthode de Levenshtein, la distance entre ces deux variables serait relativement élevée en raison des différences dans les

séquences de caractères. Cependant, en utilisant BERT, qui prend en compte le contexte et la sémantique des mots, la similarité entre ces deux variables serait élevée en raison de leur similitude conceptuelle liée à la mesure de la productivité agricole.

Cet exemple met en évidence la capacité de BERT à capturer des informations sémantiques et contextuelles, ce qui peut conduire à de meilleures correspondances entre les variables en agroécologie, même lorsque les termes sont lexicalement différents.

3.3.3 XLNet

XLNet [19] est un modèle de langue pré-entraîné basé sur une approche de **Permutations de mots**, contrairement à **BERT** qui utilise une approche de **Masquage de mots**. Cette approche permet à **XLNet** de capturer les dépendances entre les mots dans toutes les directions.

Les principales différences entre **XLNet** et **BERT** sont résumées dans le tableau suivant :

Caractéristiques	**XLNet**	**BERT**
Approche d'apprentissage	Permutation de mots	Masquage de mots
Exploration des permutations	Oui	Non
Capture des dépendances	Dans toutes les directions	Limitée par le masquage
Mécanisme d'attention relatif	Oui	Non
Modélisation des relations longue distance	Améliorée	Limitée

TABLE 3.6 – Différences entre XLNet et BERT

Il convient de noter que **XLNet**, grâce à son approche de permutation de mots, peut explorer toutes les permutations possibles lors de l'entraînement, ce qui lui permet de mieux capturer les relations de dépendance entre les mots et d'améliorer la qualité des représentations vectorielles. De plus, l'introduction d'un mécanisme d'attention relatif dans **XLNet** lui permet de considérer les relations entre les mots en fonction de leur position relative dans la phrase, ce qui aide à modéliser les relations longue distance et à capturer les dépendances complexes entre les mots.

Un exemple en agroécologie pour illustrer la différence entre XLNet et BERT serait le suivant : supposons que nous ayons deux phrases : ***l'utilisation des pesticides affecte la biodiversité*** et ***la biodiversité est impactée par l'utilisation des pesticides***. En utilisant **BERT**, qui se base sur le masquage de mots, les représentations vectorielles des mots ***affecte*** et ***impactée*** seraient similaires, car ces mots partagent une similarité de contexte dans les deux phrases. Cependant, en utilisant **XLNet**, qui explore toutes les permutations possibles, les représentations vectorielles des mots seraient différentes. Ainsi, **XLNet** pourrait mieux capturer la différence de sens entre les phrases et fournir des représentations vectorielles distinctes pour les mots ***affecte*** et ***impactée***.

Cet exemple met en évidence la capacité de **XLNet** à capturer les dépendances entre les mots dans toutes les directions et à considérer le contexte global, ce qui lui confère un avantage par rapport à BERT dans la modélisation des relations complexes entre les mots [19].

3.3.4 RoBERTa

RoBERTa [20] est une variante de **BERT** qui a été développée pour améliorer sa performance et sa robustesse. Tout comme **BERT, RoBERTa** est basé sur une architecture de transformers et utilise un pré-entraînement sur de larges corpus de textes pour apprendre des représentations vectorielles de mots. Cependant, il existe quelques différences notables entre RoBERTa et **BERT**, comme présenté dans le tableau ci-dessous :

Caractéristiques	**BERT**	**RoBERTa**
Pré-entraînement	Wikipedia, BooksCorpus	Wikipedia, CC-News, OpenWebText
Taille du modèle (nombre de paramètres)	110 millions	355 millions
Méthode de masquage	WordPiece	SentencePiece
Durée d'entraînement	4 jours	4 semaines
Taille des mini-batch	256	8 192
Nombre d'itérations d'entraînement	1 million	500 000

TABLE 3.7 – Différences entre BERT et RoBERTa [21]

Pour illustrer la différence entre **BERT** et **RoBERTa** en agroécologie, prenons l'exemple suivant : supposons que nous ayons un corpus de textes relatifs à la prédiction des rendements agricoles. En utilisant **BERT**, le modèle pourrait être capable de comprendre les relations entre les mots et les concepts liés à l'agriculture, mais il pourrait rencontrer des difficultés à traiter des termes spécifiques à l'agroécologie. En revanche, avec **RoBERTa**, qui a été entraîné sur un corpus plus diversifié incluant des textes provenant de différentes sources, le modèle pourrait mieux saisir les subtilités et les spécificités des termes et des concepts propres à l'agroécologie, améliorant ainsi la qualité des représentations vectorielles et les performances dans ce domaine.

Dans la section suivante, nous aborderons le développement d'une interface web dédiée à l'application des méthodes précédemment décrites et à la visualisation des résultats obtenus. Cette interface permettra aux utilisateurs d'exploiter facilement les différentes techniques et de bénéficier d'une présentation claire et intuitive des résultats obtenus.

3.4 Proposition et mise en œuvre d'une interface web

Au cours de ce stage, une **interface web** a été développée, dans le but de fournir aux chercheurs un outil pratique. Cette interface leur permet d'appliquer des **mesures de correspondance** entre les variables **sources** et les variables **candidates**, de visualiser les résultats et de sélectionner le nombre de variables **candidates** à afficher pour chaque variable **source**. Par défaut, ce nombre est fixé à **10**. De plus, les chercheurs ont la possibilité de choisir la mesure utilisée pour afficher les résultats, telles que **lexicale, contextuelle ou une combinaison des deux**. L'interface leur permet également de sélectionner les variables **candidates** qu'ils estiment correctes pour chaque variable **source**, puis de télécharger les résultats finaux dans différents formats tels que **CSV, Excel** et **PDF**. Enfin, une barre de recherche est disponible pour faciliter la recherche des variables (Annexe **3.4.3**).

3.4.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les interactions entre les **acteurs** et le **système**. Il permet de visualiser les fonctionnalités offertes par le système du point de vue des utilisateurs. Dans notre cas, le système est représenté par l'interface web développée lors du stage, tandis que les utilisateurs sont les chercheurs.

Le diagramme de cas d'utilisation ci-dessous illustre les principales fonctionnalités de l'interface web :

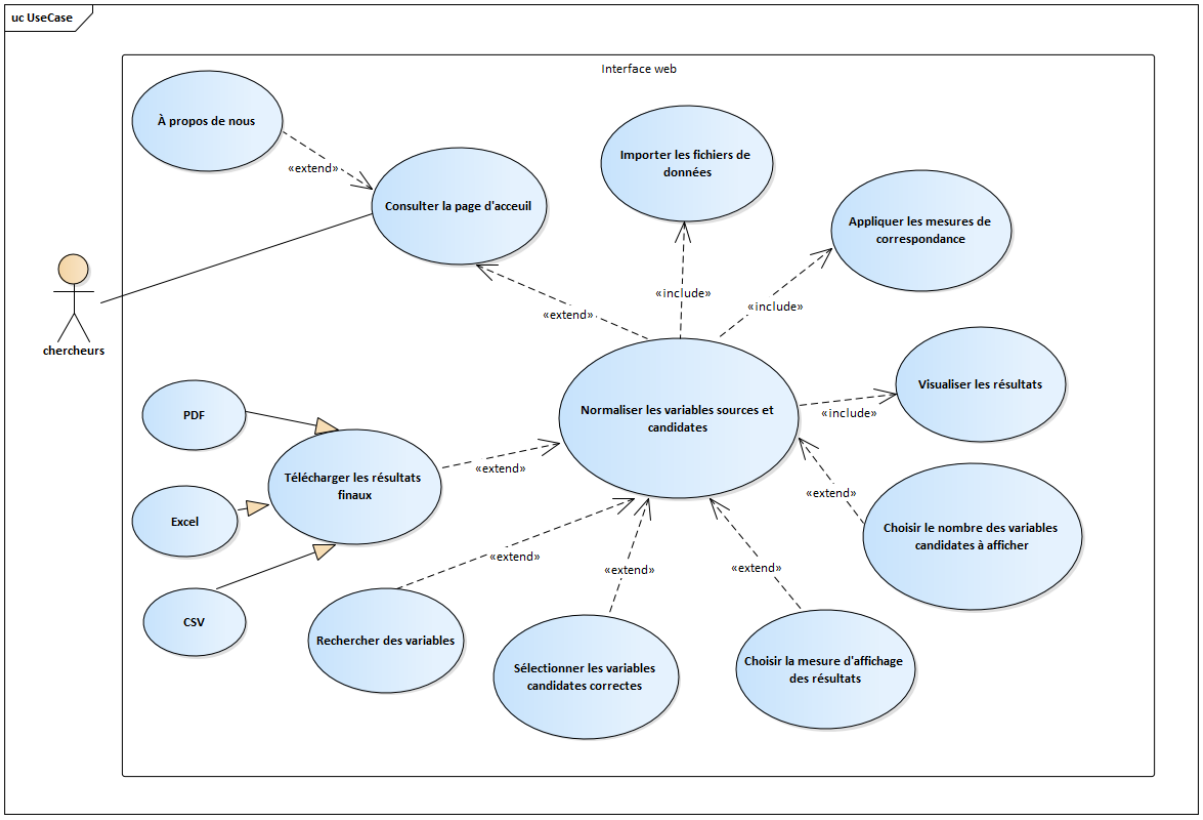

FIGURE 3.2 – Diagramme de cas d'utilisation

Les cas d'utilisation identifiés sont les suivants :

— **La page d'acceuil** : Ce cas d'utilisation permet aux utilisateurs de l'interface web de parcourir des citations sur l'agroécologie. Ils ont également la possibilité de basculer vers d'autres cas d'utilisation tels que **À propos de nous** ou **Normaliser les variables sources et candidates**.

— **À propos de nous** : Ce cas d'utilisation permet aux utilisateurs de l'interface web de consulter des informations sur l'équipe de ce stage. Vous y trouverez une description du développeur de l'interface (moi-même), ainsi que des informations sur les encadrants de stage et le stagiaire de l'année 2022.

— **Normaliser les variables sources et candidates** : Ce cas d'utilisation permet aux utilisateurs de faire correspondre les variables sources et candidates.

— **Importer les fichiers de données** : Ce cas d'utilisation permet aux utilisateurs d'importer les fichiers de données contenant les noms et les descriptions des variables sources et candidates. Après l'importation, une vérification des formats est effectuée afin de garantir l'intégrité des données.

— **Appliquer les mesures de correspondance** : Permet aux chercheurs d'appliquer les méthodes de correspondance des variables sources et candidates.

— **Visualiser les résultats** : Permet aux chercheurs de visualiser les résultats des correspondances effectuées sous forme d'un tableau paginé.

— **Choisir le nombre de variables candidates à afficher** : Permet aux chercheurs de sélectionner le nombre de variables candidates à afficher pour chaque variable source.

— **Choisir la mesure d'affichage des résultats** : Permet aux chercheurs de choisir la mesure utilisée pour afficher les résultats, telle que lexicale, contextuelle ou une combinaison des deux.

— **Sélectionner les variables candidates correctes** : Permet aux chercheurs de sélectionner les variables candidates qu'ils estiment correctes pour chaque variable source.

— **Rechercher des variables** : Permet aux chercheurs d'utiliser la barre de recherche pour trouver des variables spécifiques.

— **Télécharger les résultats finaux** : Permet aux chercheurs de télécharger les résultats finaux des correspondances dans différents formats tels que CSV, Excel et PDF.

3.4.2 Diagramme de séquence

Le diagramme de séquence ci-dessous illustre les interactions entre les acteurs et l'interface web lors du processus de **normalisation des variables sources et candidates**. Les chercheurs et les utilisateurs de l'interface web, qui agissent en tant qu'acteurs, interagissent avec le système en effectuant diverses actions telles que l'importation des fichiers de données, l'application des mesures de correspondance, la visualisation des résultats, la sélection des variables candidates appropriées, la recherche de variables, le téléchargement des résultats finaux, et bien d'autres.

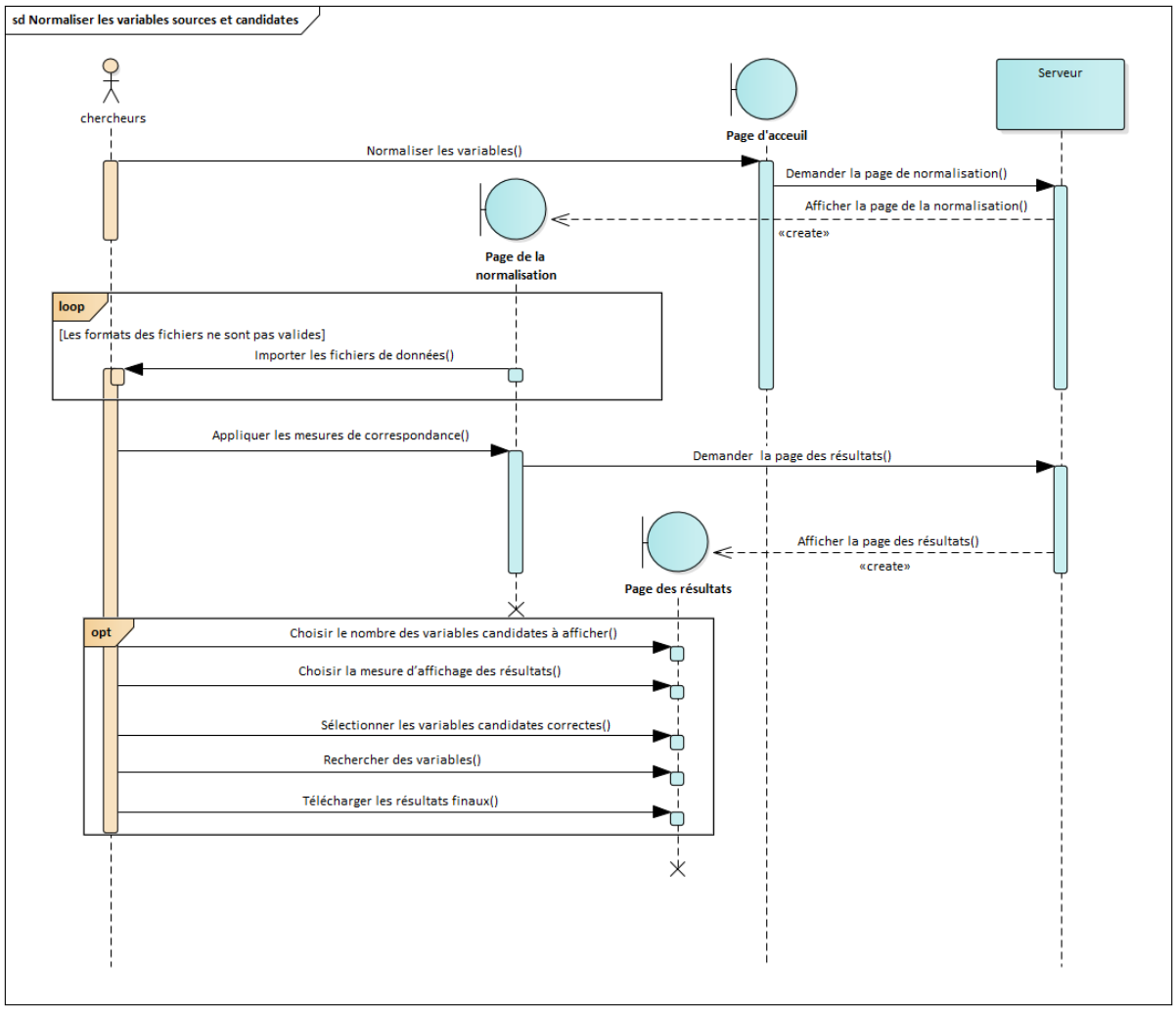

FIGURE 3.3 – Diagramme de séquence

Après avoir expliqué les fonctionnalités de l'interface web, la section suivante présente des captures d'écran de celle-ci pour faciliter la compréhension de son fonctionnement.

3.4.3 Interface web

A. Page d'accueil

L'image suivante représente la page d'accueil de l'interface. Elle offre aux utilisateurs un aperçu de l'application et leur permet de naviguer vers les différentes fonctionnalités disponibles.

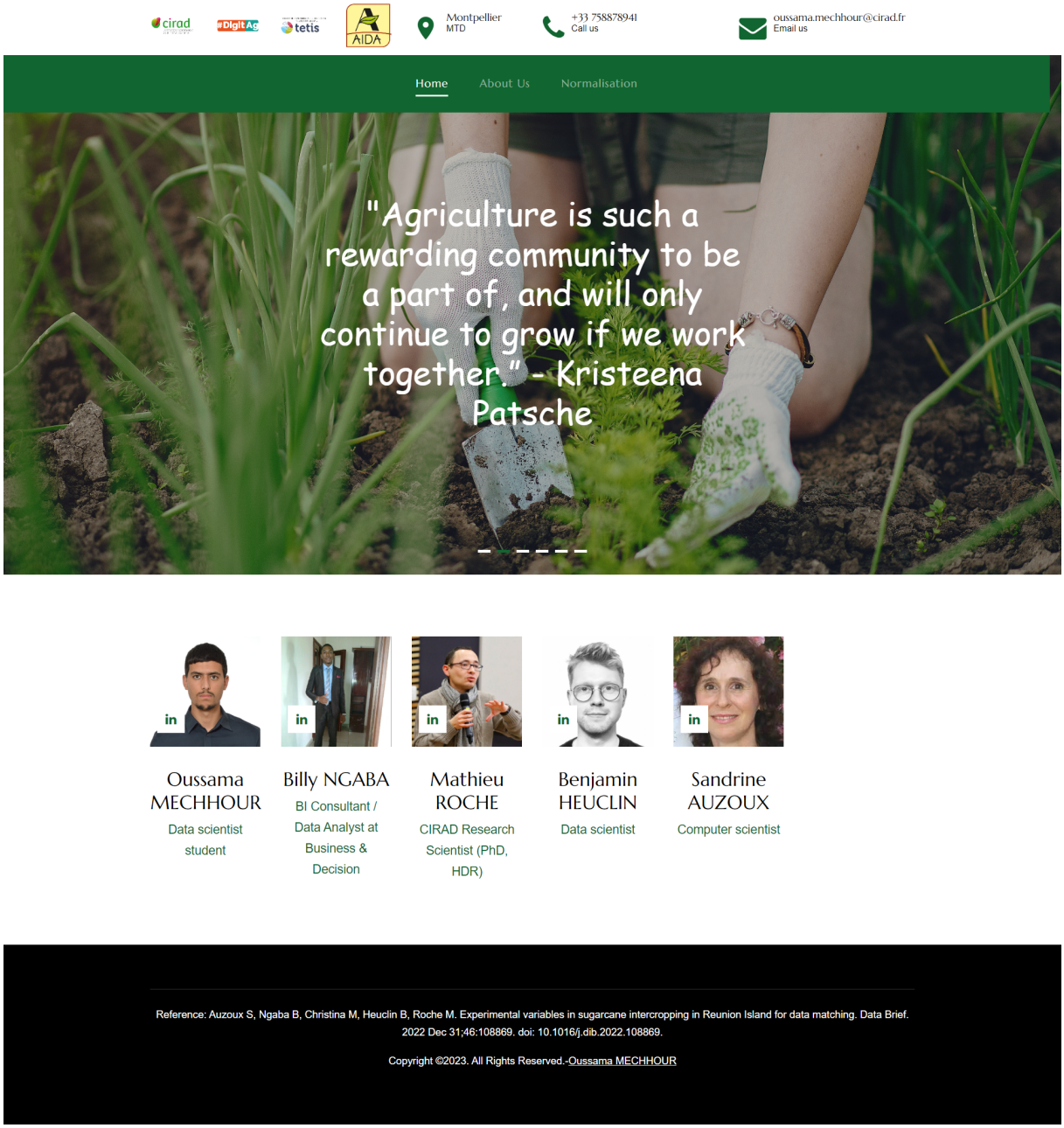

FIGURE 3.4 – Page d'accueil de l'interface web

B. Page de correspondance des variables

Cette image met en évidence la fonctionnalité d'importation des fichiers de données contenant les noms et les descriptions des variables sources et candidates. Les utilisateurs peuvent sélectionner les fichiers à importer et effectuer une vérification de format pour garantir l'intégrité des données. Ensuite, ils peuvent appliquer les mesures de correspondance des variables en cliquant sur le bouton **Matching**.

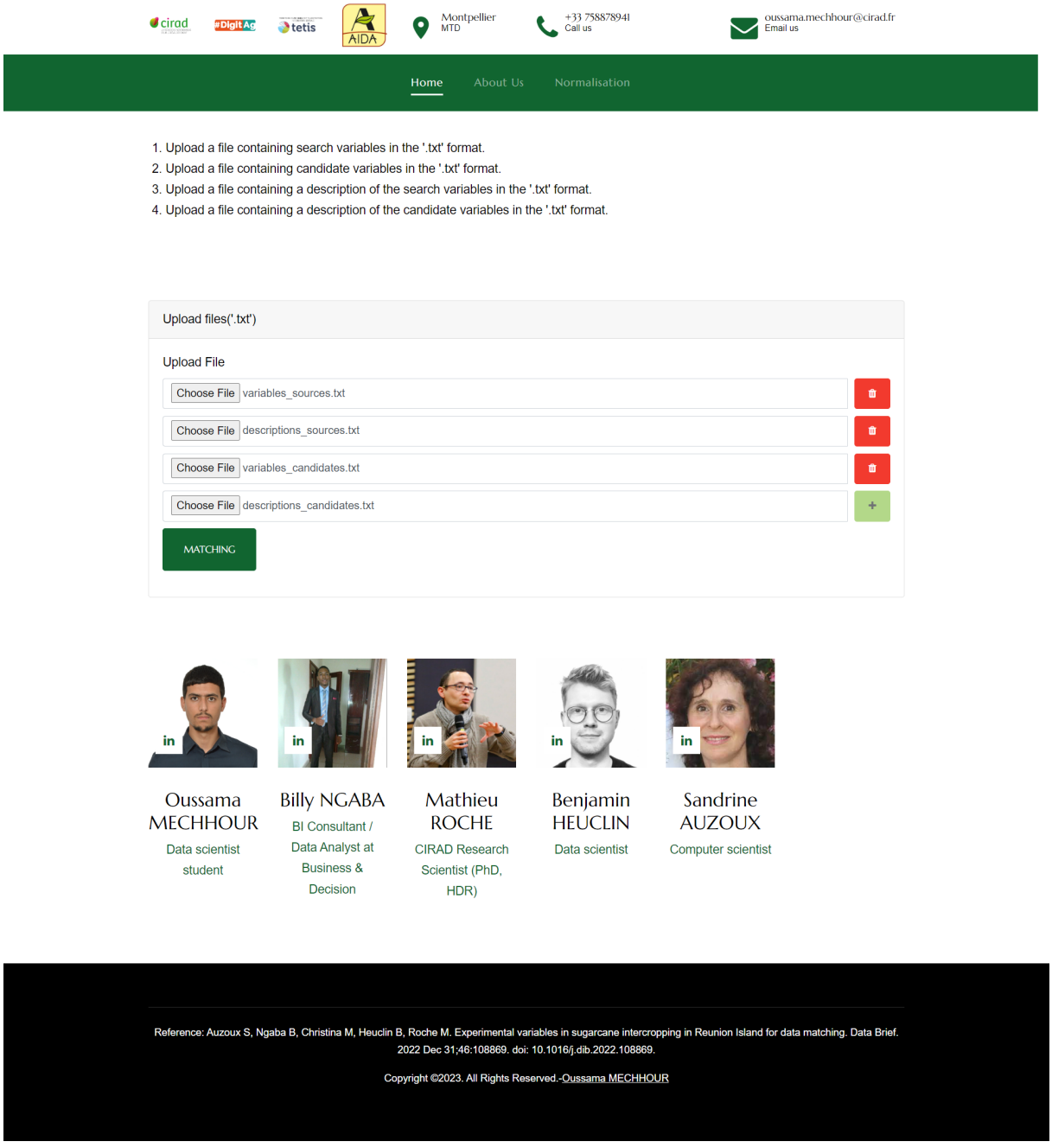

FIGURE 3.5 – Page de correspondance des variables

C. Visualisation des résultats

L'image suivante illustre la fonctionnalité de visualisation des résultats de correspondance entre les variables sources et candidates. Les utilisateurs peuvent explorer les résultats et effectuer différentes actions, telles que la sélection des variables candidates appropriées et le téléchargement des résultats finaux, ainsi que d'autres actions.

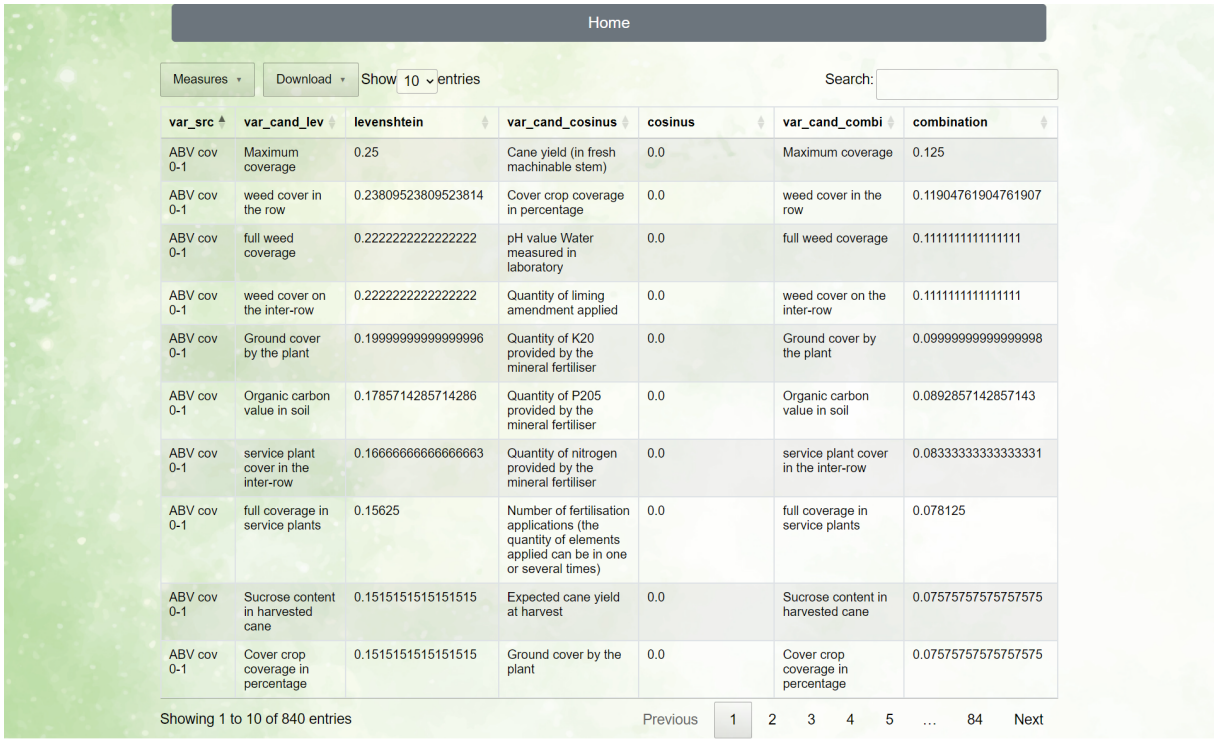

FIGURE 3.6 – Visualisation des résultats

Note : [26] contient le code complet.

Chapitre 4

Résultats et discussion

4.1 Comparaison des résultats avec le stage précédent en 2022

4.1.1 Sans contexte

A. Dans le contexte de notre étude, notre objectif est d'évaluer la similarité entre chaque variable **source** et chaque variable **candidate**. Pour ce faire, nous devons calculer la probabilité de similarité entre ces variables. Cependant, la mesure de similarité utilisée, la distance de **Levenshtein**, est définie dans l'intervalle $[0, \infty[$, ce qui rend difficile son utilisation directe dans la méthode de **combinaison**.

Pour résoudre ce problème, nous devons restreindre la distance de **Levenshtein** à une plage de valeurs entre 0 et 1. [5] a adopté cette approche dans le travail réalisé en utilisant la distance de **Levenshtein** comme mesure de similarité entre les **noms** des variables, le **TF-IDF** pour la vectorisation des **descriptions** des variables, et le **cosinus** comme mesure de similarité entre les **vecteurs de ces descriptions** (Table **4.1.1**).

Nombre de bons résultats après lemmatisation (sur les 84 variables sources)											
Levenshtein				**TF-IDF + cosinus**				**Combinaison**			
p@1	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**
15.48%	19.05%	23.81%	42.86%	33.33%	42.86%	51.19%	60.71%	41.67%	51.19%	64.29%	73.81%

TABLE 4.1 – Résultats précédents (sans contexte) [5]

Afin de normaliser la distance de **Levenshtein** et de la ramener dans l'intervalle $[0, 1]$, [5] a utilisé la technique suivante :

$$Lev(a, b) = 1 - \frac{Levenshtein(a, b)}{\max(\text{longueur}(a), \text{longueur}(b))} [5]$$

La méthode qui nous intéresse est la **combinaison**. La meilleure valeur de α, pour laquelle [5] a obtenu les meilleurs résultats, est $\alpha = 0.3$. Grâce à cette méthode, il existe une probabilité de **73,81%** que la solution pertinente, c'est-à-dire la variable **candidate**

qui correspond réellement à la variable **source**, soit parmi les 10 premières variables **candidates** proposées (la Précision à la position 10).

B. Pour le travail actuel, l'architecture **BERT-base** avec 2 couches cachées a été utilisée, ce qui a donné de meilleurs résultats. Cette configuration a été employée pour la vectorisation des **noms et des descriptions des variables** (Annexe **4.1.3**). La mesure de similarité du **cosinus** a été utilisée pour évaluer les similarités. La méthode de **combinaison** [1] a montré des améliorations significatives dans les résultats obtenus (Table **4.1.1**).

Nombre de bons résultats après lemmatisation (sur les 84 variables sources)											
BERT-base + cosinus (noms des variables)				**BERT-base + cosinus (descriptions des variables)**				**Combinaison**			
p@1	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**
11.90%	28.57%	36.90%	53.57%	33.33%	44.05%	52.38%	57.14%	41.67%	60.71%	69.05%	79.76%

TABLE 4.2 – Résultats actuels (sans contexte)

Les améliorations de précision pour différentes positions dans la méthode de **combinaison** sont les suivantes :

1. La précision à la position **3** a été augmentée de **9,52%**.
2. La précision à la position **5** a été augmentée de **4,76%**.
3. La précision à la position **10** a été augmentée de **5,95%**.

Il a également été remarqué que la similarité entre les **noms** des variables dans les résultats actuels est plus **élevée** que dans les résultats précédents. Un récapitulatif des améliorations est le suivant :

1. La précision à la position **3** a été augmentée de **9,52%**.
2. La précision à la position **5** a été augmentée de **1,09%**.
3. La précision à la position **10** a été augmentée de **10,71%**.

En outre, il a été observé que la similarité entre les **descriptions** des variables dans les résultats actuels est également plus **élevée** que dans les résultats précédents. Un récapitulatif des améliorations est le suivant :

1. La précision à la position **3** et **5** a été augmentée de **1,19%**.

1. La meilleure valeur de α, qui a conduit aux meilleurs résultats dans le cadre de ce rapport, est $\alpha = 0.79$.

4.1.2 Avec contexte

A. Pour le travail précédent, un **corpus** de documents a été utilisé dans [5]. Ce **corpus** est composé d'articles scientifiques, de chapitres d'ouvrages, de rapports et de thèses, axés principalement sur les thèmes de la canne à sucre, de la fertilisation des sols, des plantes de service et des adventices. Il est important de noter que ces documents ont été triés et sélectionnés par **Mme. Sandrine AUZOUX**, et qu'ils sont tous rédigés en anglais. Au total, **122 documents** ont été utilisés, variant en longueur de 5 à 160 pages [5].

Nombre de bons résultats après lemmatisation (sur les 84 variables sources)											
Levenshtein				**TF-IDF + cosinus**				**Combinaison**			
p@1	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**
15.48%	19.05%	23.81%	42.86%	33.33%	42.86%	51.19%	60.71%	44.05%	55.95%	64.29%	73.81%

TABLE 4.3 – Résultats précédents (avec contexte) [5]

Dans un premier temps, [5] a sélectionné les variables **sources** pour lesquelles la précision à la position 10 était **nulle**. Ensuite, il a appliqué la méthode suivante :

1. Il a examiné les descriptions de chaque variable.
2. Dans ces descriptions, il a identifié les termes clés, c'est-à-dire les mots essentiels qui les décrivent.
3. Pour chaque terme clé, il a extrait les n mots précédant et suivant qui l'entourent dans les 122 documents supplémentaires. Par exemple, pour le terme **cane yield**, chaque fois qu'il apparaît dans un document, il a récupéré les n mots à sa gauche et à sa droite.
4. Ensuite, il a sélectionné les m premiers mots les plus fréquents parmi ceux extraits, afin de constituer le corpus associé aux variables sources n'ayant pas atteint une précision de rang 10.

Après avoir expérimenté une trentaine de configurations pour les paramètres (n, m), les valeurs qui ont donné les meilleurs résultats sont $n = 6$, $m = 20$ et $\alpha = 0.3$ (Table **4.1.2**).

B. Pour le travail actuel, **15 articles** scientifiques préparés par **Mme. Sandrine AUZOUX** ont été utilisés. Ces articles ont été regroupés dans un **corpus unique**, également appelé **contexte**. Le **corpus** a été fusionné avec les **noms** et les **descriptions** des variables **candidates** et **sources**, tous prétraités et représentés sous forme d'une liste de chaînes de caractères. Ces données ont ensuite été utilisées par **TF-IDF** pour constituer le vocabulaire (Annexe **4.8**).

TF-IDF (avec contexte) a été utilisé pour la vectorisation des **descriptions** des variables. Le **cosinus** a été utilisé comme mesure de similarité, à la fois entre les **vecteurs de descriptions** et entre les **vecteurs des noms des variables.** Pour la vectorisation des **noms** des variables, l'architecture **BERT-base** avec 2 couches cachées a été employée, ce qui a conduit à des résultats améliorés (Table **4.1.2**).

Nombre de bons résultats après lemmatisation (sur les 84 variables sources)												
BERT-base + cosinus (noms des variables)				TF-IDF + cosinus (descriptions des variables)				Combinaison			
p@1	p@3	p@5	p@10	p@1	p@3	p@5	p@10	p@1	p@3	p@5	p@10
11.90%	28.57%	36.90%	53.57%	29.76%	42.86%	51.19%	64.29%	52.38%	66.67%	71.43%	80.95%

TABLE 4.4 – Résultats actuels (avec contexte)

Il a été constaté que les résultats de la méthode de **combinaison** [2] ont été améliorés comme suit :

1. La précision à la position **1** a été augmentée de **8,33%**.
2. La précision à la position **3** a été augmentée de **10,72%**.
3. La précision à la position **5** et **10** a été augmentée de **7,14%**.

Note : Pour consulter toutes les méthodes utilisées ainsi que leurs résultats, la présentation des résultats au format PowerPoint est disponible dans la bibliographie [22].

4.1.3 Discussion

Différentes méthodes ont été utilisées pour la vectorisation des **noms** de variables (**TF-IDF, BERT-base, BERT-large, RoBERTa** et **XLNet**). Parmi ces méthodes, le modèle **BERT-base** avec 2 couches cachées a donné les meilleurs résultats (Table **4.1.3**).

De même, pour la vectorisation des **descriptions** des variables, plusieurs méthodes ont été explorées. Cependant, les meilleurs résultats ont été obtenus avec l'utilisation de **TF-IDF** (avec l'utilisation de 15 articles). À cet effet, **TF-IDF** a été entraîné sur un **corpus** composé des **noms, des descriptions** des variables et d'un contexte supplémentaire issu de 15 articles (Table **4.1.3**).

BERT-base + cosinus (noms des variables)			
p@1	p@3	p@5	p@10
11.90%	28.57%	36.90%	53.57%

TABLE 4.5 – Les meilleurs résultats obtenus pour la vectorisation des **noms** des variables

TF-IDF + cosinus (descriptions des variables)			
p@1	p@3	p@5	p@10
41.67%	54.76%	60.71%	69.05%

TABLE 4.6 – Les meilleurs résultats obtenus pour la vectorisation des **descriptions** des variables

Combinaison			
p@1	p@3	p@5	p@10
52.38%	66.67%	71.43%	80.95%

TABLE 4.7 – Les meilleurs résultats obtenus pour la **combinaison**

2. La meilleure valeur de α, qui a conduit aux meilleurs résultats dans le cadre de ce rapport, est $\alpha = 0.25$.

Plusieurs combinaisons de méthodes ont été effectuées pour la **vectorisation** des variables. Les meilleurs résultats de ces combinaisons sont présentés en **Table 4.1.3**. Il est important de noter que ces résultats ont été obtenus en utilisant les meilleurs résultats pour la vectorisation des **noms** des variables, mais pas les meilleurs résultats pour la vectorisation des **descriptions** de ces variables.

Cette différence de performances peut s'expliquer par le fait que l'utilisation de probabilités peut conduire à des situations où une faible probabilité coexiste avec une probabilité élevée. Afin d'approfondir cette problématique, plusieurs combinaisons de méthodes ont été réalisées, et les détails ainsi que les résultats complets sont présentés dans une présentation PowerPoint [22]. Cette présentation fournit une vue détaillée des différentes méthodes utilisées et de leurs performances respectives.

Pour mieux comprendre les explications ci-dessus, les deux tableaux ci-dessous les illustrent :

Nombre de bons résultats après lemmatisation (sur les 84 variables sources)											
BERT-base + cosinus (noms des variables)				**TF-IDF + cosinus (descriptions des variables)**				**Combinaison**			
p@1	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**
11.90%	28.57%	36.90%	53.57%	41.67%	54.76%	60.71%	69.05%	52.38%	61.90%	66.67%	79.76%

TABLE 4.8 – La méthode de combinaison (meilleurs résultats de la vectorisation des **noms** et des **descriptions** des variables)

Nombre de bons résultats après lemmatisation (sur les 84 variables sources)											
BERT-base + cosinus (noms des variables)				**TF-IDF + cosinus (descriptions des variables)**				**Combinaison**			
p@1	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**	**p@1**	**p@3**	**p@5**	**p@10**
11.90%	28.57%	36.90%	53.57%	29.76%	42.86%	51.19%	64.29%	52.38%	66.67%	71.43%	80.95%

TABLE 4.9 – La méthode de combinaison (meilleurs résultats de la vectorisation des **noms**, mais pas les meilleurs résultats de la vectorisation des **descriptions** des variables)

Conclusion et perspectives

Pour conclure, au cours de ce stage, plusieurs méthodes ont été appliquées, notamment **TF-IDF, BERT-base, BERT-large, RoBERTa** et **XLNet**, pour la vectorisation des **noms** et des **descriptions** des variables. Les résultats obtenus ont largement dépassé les résultats précédents [5]. De plus, des méthodes de mesure de similarité telles que **Levenshtein** et le **cosinus** ont été utilisées pour évaluer la proximité entre les variables. Cependant, malgré ces avancées, il reste encore des pistes d'amélioration à explorer.

Au cours de ce stage, plusieurs problématiques ont été identifiées. Tout d'abord, il y a un nombre limité de variables en anglais (**84 variables**), et ces variables sont souvent formulées de manière **non canonique**, c'est-à-dire de façon **non formelle** ou **non structurée**. De plus, certaines variables sont en français (non utilisées dans ce stage). De plus, les **descriptions** des variables sont très courtes, ce qui affecte les performances des modèles de langues basés sur le contexte. De plus, les **ontologies** associées à ces variables n'ont pas été abordées dans le cadre du stage.

Pour résoudre ces problèmes, une thèse a été construite en tant que prolongement du travail réalisé au cours de ce stage, et j'ai été accepté en tant que doctorant pour cette thèse. Elle sera encadrée par **Mme. Sandrine AUZOUX**, **M. Mathieu ROCHE** et **M. Clement Jonquet**. Nous divisons les perspectives en deux types, à court terme et à moyen terme, compte tenu de la fin du stage prévue le 31 juillet.

A. Perspectives à court terme

- Identifier des contextes issus du web crawling et des modèles génératifs (par exemple, ChatGPT).
- Réaliser des expérimentations complémentaires avec d'autres jeux de données pour étudier la généricité des propositions.

B. Perspectives à moyen terme

Dans la première phase, qui consiste à construire un **corpus** (contexte) :

1. Traduire les variables (noms et descriptions) dans la même langue que les ontologies afin de ne pas perdre les informations sémantiques associées à ces ontologies.
2. Transformer les variables (noms et descriptions) non canoniques en variables canoniques.
3. Effectuer une tokenisation des noms des variables.
4. Construire un corpus à partir des données mentionnées dans les étapes précédentes et d'ontologies.

Dans la deuxième phase, qui concerne **l'extension du contexte** :

1. Bénéficier d'autres corpus hétérogènes et multilingues dans le domaine de l'agroécologie. Pour cela, une traduction vers la même langue que les ontologies sera envisagée.
2. Utiliser des modèles génératifs pour générer d'autres contextes similaires au contexte réalisé dans la première phase afin de résoudre le problème des données de petite taille.
3. Construire un corpus étendu à partir du contexte réalisé dans la première phase et des deux étapes de cette phase.

Dans la troisième phase :

1. Utiliser des méthodes de plongement de mots telles que **BERT, TF-IDF**, etc., ou affiner le modèle **BERT**.
2. Utiliser des mesures de similarité telles que le **cosinus**.

Ces propositions permettront d'adresser les problématiques identifiées et d'améliorer davantage les résultats dans de futurs travaux.

Pour avoir une vue d'ensemble de notre proposition, veuillez consulter la Figure ci-dessous :

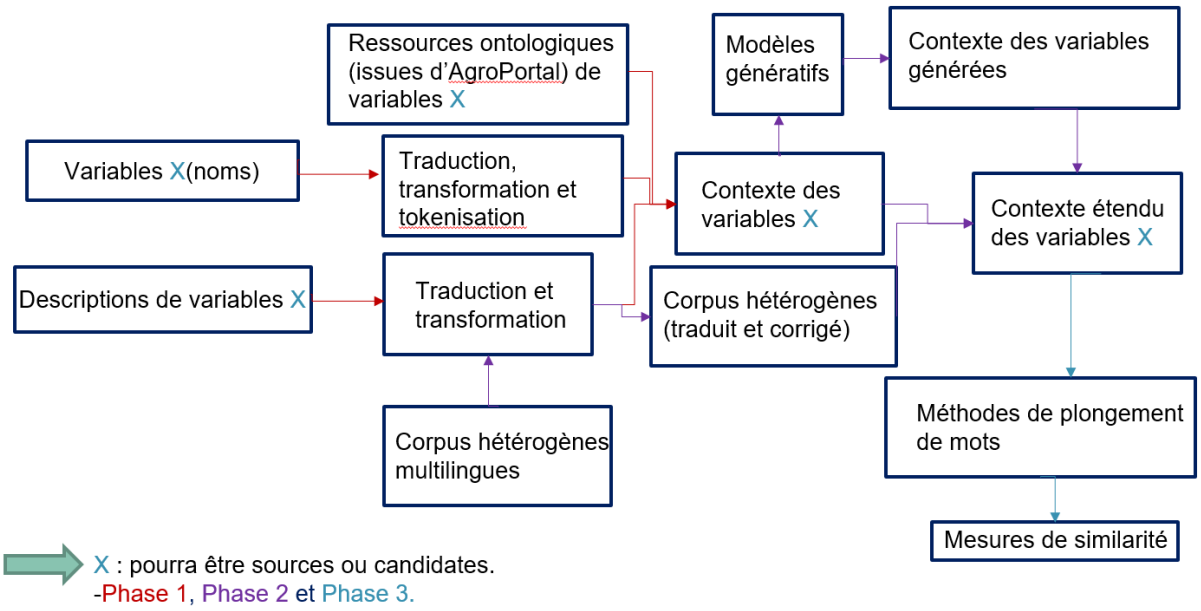

FIGURE 4.1 – Notre proposition pour améliorer encore les résultats.

Afin de conclure, Pendant ce stage, j'ai eu l'opportunité d'approfondir mes connaissances dans plusieurs domaines clés. Voici quelques éléments marquants de cette expérience :

1. Participation au séminaire #DigitAg : Durant trois jours, j'ai pu assister à ce séminaire où j'ai pu échanger avec des chercheurs, des doctorants et des post-doctorants. Ce fut l'occasion d'explorer les applications réelles de l'intelligence artificielle en agroécologie.
2. Présentation lors du stagierothon : J'ai également participé à cet événement où j'ai pu présenter mes travaux et discuter des avantages et des inconvénients de mon travail. Cela m'a donné une vision globale des projets réalisés par les autres stagiaires, ce qui m'a permis de découvrir les différentes applications de l'intelligence artificielle dans le domaine de l'agroécologie.
3. Approfondissement de mes connaissances en TALN : J'ai saisi l'occasion pour approfondir mes connaissances en Traitement Automatique du Langage Naturel (TALN) en lisant des articles scientifiques et en participant à des présentations.

Dans l'ensemble, ce stage a été une période cruciale de ma vie durant laquelle j'ai pu développer mes connaissances, élargir ma culture et affirmer ma personnalité. J'ai acquis de nouvelles compétences et une meilleure compréhension des domaines de l'agroécologie et de l'intelligence artificielle.

Bibliographies

Bibliographie

[1] Salton, G., & McGill, M. J. (1986). Introduction to Modern Information Retrieval. McGraw-Hill.

[2] Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8), 707-710. doi :10.1007/BF02291170

[3] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

[5] Ngaba, B., "Rapport de stage", Couplage d'un modèle de culture avec une plateforme de capitalisation des données issues d'agroécosystèmes à La Réunion, [En ligne]. Disponible : `https://agritrop.cirad.fr/601877/`.

[6] Muller, B., Sagot, B., & Seddah, D. (2019). Enhancing BERT for Lexical Normalization.

[7] Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing (3rd Edition). Pearson.

[8] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[9] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

[10] Tracz, J., Wojcik, P., Jasinska-Kobus, K., Belluzzo, R., Mroczkowski, R., & Gawlik, I. (2023). BERT-based similarity learning for product matching. ML Research at Allegro.pl.

[11] Petrova-Antonova, D., & Tancheva, R. (2023). Data Cleaning : A Case Study with OpenRefine and Trifacta Wrangler. Sofia University "St. Kl. Ohridski", GATE Institute, Sofia, Bulgaria.

[12] Ngaba, B., "variables_sources.txt", fichier des noms des variables sources, [En ligne]. Disponible : `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[13] Ngaba, B., "descriptions_sources.txt", fichier des descriptions des variables sources, [En ligne]. Disponible : `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[14] Ngaba, B., "variables_candidates.txt", fichier des noms des variables candidates, [En ligne]. Disponible : `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[15] Ngaba, B., "descriptions_candidates.txt", fichier des descriptions des variables candidates, [En ligne]. Disponible : `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[16] Ngaba, B., "Correspondances.csv", fichier de correspondances, [En ligne]. Disponible : `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[17] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT : Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Vol. 1, pp. 4171-4186).

[18] Hugging Face *Pretrained Models - Hugging Face Transformers Documentation* Disponible sur : `https://huggingface.co/transformers/v2.5.1/pretrained_models.html`

[19] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet : Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv :1906.08237*.

[20] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv :1907.11692*.

[21] Baevski, A. L., et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv :1910.10683 [cs], oct. 2019. Disponible sur : `https://arxiv.org/abs/1910.10683`.

[22] Mechhour, O., "Résultats obtenus", présentation des résultats, [En ligne]. Disponible : `https://drive.google.com/drive/folders/1XCWIqtdDOZoe5EFISOxhQOnSLwJQGWHd?usp=sharing`.

[23] Auzoux, S., Bernard, M., Courtonne, J.Y., et al. (2019), *AEIGIS : An open online platform to store, manage and process data from agroecological experiments in the global South*, Ecological Informatics, 50, 135-145.

[24] Mechhour, O., "*BERT-base.py*". Code source. Disponible sur `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[25] Mechhour, O., "*BERT-base et TF-IDF.py*". Code source. Disponible sur `https://github.com/OussamaMECHHOUR/Master-s_internship.git`.

[26] Mechhour, O., *User_ Interface*. Code source. Disponible sur `https://drive.google.com/drive/folders/1XCWIqtdDOZoe5EFISOxhQOnSLwJQGWHd?usp=sharing`.

Annexes

Exemples de données avant et après le prétraitement et codes

A. Clean_text()
Voici le code de la fonction clean_text() :

```python
def clean_text(texte):
    # Supprimer les nombres
    texte = re.sub(r'\d+', '', texte)

    # Supprimer les parenthèses et leur contenu
    texte = re.sub(r'\([^()]*\)', '', texte)

    # Supprimer les autres caractères spéciaux
    texte = re.sub(r'[^a-zA-Z0-9\s]', '', texte)

    # Convertir en minuscules
    texte = texte.lower()

    # Retourner le texte nettoyée
    return texte
```

FIGURE 4.2 – Code de la fonction clean_text()

Avant le prétraitement	**Après le prétraitement**
dry root biomass **(plant scale)**	dry root biomass
maximum recovery rate of the plant **(or species)**	maximum recovery rate of the plant
sum heights tvd live stems **(>20cm) (height profile)**	sum heights tvd live stems
...	...

TABLE 4.10 – Exemples des descriptions des variables candidates avant et après le prétraitement (clean_text())

B. Remove_stopwords()

Voici le code de la fonction remove_stopwords() :

```python
def remove_stopwords(texte):
    stop_words = set(stopwords.words('english'))  # Choisissez la langue appropriée
    words = texte.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    return ' '.join(filtered_words)
```

FIGURE 4.3 – Code de la fonction remove_stopwords()

Avant le prétraitement	Après le prétraitement
measurement **of** dry stem biomass **at** plot level Height **of the** apex **of the** sugar stem sample sum **of** tvd heights ...	measurement dry stem biomass plot level Height apex sugar stem sample sum tvd heights ...

TABLE 4.11 – Exemples des descriptions des variables candidates avant et après le prétraitement (remove_stopwords())

C. Lemmatize()

Voici le code de la fonction lemmatize() :

```python
def lemmatize(texte):
    lemmatizer = WordNetLemmatizer()
    words = texte.split()
    lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
    return ' '.join(lemmatized_words)
```

FIGURE 4.4 – Code de la fonction lemmatize()

Avant le prétraitement	Après le prétraitement
total number leaf **blades** tracking labeled **stems** soluble sugar concentration **leaves** ...	total number leaf **blade** tracking labeled **stem** soluble sugar concentration **leaf** ...

TABLE 4.12 – Exemples des descriptions des variables candidates avant et après le prétraitement (lemmatize())

D. Remove_punctuation()
Voici le code de la fonction remove_punctuation() :

```
def remove_punctuation(texte):
    translator = str.maketrans('', '', string.punctuation)
    texte = texte.translate(translator)
    return texte
```

FIGURE 4.5 – Code de la fonction remove_punctuation()

E. Replace_synonyms()
Voici le code de la fonction replace_synonyms() :

```
def replace_synonyms(texte):
    words = texte.split()
    replaced_words = []
    for word in words:
        synonyms = wordnet.synsets(word)
        if synonyms:
            replaced_words.append(synonyms[0].lemmas()[0].name())
        else:
            replaced_words.append(word)
    return ' '.join(replaced_words)
```

FIGURE 4.6 – Code de la fonction replace_synonyms()

Avant le prétraitement	Après le prétraitement
measurement root dry biomass plot **level**	measurement root dry biomass plot **degree**
stem juice **yield**	stem juice **output**
Height **apex** sugar **stem** sample	Height **vertex** sugar **root** sample
...	...

TABLE 4.13 – Exemples des descriptions des variables candidates avant et après le prétraitement (replace_synonyms)

F. Code BERT-base :

```python
import torch
from transformers import BertTokenizer, BertModel, BertConfig

# Vérifier si un GPU est disponible
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

config = BertConfig(num_hidden_layers=2)

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased', config=config)
model = BertModel.from_pretrained('bert-base-uncased', config=config).to(device)

src_inputs = tokenizer(lines_des_src_pre5, padding=True, truncation=True, return_tensors="pt").to(device)
src_outputs = model(**src_inputs)

cand_inputs = tokenizer(lines_des_cand_pre5, padding=True, truncation=True, return_tensors="pt").to(device)
cand_outputs = model(**cand_inputs)

similarity_scores = torch.cosine_similarity(src_outputs.last_hidden_state.mean(dim=1).unsqueeze(1),
                                            cand_outputs.last_hidden_state.mean(dim=1).unsqueeze(0), dim=2)



src_inputs_leven = tokenizer(lines_var_src_pre3, padding=True, truncation=True, return_tensors="pt").to(device)
src_outputs_leven = model(**src_inputs_leven)

cand_inputs_leven = tokenizer(lines_var_cand_pre3, padding=True, truncation=True, return_tensors="pt").to(device)
cand_outputs_leven = model(**cand_inputs_leven)

similarity_scores_leven = torch.cosine_similarity(src_outputs_leven.last_hidden_state.mean(dim=1).unsqueeze(1),
                                                  cand_outputs_leven.last_hidden_state.mean(dim=1).unsqueeze(0), dim=2)
```

FIGURE 4.7 – Code pour la vectorisation des noms et des descriptions des variables en utilisant BERT-base et le calcul de la similarité entre ces vecteurs en utilisant le cosinus.

Note : [24] contient le code complet.

G. Code du TF-IDF en utilisant un corpus :

```python
################################

# Vocabulaire pré-traité
vocab_sans_ponctuations = lines_des_cand_pre5 + lines_var_cand_pre3 + contexte_pre5 + lines_des_src_pre5 + lines_var_src_pre3#+ l
```

```python
vectorizer = TfidfVectorizer(stop_words='english')
esp_vec = vectorizer.fit(vocab_sans_ponctuations)
des_src_vect = esp_vec.transform(lines_des_src_pre5)
#var_cand_vect = esp_vec.transform(lines_var_cand)
des_cand_vect = esp_vec.transform(lines_des_cand_pre5)
```

FIGURE 4.8 – Code TF-IDF avec contexte

Note : [25] contient le code complet.