MAEVa : Une approche hybride pour la mise en relation des variables expérimentales agroécologiques

Oussama Mechhour[1,2,3,4*], Sandrine Auzoux[1,2], Clément Jonquet[5,6], Mathieu Roche[3,4]

[1] CIRAD, UPR AIDA, F-34398 Montpellier, France
[2]AIDA, CIRAD, Université de Montpellier, Montpellier, France
[3] CIRAD, UMR TETIS, F-34398 Montpellier, France
[4] TETIS, Université de Montpellier, AgroParisTech, CIRAD, INRAE, Montpellier, France
[5] MISTEA, Univ. de Montpellier, INRAE, Institut Agro, Montpellier, France
[6] LIRMM, Univ. de Montpellier, CNRS, Montpellier, France
[*]Corresponding author: oussama.mechhour@cirad.fr

Résumé

Les variables sources ou les propriétés observables utilisées pour décrire les expérimentations agroécologiques sont hétérogènes, non standardisées et multilingues, rendant leur compréhension, explication et utilisation difficiles dans la modélisation des systèmes de culture et les évaluations multicritères de la performance des systèmes agroécologiques. L'annotation des données via un vocabulaire contrôlé, appelé variables candidates de Agroecological Global Information System (AEGIS), constitue une solution. Les mesures de similarité textuelle jouent un rôle clé dans la désambiguïsation du sens des mots, l'appariement de schémas dans les bases de données et l'annotation des données. Les approches courantes incluent (a) la similarité fondée sur les chaînes de caractères, (b) sur le corpus, (c) sur les connaissances et (d) les approches hybrides combinant deux ou plusieurs de ces méthodes. Ce travail propose une approche hybride, Matching Agroecological Experiment Variables (MAEVa), visant à apparier les variables sources et candidates selon (1) l'appariement des noms, (2) celui des descriptions, (3) une combinaison linéaire de (1) et (2), et (4) une méthode de sélection des résultats pour l'évaluation finale. Pour l'appariement des noms, nous étendons BERT-base avec une couche d'attention multi-têtes externe (BERTmha). Pour les descriptions, nous enrichissons celles existantes avec l'API GPT-3.5 Turbo et utilisons TF-IDF pour la représentation vectorielle. Nos résultats montrent que BERTmha améliore la précision de plus de 11% par rapport à BERT-base seul et que notre corpus améliore celle de TF-IDF de plus de 4%. Notre évaluation (étape 4) montre que MAEVa atteint une précision de plus de 66% de $P@1$ à $P@10$.

Mots-clés

Propriétés observables, Similarité fondée sur les chaînes de caractères, Similarité fondée sur les corpus, Similarité hybride, Modèles de langage pré-entraînés (PLMs), Grands modèles de langage (LLMs)

Abstract

Source variables or observable properties used to describe agroecological experiments are heterogeneous, non-standardized, and multilingual, making them challenging to understand, explain, and use in cropping system modeling and multicriteria evaluations of agroecological system performance. Data annotation via a controlled vocabulary, known as candidate variables from the Agroecological Global Information System (AEGIS), offers a solution. Text similarity measures play crucial roles in tasks such as word sense disambiguation, schema matching in databases, and data annotation. Commonly used measures include (a) string-based, (b) corpus-based, (c) knowledge-based, and (d) hybrid-based similarity, which combines two or more of these methods. This work introduces a hybrid approach called Matching Agroecological Experiment Variables (MAEVa), designed to match source and candidate variables based on (1) matching variable names, (2) matching variable descriptions, (3) a combination of (1) and (2) via a linear function, and (4) a method for selecting results for the final evaluation. For matching variable names, we propose a novel approach that extends BERT-base with an external multi-head attention layer (BERTmha). For matching variable descriptions, we augment existing descriptions using GPT-3.5 Turbo API to provide richer contextual information and employ TF-IDF to construct the vector space. Our experimental results demonstrate that BERTmha improves the precision of matching variable names by more than 11% compared to BERT-base alone, and that our constructed corpus enhances TF-IDF-based matching by more than 4%. Our evaluation (step 4) shows that MAEVa achieves a precision of over 66% from $P@1$ to $P@10$.

Keywords

Observable properties, String-based similarity, Corpus-based similarity, Hybrid-based similarity, Pre-trained language models (PLMs), Large language models (LLMs)

1 Introduction

Les expérimentations agroécologiques génèrent des bases de données pouvant être multi-échelles (plante, système de culture, exploitation, paysage, territoire), multi-espèces (cultures, plantes associées, adventices, plantes fourragères) et multidisciplinaires (agronomie, malherbologie, entomologie, sciences économiques et sociales, environnement, informatique). Les variables sources ou propriétés observables [1] utilisées pour décrire les expérimentations agroécologiques sont souvent nommées et décrites à l'aide d'homonymes, de synonymes, d'acronymes, de multiples langues et parfois de termes non standardisés, ce qui peut rendre leur interprétation et leur explication complexes. Elles sont également mesurées avec des unités hétérogènes et présentent une hétérogénéité d'un point de vue linguistique, structurel, sémantique, syntaxique et taxonomique. Cette complexité les rend difficiles à utiliser dans la modélisation des systèmes de culture et l'évaluation multicritère de la performance des systèmes agroécologiques. Pour relever ces défis, l'annotation des données via un vocabulaire contrôlé ou une ontologie commune constitue une solution. Afin d'harmoniser ces variables sources non standardisées et hétérogènes, le centre de Coopération Internationale en Recherche Agronomique pour le Développement (CIRAD) a développé *Agroecological Global Information System* (AEGIS) [1]. AEGIS intègre une chaîne harmonisée d'acquisition et de traitement des données utilisant un ensemble de variables candidates, combinant des termes sémantiques issus d'ontologies de référence (ontologie des plantes, des cultures, de l'environnement et de l'agronomie) et des connaissances expertes en agroécologie. L'approche adoptée dans AEGIS permet aux chercheurs d'utiliser librement leurs propres noms de variables sources, descriptions et unités de mesure, tout en fournissant une liste de variables candidates pour harmoniser et standardiser ces variables sources.

Dans ce travail, nous avons utilisé des variables sources hétérogènes collectées lors d'expérimentations sur la canne à sucre associée à des plantes de service, issues des projets CanecoH [2], Ecocanne [3], et AgriecoH, conduits par eRcane [4] en collaboration avec le CIRAD [5] à La Réunion.

La Table 1 présente des exemples de variables sources, et la Table 2 illustre des exemples de variables candidates. Ces tableaux mettent en évidence la complexité de ces variables. Par exemple, dans la Table 1, le premier nom de variable est multilingue : il contient *"yield"* en anglais et *"CAS"*, une abréviation française pour *"canne à sucre"*. Le second nom de variable est en français, et différentes manières d'écrire les unités de mesure existent. Par exemple, les deux notations suivantes représentent une division différemment : *"t.ha^{-1}"* et *"kg/m^2"*. Dans la Table 2, une caractéristique supplémentaire, *"méthode de calcul"*, est ajoutée, alors qu'elle est absente de la Table 1. Les lignes 2 et 3 de la Table 1 se réfèrent à la même variable que la ligne 3 de la Table 2, bien que leurs descriptions soient lexiquement différentes, elles reflètent néanmoins la même information.

Comme illustré dans les Tables 1 et 2, les variables sont principalement exprimées sous forme de descriptions textuelles, ce qui rend possible l'utilisation de mesures de similarité textuelle pour la mise en lien des variables sources et candidates. La mesure de similarité textuelle est un concept fondamental en théorie de l'information et est utilisée pour évaluer la proportion de contenu partagé entre deux textes [16]. Un score de similarité élevé indique une forte proximité entre les textes. La similarité textuelle consiste à évaluer la proximité entre deux textes en tenant compte à la fois des similarités lexicales et sémantiques. Cela signifie que même si deux textes ne partagent pas les mêmes mots, ils peuvent néanmoins véhiculer une sémantique proche. Ce concept est devenu central dans plusieurs domaines du traitement automatique des langues naturelles (TALN), notamment les systèmes de questions-réponses automatiques [11], la traduction automatique [30] et l'appariement de documents [23]. Les mesures de similarité textuelle courantes [6] peuvent être regroupées en quatre catégories : (1) les mesures fondées sur les chaînes de caractères (lexicales), (2) les mesures qui s'appuient sur les corpus, (3) les mesures fondées sur les connaissances et (4) les mesures hybrides.

Les mesures de similarité basées sur les chaînes de caractères évaluent la ressemblance entre deux textes en se basant uniquement sur la comparaison littérale des mots ou des caractères, sans prendre en compte les synonymes, le contexte ou les relations sémantiques entre les mots [9]. **Les mesures de similarité fondées sur les corpus** utilisent des informations dérivées d'un corpus pour calculer la similarité, en exploitant des caractéristiques textuelles ou des probabilités de cooccurrence. Par exemple, *TF–IDF* (*Term Frequency–Inverse Document Frequency*) [12] est une mesure qui se concentre principalement sur les aspects quantitatifs des mots dans un texte, sans prendre en compte l'ordre des mots, le contexte ou les relations sémantiques entre eux. Une autre approche est la représentation des mots sous forme de vecteurs denses, qui peuvent être statiques ou dynamiques. Les vecteurs statiques, tels que ceux produits par Word2Vec [18], FastText [13] et GloVe [21], sont indépendants du contexte. En revanche, les vecteurs dynamiques, tels que ceux produits par BERT [2], capturent les variations contextuelles des mots. **Les mesures de similarité basées sur les connaissances** reposent sur des réseaux structurés de concepts sémantiquement connectés pour évaluer la similarité entre les mots et peuvent être étendues à l'analyse au niveau des phrases. Ces réseaux sont souvent spécifiques à un domaine (par exemple, la biomédecine ou le droit) ou génériques, comme WordNet, qui est largement utilisé pour la mesure de similarité basée sur les connaissances [6]. **Les mesures de similarité hybrides** combinent deux ou plusieurs de ces approches de mesure de similarité

1. Une propriété observable est la description d'un élément observé ou dérivé.
2. https://ecophytopic.fr/dephy/concevoir-son-systeme/projet-canecoh
3. https://umr-pvbmt.cirad.fr/recherche/principaux-projets/ecocanne
4. https://www.ercane.re/en/home/
5. https://www.cirad.fr/

TABLE 1 – Exemples de variables sources. Chaque nom est une combinaison du nom de la variable et de son unité de mesure, avec l'unité apparaissant après le dernier underscore.

Noms	**Descriptions**	**Unités de mesure**
Yield_CAS_t.ha^{-1}	Cane yield (in fresh machinable stem)	t.ha^{-1}
Rec_globale_plein_%	Full weed and service plant coverage	%
Cov_end_CP1_%	Cover plant 1 coverage at the end de the trial	%
DM_end_weed_kg/m^2	Weed aerial dry mass at the end de the trial	kg/m^2
...	...	...

TABLE 2 – Exemples de variables candidates. Chaque nom est une combinaison du nom de la variable et de son échelle, avec l'échelle apparaissant après le dernier underscore.

Noms	**Descriptions (Traits)**	**Échelles**	**Méthodes de calcul**
leaf_plant_dm_kg	Measurement de foliar dry biomass at the individual level (plant scale)	kg	Common measurement method
abv_om_dm_content_%	Organic matter concentration de the WAB	%	Organic matter concentration defined based on dry matter concentration and mineral concentration
plant_ground_cover_%	Measurement de plant (or species) recovery by ceptometry	%	Common measurement method
fruit_plant_fm_kg	Measurement de fresh fruit biomass at the individual level (plant scale)	kg	Common measurement method
...	...	...	...

textuelle afin d'exploiter leurs forces respectives et d'améliorer la précision des mesures de similarité.

Cette étude introduit une approche hybride appelée "Matching Agroecological Experiment Variables" (MAEVa), conçue pour mettre en correspondance les variables sources et candidates selon (1) la mise en lien des noms de variables, (2) la mise en lien des descriptions de variables, (3) une combinaison de (1) et (2) via une fonction linéaire, et (4) une méthode de sélection des résultats pour l'évaluation finale.

Les principales contributions de cet article sont :

1. Une extension de BERT avec une couche d'attention multi-têtes externe (BERTmha) pour l'appariement des noms de variables.
2. Une augmentation des données via GPT-3.5 Turbo et l'utilisation de TF-IDF pour l'appariement des descriptions.
3. MAEVa, une approche hybride et générique adaptable à toute mesure de similarité textuelle, technique de construction de corpus et méthode de combinaison.

La suite de cet article est structurée comme suit. La Section 2 présente un état de l'art des mesures de similarité textuelle, en mettant en évidence leurs limites et ce que nous proposons dans nos travaux. La Section 3 introduit notre approche hybride MAEVa avec des expérimentations, résultats et une discussion des résultats détaillés en Section 4. Enfin, la Section 5 conclut l'article avec un résumé, les limites de notre travail et des perspectives.

2 État de l'art

Dans cette section, nous nous concentrerons sur les travaux connexes concernant les mesures de similarité fondées sur les corpus et les mesures de similarité hybrides.

2.1 Mesures de similarité basées sur les corpus

Les mesures de similarité basées sur les corpus ont été largement appliquées dans divers travaux. [20] a comparé TF–IDF (Baseline) [12], FastText [13], Doc2Vec [14], BERT [2] et ADA [8] dans des tâches de recommandation et l'analyse de rapports de bugs. [20] a démontré que BERT surpassait généralement les autres modèles en termes de rappel, suivi d'ADA, Doc2Vec, FastText et TF–IDF. Dans une autre étude, [7] a comparé TF–IDF, Doc2Vec, BERT et sentence-BERT (SBERT) [24] avec la similarité cosinus pour mesurer la similarité entre des thèses et mémoires académiques. Les résultats ont montré que TF–IDF surpassait les autres méthodes en termes de précision, rappel, score F1 et temps de traitement, suivi de Doc2Vec, BERT et SBERT.

2.2 Mesures de similarité hybrides

Deux ou plusieurs mesures de similarité textuelle, telles que les méthodes fondées sur les chaînes de caractères, les corpus et les connaissances, peuvent être combinées afin d'exploiter les avantages de chacune et d'améliorer la précision des mesures de similarité. Plusieurs études ont exploré ces approches. [4] a proposé une approche hybride appelée mesure de similarité sémantique agrégée (ASSM) pour comparer deux mots. La similarité sémantique entre deux mots (ou paires de mots) correspond à la similarité maximale obtenue à partir de deux méthodes : (1) la similarité cosinus entre les vecteurs obtenus avec Word2Vec préentraîné de Google, basé sur un corpus de 100 milliards de mots du Google News dataset, et (2) une fonction linéaire combinant la similarité cosinus de Word2Vec avec la similarité de Wu & Palmer [31]. [3] a comparé ELMo [22], les embeddings de bases de connaissances (KBs) via ComplEx [29] et la concaténation de ces méthodes pour les tâches de typage d'entités et de typage de relations, qui sont des tâches de classification multiclasse. Enfin, [27] a analysé l'efficacité de la similarité cosinus appliquée à plusieurs méthodes d'embedding de mots pour capturer la similarité lexicale entre des paires de mots de cinq ensembles de données différents. Les auteurs ont évalué Word2Vec [18], GloVe [21], FastText [13], LexVec [25] et ConceptNetNumberbatch, un modèle hybride combinant GloVe, Word2Vec et des réseaux sémantiques tels que ConceptNet [26] et PPDB [5].

2.3 Comparaison des approches

Nous avons résumé diverses méthodes et travaux existants pour mesurer la similarité textuelle. Chaque méthode présente ses propres avantages et inconvénients. Par exemple, les *approches fondées sur la fréquence ou les statistiques* traitent deux mots ayant une sémantique proche de manière indépendante et ne tiennent pas compte de l'ordre des mots. En raison de leur nature purement statistique, elles peuvent présenter certaines limites, telles que la *malédiction de la dimensionnalité*, où les vecteurs deviennent trop volumineux pour être gérés efficacement, ou encore des problèmes de *mots hors vocabulaire* (*out-of-vocabulary* - OOV) lorsque les mots dans de nouveaux textes ne sont pas présents dans le corpus d'origine. Les *mesures de similarité fondées sur une fenêtre contextuelle restreinte*, telles que Word2Vec, FastText et GloVe, qui utilisent une fenêtre de contexte limitée et attribuent un unique vecteur dense à chaque mot, présentent également des limites : quel que soit le contexte d'un mot, sa représentation vectorielle reste identique. À l'inverse, des méthodes comme BERT, RoBERTa et XLNet prennent en compte le contexte en générant plusieurs vecteurs pour un même mot en fonction de son environnement textuel immédiat, ce qui améliore la représentation sémantique du contexte. Cependant, ces méthodes sont moins efficaces lorsqu'un contexte étendu est nécessaire pour comprendre pleinement le sens du texte ou du mot. Pour exploiter les forces de chaque méthode tout en surmontant leurs limites, les *mesures de similarité hybrides* ont été introduites. Ces mesures hybrides combinent souvent des méthodes telles qu'elles sont, sans modification ni amélioration. Pour remédier à cette limite, nous proposons une approche hybride novatrice appelée *MAEVa*, qui combine les mesures de similarité fondées sur les corpus grâce à l'appariement des noms et des descriptions des variables. Pour l'appariement des noms, nous proposons une nouvelle approche où nous étendons BERT-base avec une couche d'attention multi-têtes externe (BERTmha). Pour les descriptions, nous enrichissons celles existantes avec l'API GPT-3.5 Turbo et utilisons TF-IDF pour la représentation vectorielle.

3 Approches proposées

MAEVa est un pipeline utilisé pour apparier les variables sources et candidates en procédant comme suit : (1) appariement des noms de variables, (2) appariement des descriptions de variables, (3) combinaison de (1) et (2) dans une fonction linéaire, et (4) une méthode de sélection des résultats pour l'évaluation finale. Avant de décrire chaque étape, nous commençons par présenter le prétraitement des données.

3.1 Prétraitement des données

Le prétraitement des données joue un rôle crucial en transformant le corpus, les noms et descriptions des variables dans une forme adaptée afin de faciliter leur appariement. Le processus utilisé dans cet article comprend plusieurs étapes appliquées dans un ordre précis. La fonction `clean_text()` nettoie les variables sources et candidates ainsi que le corpus en supprimant les chiffres, les parenthèses et leur contenu, et en convertissant les textes en minuscules, garantissant ainsi que l'appariement des variables est insensible à la casse. Ensuite, `remove_stopwords()` élimine les mots vides (*stopwords*), qui n'apportent pas de signification spécifique dans le contexte de l'agroécologie, mettant ainsi l'accent sur les termes clés pour une meilleure mise en correpondance des variables. La fonction `lemmatize()` applique la lemmatisation, qui réduit les mots à leur forme canonique (ex. *rédiges*, *rédigé* et *rédigeras* deviennent *rédiger*), améliorant ainsi la précision des calculs de similarité en comparant les lemmes plutôt que les différentes formes d'un mot. Par ailleurs, `remove_punctuation()` supprime la ponctuation des descriptions et du corpus pour favoriser l'appariement des variables. Enfin, `replace_synonyms()` facilite la mise en correspondance en remplaçant certains mots par leurs synonymes via WordNet, par exemple, *niveau* peut être remplacé par *degré*. Toutes ces fonctions sont appliquées dans leur ordre d'apparition, chacune prenant en entrée la sortie de la fonction précédente, au corpus ainsi qu'aux descriptions des variables sources et candidates. Étant donné que la majorité de nos noms de variables sont souvent des abréviations et ne contiennent pas de ponctuation, nous avons appliqué uniquement les trois premières fonctions pour leur prétraitement. Celles-ci peuvent, dans certains cas, modifier significativement le nom de la variable. Par exemple, la variable 'Number of leaves_n' est transformée en 'number leaf_n' après l'application des

trois fonctions. À l'inverse, certains noms restent inchangés après prétraitement, comme 'abv_om_dm_content_%'. Après cette étape de prétraitement des données, nous détaillons chaque étape du pipeline MAEVa dans les sous-sections suivantes.

3.2 Appariement des noms de variables

L'objectif de l'appariement des noms de variables est de comparer les variables à travers leurs noms (Étape 1). Dans cette étape, nous avons appliqué trois fonctions pour le prétraitement des noms de variables, comme décrit précédemment. Nous avons ensuite utilisé les sorties comme entrées de : (a) BERT (baseline) et (b) notre approche BERTmha pour apparier les noms de variables.

Nous avons utilisé BERT pour encoder chaque nom de variable, puis nous avons appliqué la similarité cosinus afin de mesurer la similarité entre chaque nom de variable source et tous les noms de variables candidates. Ensuite, les résultats ont été classés du candidat le plus similaire au moins similaire pour chaque nom de variable source. Concernant l'approche originale BERTmha que nous proposons, nous avons introduit les noms de variables sources prétraités dans BERT et avons extrait les derniers états cachés contenant l'encodage de chaque nom de variable source. Ensuite, les tenseurs des derniers états cachés du modèle BERT pour les noms de variables sources ont été utilisés comme entrées pour un mécanisme d'attention multi-têtes externe. Ces tenseurs ont été traités à travers des couches linéaires afin de générer les matrices de requête (Q), clé (K) et valeur (V) nécessaires au calcul de l'attention. Dans notre cas, les requêtes (Q) et les clés (K) ont été dérivées des états cachés de BERT pour les noms de variables sources, nous permettant ainsi de mieux comprendre les relations entre les mots composant chaque nom de variable source.

L'attention multi-têtes est définie par :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

où d_k est la dimension des vecteurs de clé. Le mécanisme multi-têtes applique ce calcul sur plusieurs sous-espaces en parallèle :

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \qquad (2)$$

où chaque tête est calculée comme suit :

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \qquad (3)$$

avec W_i^Q, W_i^K, et W_i^V étant les matrices de projection pour la $i^{\text{ème}}$ tête, et W^O la matrice de projection finale. Ce mécanisme permet de capturer les dépendances contextuelles sous plusieurs perspectives, enrichissant ainsi la représentation des noms de variables.

Le mécanisme d'attention multi-têtes produit ensuite un nouvel encodage pour chaque nom de variable source. Ce processus comprend le calcul des scores d'attention, l'application de la fonction *softmax* pour obtenir des probabilités, l'utilisation du *dropout* pour la régularisation, et enfin l'agrégation des vecteurs de contexte pour produire la sortie finale de l'attention. Le même processus de BERTmha a été appliqué aux noms de variables candidates.

Ensuite, nous avons calculé la similarité entre chaque nom de variable source et tous les noms de variables candidates à travers leurs nouvelles embeddings via la similarité cosinus. Les résultats ont ensuite été classés du plus au moins similaire pour chaque nom de variable source. Dans ce travail, pour initialiser les couches d'attention multi-têtes, nous avons adopté une technique de distribution uniforme, dans laquelle les poids ont été initialisés aléatoirement dans l'intervalle $[-1, 1]$.

3.3 Appariement des descriptions de variables

L'objectif de l'appariement des descriptions de variables est de comparer les variables sur la base de leurs descriptions en s'appuyant sur un corpus (Étape 2). Nous avons commencé par générer automatiquement un nouveau corpus contenant une nouvelle description pour chaque variable en utilisant le prompt suivant sur l'API GPT-3.5 Turbo : *You are a farmer. You have knowledge about sugarcane culture and other relevant information. Always answer with the goal of describing [variable's description] and use these descriptions as data for training TF–IDF. Provide the results in a complete paragraph, with a maximum of 500 words.* Nous nous référons au corpus généré sous le nom de *Corpus (GPT-prompt)*. Ce corpus contient toutes les nouvelles descriptions générées pour l'ensemble de nos variables. Ensuite, nous avons prétraité le corpus en utilisant les cinq fonctions décrites en Subsection 3.1, puis nous avons utilisé le *Corpus (GPT-prompt)* prétraité en appliquant une pondération TF-IDF pour construire l'espace vectoriel et extrait la représentation de chaque description de variable.

Comme pour l'appariement des noms de variables, nous avons calculé la similarité cosinus entre chaque description de variable source et toutes les descriptions de variables candidates, puis nous avons classé les résultats du candidat le plus similaire au moins similaire pour chaque description de variable source.

3.4 Méthode de combinaison

À l'étape 3, nous avons utilisé une fonction linéaire qui combine l'appariement des noms et des descriptions de variables afin d'augmenter la précision du processus de mise en correspondance, comme illustré dans l'Équation 4.

$$\text{combinaison} = \alpha \cdot \text{B} + (1 - \alpha) \cdot A \qquad (4)$$

Cela signifie que pour chaque variable source, nous avons combiné son appariement fondé sur le nom (B) et celui basé sur la description (A) avec chaque variable candidate. Les résultats ont ensuite été classés du plus au moins similaire pour chaque variable source. A représente la similarité cosinus entre un nom de variable source et un nom de variable candidate donné, et B la similarité cosinus entre la même description source et la description candidate corres-

pondante. Le paramètre $\alpha \in]0,1[$ est un facteur de pondération ajustant l'importance de A et B (Étape 3). Il influence fortement le résultat : de faibles variations peuvent avoir un impact notable. Une expérimentation rigoureuse est donc nécessaire pour en fixer la valeur optimale.

3.5 Méthode de sélection des résultats

Nous proposons de rassembler et synthétiser les résultats de toutes nos méthodes. Nous proposons que si au moins une méthode identifie correctement la variable candidate réelle pour une variable source donnée, nous la considérons comme un appariement correct.

4 Expérimentations

Dans ce travail, nous avons utilisé des échantillons de variables sélectionnés par des experts en agroécologie afin d'illustrer différentes complexités et de mesurer l'efficacité de MAEVa. Nous disposons de 84 variables sources, de 170 variables candidates, ainsi que du corpus généré par GPT (*Corpus (GPT-prompt)*)[6].

4.1 Configuration expérimentale

Dans cette étude, nous avons utilisé le modèle BERT-base (*"bert-base-uncased"*) de Hugging Face[7] avec 2 couches cachées (2HLs). La méthode de combinaison utilise un seul paramètre, α, dont la valeur a été testée entre 0.01 et 0.99 avec un pas de 0.01. Les meilleurs résultats ont été obtenus avec $\alpha = 0.25$. Pour l'attention multi-têtes, nous avons utilisé une régularisation par dropout (DT = 0.1), une distribution uniforme des poids (UDW) et 256 têtes (Hs).

4.2 Résultats et discussion

4.2.1 Résultats

Pour chaque variable source, nous avons calculé sa similarité avec toutes les variables candidates et classé les résultats du plus similaire au moins similaire. Étant donné que chaque variable source possède exactement une variable candidate correcte, nous utilisons la précision au rang K ($P@K$) comme métrique d'évaluation.

Métrique d'évaluation La précision au rang K (P@K) évalue la capacité du modèle à classer la variable candidate correcte parmi les K premières positions de la liste ordonnée par similarité pour chaque variable source. Formellement, $P@K$ est définie dans l'Équation 5 :

$$P@K = \frac{N_{\text{correct@}K}}{N_{\text{total}}} \tag{5}$$

Où :

- $N_{\text{correct@}K}$: Nombre de variables sources où la variable candidate correcte est classée parmi les K premières.
- N_{total} : Nombre total de variables sources (84).

La Table 3 présente les résultats de l'appariement des noms de variables, tandis que la Table 4 montre les résultats de l'appariement des descriptions de variables. La Table 5 donne les résultats finaux obtenus en combinant les résultats de la mise en lien des noms de variables, des descriptions et de la méthode de combinaison, ce qui signifie que si au moins une méthode identifie correctement la variable candidate réelle pour une variable source donnée, nous la considérons comme un appariement correct (Subsection 3.5).

TABLE 3 – Résultats de l'appariement des noms de variables.

Méthode	P@1	P@3	P@5	P@10
BERTcos	11.90%	28.57%	36.90%	53.57%
BERTmha	**30.90%**	**40.33%**	**51.00%**	**64.69%**

BERTcos : BERT-base (2 HLs) + similarité cosinus, BERTmha : BERT-base (2 HLs) + Attention Multi-Têtes (256 Hs, DT=0.1 et UDW) + similarité cosinus, HLs : Couches cachées, Hs : Têtes, DT : Régularisation par dropout, et UDW : Distribution uniforme des poids.

TABLE 4 – Résultats de l'appariement des descriptions de variables.

Méthode	Corpus	P@1	P@3	P@5	P@10
TF-IDF	×	33.33%	42.86%	51.19%	60.71%
TF-IDF	Corpus (GPT-prompt)	**38.10%**	**50.00%**	**60.71%**	**66.67%**

TABLE 5 – Résultats finaux de l'appariement des variables (MAEVa pipeline).

Noms	Descriptions	Corpus	P@1	P@3	P@5	P@10
BERTcos	TF-IDF	×	59.52%	72.61%	77.38%	84.52%
BERTmha	TF-IDF	Corpus (GPT-prompt)	**66.66%**	**83.33%**	**85.71%**	**86.90%**

BERTcos : BERT-base (2 HLs) + similarité cosinus, BERTmha : BERT-base (2 HLs) + Attention Multi-Têtes (256 Hs, DT=0.1 et UDW) + similarité cosinus, HLs : Couches cachées, Hs : Têtes, DT : Régularisation par dropout, et UDW : Distribution uniforme des poids.

4.2.2 Discussion

La Table 3 montre que notre approche BERTmha a amélioré les résultats de plus de 11 points de $P@1$ à $P@10$. Cela démontre que l'attention multi-têtes ajoutée permet de mieux "comprendre" les noms de variables que l'auto-attention utilisée avec BERT-base. La Table 4 montre une amélioration des résultats de plus de 4 points de $P@1$ à $P@10$, ce qui signifie que les nouvelles descriptions générées par l'API GPT-3.5 Turbo sont efficaces et permettent une meilleure représentation des variables.
Notre pipeline MAEVa, qui combine plusieurs techniques d'appariement, a permis d'améliorer les résultats de plus de 2 points par rapport à la baseline. De plus, les valeurs obtenues avec $P@3$ de MAEVa sont proches des résultats avec $P@10$ de la baseline, comme illustré dans la Table 5. L'amélioration des résultats obtenus avec MAEVa montre que nos méthodes sont complémentaires et que la meilleure approche consiste à utiliser l'ensemble du pipeline pour obtenir des performances optimales. Nos approches présentent certaines limites. Le modèle BERT-base, généraliste, n'est pas adapté au domaine agroécologique ; un af-

6. https://doi.org/10.18167/DVN1/9X3IVR
7. https://huggingface.co/google-bert/bert-base-uncased

finement ou une spécialisation sur ce domaine serait plus pertinent. De plus, les noms de variables, souvent abrégés et peu contextualisés, nuisent à l'explicabilité. À l'inverse, les unités de mesure textuelles (kilogramme) sont plus discriminantes et explicatives. Nous prévoyons de nous concentrer sur ces dernières, en les normalisant, pour l'appariement des noms de variables. Enfin, TF-IDF ne tenant pas compte du contexte, nous envisageons d'adapter des modèles généralistes comme BERT et AgriBERT au domaine agroécologique pour l'appariement des descriptions.

5 Conclusion et perspectives

Dans ces travaux, nous avons présenté une approche hybride appelée MAEVa, qui combine l'appariement des noms et des descriptions de variables pour mettre en lien les variables sources et candidates. Nous avons proposé une nouvelle approche appelée BERTmha pour l'appariement des noms de variables. BERTmha encode les noms de variables et nous extrayons ses couches d'états cachés pour les injecter dans une couche d'attention multi-têtes externe afin de mieux capturer une certaine sémantique des noms de variables. BERTmha a surpassé BERT-base de plus de 11 points de $P@1$ à $P@10$. Pour l'appariement des descriptions de variables, nous avons généré un nouveau corpus de descriptions en utilisant l'API GPT-3.5 Turbo avec un prompt spécifique, puis nous avons appliqué la pondération TF-IDF et la similarité cosinus pour l'appariement. Les résultats montrent que nos nouvelles descriptions comportent des informations plus adaptées que les descriptions originales, et l'appariement des descriptions de variables a été amélioré de plus de 4 points.

Malgré ces résultats encourageants, nous n'avons pas appliqué de méthodes de similarité sémantique ni utilisé d'informations sémantiques dans notre corpus. L'un des principaux défis rencontrés était l'hétérogénéité des variables, en particulier leurs unités de mesure, ce qui peut compliquer le processus de mise en correspondance. Pour y remédier, nous avons initié des travaux sur l'extraction d'informations sémantiques liées aux unités de mesure, y compris les labels et URIs, en utilisant des ontologies telles que QUDT (*Quantities, Units, Dimensions, and Types*), TO (*Plant Trait Ontology*), PO (*Plant Ontology*), UO (*Units de Measurement Ontology*) et OM (*Ontology de Units de Measure*). Notre objectif est de normaliser ces unités et d'utiliser les informations sémantiques extraites comme descripteurs pour faciliter le processus de mise en lien. Étant donné que MAEVa est une approche hybride et générique qui peut être facilement adaptée à différentes mesures de similarité textuelle, diverses techniques de construction de corpus et différentes méthodes de combinaison, nous prévoyons d'intégrer des informations sémantiques en affinant un modèle de langage pré-entraîné tel que RoBERTa [17], BERT [2] ou des grands modèles de langue tels que GPT-4 [19] ou LLaMA [28]. Pour la construction de corpus, il est envisageable d'explorer d'autres techniques, telles que l'extraction d'informations sémantiques à partir d'ontologies et de thésaurus. Nous prévoyons également d'utiliser des techniques de *Retrieval-Augmented Generation* (RAG) [15] afin de rechercher les documents ou textes pertinents pour une variable donnée et d'utiliser ce contexte pour améliorer la génération de GPT. Enfin, nous envisageons d'appliquer d'autres modèles génératifs, tels que GPT-4, LLaMa et Mistral, pour générer et contextualiser nos variables [10].

Remerciements. Ce travail est financé par l'Agence Nationale de la Recherche (ANR) au titre de France 2030 portant la référence ANR-16-CONV-0004 (#DigitAg) et par le programme de recherche et d'innovation Horizon Europe dans le cadre de la convention 101081973 - IntercropValuES. Ce travail a bénéficié du soutien du Conseil régional de La Réunion, du ministère français de l'Agriculture et de l'Alimentation, de l'Union européenne (programme Feader, subvention AG/974/DAAF/2016-00096 et programme Feder, subvention GURTDI 20151501-0000735).

Références

[1] Sandrine Auzoux, Mathias Christina, François-Régis Goebel, Alizé Mansuy, and Daniel Marion. A dictionary of variables to harmonize data from agro-ecological experiments on sugarcane. In *Proceedings of the 3rd ISSCT Agricultural Engineering, Agronomy and Extension Workshop*, page 22, Saint-Gilles, La Réunion, 2018. ISSCT.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[3] Léa Dieudonat, Kelvin Han, Phyllicia Leavitt, and Esteban Marquer. Exploring the combination of contextual word embeddings and knowledge graph embeddings. *arXiv preprint arXiv :2004.08371*, 2020.

[4] Aissa Fellah, Ahmed Zahaf, and Atilla Elçi. Semantic similarity measure using a combination of word2vec and wordnet models. *Indonesian Journal of Electrical Engineering and Informatics*, 12(2) :455–464, 2024.

[5] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb : The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 758–764, Atlanta, Georgia, 2013. Association for Computational Linguistics.

[6] Wael H. Gomaa and Aly A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13) :13–18, 2013.

[7] Ramadan T. Hassan and Nawzat S. Ahmed. Evaluating the efficacy of semantic similarity methods for comparison of academic thesis and dissertation texts. *Science Journal of University of Zakho*, 11(3) :396–402, 2023.

[8] Aryan Jadon and Shashank Kumar. Leveraging generative ai models for synthetic data generation in healthcare : Balancing research and privacy. In *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pages 1–4. IEEE, 2023.

[9] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406) :414–420, 1989.

[10] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv :2310.06825*, 2023.

[11] Nanjiang Jiang and Marie-Catherine de Marneffe. Do you know that florence is packed with visitors ? evaluating state-of-the-art models of speaker commitment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4208–4213, Florence, Italy, 2019. Association for Computational Linguistics.

[12] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1) :11–21, 1972.

[13] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv :1607.01759*, 2016.

[14] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv :1405.4053*, 2014.

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv :2005.11401*, 2020.

[16] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized bert pretraining approach. *CoRR*, abs/1907.11692, 2019.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[19] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*, 2023.

[20] Avinash Patil, Kihwan Han, and Aryan Jadon. A comparative study of text embedding models for semantic text similarity in bug reports. *arXiv preprint arXiv :2308.09193*, 2023.

[21] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[22] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[23] Hieu Pham, Minh-Thang Luong, and Christopher D. Manning. Learning distributed representations for multilingual text sequences. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 88–94, Denver, Colorado, USA, 2015. Association for Computational Linguistics.

[24] Nils Reimers and Iryna Gurevych. Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics.

[25] Alexandre Salle, Aline Villavicencio, and Marco Idiart. Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, pages 419–424, Berlin, Germany, 2016. Association for Computational Linguistics.

[26] Robyn Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey, 2012. European Language Resources Association (ELRA).

[27] Martina Toshevska, Frosina Stojanovska, and Jovan Kalajdjieski. The ability of word embeddings to capture word similarities. *International Journal on Natural Language Computing*, 9(3) :25–42, 2020.

[28] H. Touvron and et al. Llama : Open and efficient foundation language models. In *arXiv preprint arXiv :2302.13971*, 2023.

[29] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080, New York, NY, USA, 2016. PMLR.

[30] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy, 2019. Association for Computational Linguistics.

[31] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA, 1994. Association for Computational Linguistics.