Natural Language Processing Journal 13 (2025) 100180

Contents lists available at ScienceDirect

Natural Language Processing Journal

journal homepage: www.elsevier.com/locate/nlp

MAEVa: A hybrid approach for matching agroecological experiment variables

Oussama Mechhour [a,b,c,*], Sandrine Auzoux [a,b], Clément Jonquet [d,e], Mathieu Roche [a,c]

[a] CIRAD, F-34398 Montpellier, France
[b] AIDA, Univ. of Montpellier, CIRAD, Reunion Island, France
[c] TETIS, Univ. of Montpellier, AgroParisTech, CIRAD, CNRS, INRAE, Montpellier, France
[d] MISTEA, Univ. of Montpellier, INRAE, Institut Agro, Montpellier, France
[e] LIRMM, Univ. of Montpellier, CNRS, Montpellier, France

HIGHLIGHTS

- Extending PLMs with multi-head attention improves name matching vs. original PLMs
- Various corpus construction techniques have been applied
- Analysis of relevance and impact of corpora constructed for matching descriptions
- An efficient combination method merges name and description matching
- Combination method outperforms name and description matching separately.

ARTICLE INFO

Keywords:
Observable properties
Corpus-based similarity
Hybrid-based similarity
Pretrained Language Models (PLMs)
Large Language Models (LLMs)

ABSTRACT

Source variables or observable properties used to describe agroecological experiments are heterogeneous, nonstandardized, and multilingual, which makes 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 (1) string-based similarity, (2) corpus-based similarity, (3) knowledge-based similarity, and (4) hybrid-based similarity, which combine two or more of these measures. This work presents a hybrid approach called Matching Agroecological Experiment Variables (MAEVa), which combines well-known techniques (PLMs, multi-head attention, TF–IDF) tailored to the challenges of aligning source and candidate variables in agroecology. MAEVa integrates the following components: (1) Our key innovation, which consists of extending pretrained language models (PLMs) (i.e., BERT, SBERT, SimCSE) with an external multi-head attention layer for matching variable names; (2) An analysis of the relevance and impact of various data collection techniques (snippet extraction, scientific articles) and prompt-based data augmentation on TF–IDF for matching variable descriptions; (3) A linear combination of components (1) and (2); and (4) A voting-based method for selecting the final matching results. Experimental results demonstrate that extending PLMs with an external multi-head attention layer improves the matching of variable names. Furthermore, TF–IDF benefits consistently from the presence of an enriched corpus, regardless of the specific enrichment technique employed.

1. Introduction

Agroecological systems are, by nature, complex: the basic elements of the system are the characteristics and activities of individual processes, these processes are heterogeneous, their characteristics can change over time, the dynamics are non-linear or even chaotic and feedback loops modify and disrupt these processes, thus increasing non-linearity creating a non-derivable system (Caquet et al., 2019). Agroecological experiments generate databases that can be multiscale (plant, cropping system, farm, landscape, territory), multispecies (crops, associated companion plants, weeds, forage plants) and multidisciplinary

* Corresponding author.
E-mail address: oussama.mechhour@cirad.fr (O. Mechhour).

https://doi.org/10.1016/j.nlp.2025.100180
Received 7 February 2025; Received in revised form 5 August 2025; Accepted 12 August 2025

(agronomy, weed science, entomology, economic and social sciences, environment). The source variables or observable properties[1] used to describe agroecological experiments are often named and described using homonyms, synonyms, acronyms, multiple languages, and sometimes non-standard terms that can be difficult to interpret and explain. They are also measured with heterogeneous units, even for the same variable, and are heterogeneous from linguistic, structural, semantic, syntactic, and taxonomic perspectives. This complexity makes them difficult to use in cropping system modeling and multi-criteria evaluations of agroecological system performance. To address these challenges, data annotation through a common controlled vocabulary or ontology serves as a solution (Arnaud et al., 2020). To harmonize and standardize these non standardized, heterogeneous source variables, the French Agricultural Research Centre for International Development (CIRAD) developed the Agroecological Global Information System (AEGIS) (Auzoux et al., 2018). AEGIS integrates a harmonized data acquisition and processing chain that utilizes a set of candidate variables, combining semantic terms from reference ontologies (Plant Ontology, Crop Ontology, Environment Ontology, and Agronomy Ontology) and expert agroecological knowledge. The approach adopted in AEGIS allows researchers to freely use their own source variable names, descriptions, and units of measurement while providing a list of candidate variables to harmonize and standardize the source variables. In this work, we utilized heterogeneous and nonstandardized source variables collected from sugarcane experiments associated with service plants, originating from CanecoH,[2] Ecocanne,[3] and AgriecoH projects conducted by eRcane[4] in collaboration with CIRAD[5] in La Réunion.

Table 1 presents samples of source variables, and Table 2 presents samples of candidate variables. These tables highlight a few examples of the complexity of these variables. For example, in Table 1, the first variable name is multilingual; i.e., it consists of "yield", which is in English, and "CAS", which is an acronym in French for "cane à sucre" (sugarcane). The second variable name is in French, and there are different ways of writing units of measurement. For example, the following two examples use different notations for division: "$t.ha^{-1}$" and "kg/m^2". In the introduction of Table 2, an additional characteristic, "method of calculation", is included, which is absent in Table 1. Furthermore, Table 2 uses "scale" instead of "units of measurement" as seen in Table 1. For example, lines 2 and 3 in Table 1 refer to the same variable as line 3 in Table 2, despite differing descriptions.

As illustrated in Tables 1 and 2, variables are mostly expressed with text descriptions, making text similarity measures a suitable solution for matching source and candidate variables. Text similarity measurement is a fundamental concept in information theory and is used to evaluate the proportion of shared content between two texts (Lin, 1998). A high similarity score indicates strong closeness between the texts. Text similarity involves assessing the closeness between two texts, considering both lexical and semantic similarities. This means that even if two texts do not share the same words, the texts may still convey similar meanings. This concept has become central in several areas of natural language processing (NLP), including information retrieval (Li et al., 2014), automatic question answering (Jiang and de Marneffe, 2019), machine translation (Wang et al., 2019), and document matching (Pham et al., 2015). Common text similarity measures (Gomaa and Fahmy, 2013) can be grouped into four categories: (1) string-based (lexical-based), (2) corpus-based, (3) knowledge-based, and (4) hybrid-based measures.

String-based similarity measures assess the resemblance between two texts solely on the basis of literal comparison of words or characters, without considering synonyms, context, or semantic relationships between words. These methods do not rely on an external semantic resource or corpus for calculating similarity and can be further divided into two subcategories:

- **Character-based similarity measures** evaluate similarity by considering exact matches, transpositions, and modifications (insertions, deletions, substitutions) that are needed to transform one string into another. Examples include:
  - **Levenshtein distance** (Levenshtein, 1966), which is a well-known example that quantifies these transformations;
  - **Jaro similarity** (Jaro, 1989), which measures matching characters and transpositions;
  - **Jaro–Winkler similarity** (Winkler, 1990), which emphasizes matches at the beginning of strings;
  - **Hamming distance** (Hamming, 1950), which counts the number of differing positions between two strings of equal length.
- **Word-based similarity measures** calculate the similarity between two sets of words by comparing the shared and distinct elements. Common methods include:
  - **Sorensen–Dice index** (Sorensen and Dice, 1948), which calculates the similarity as twice the number of common elements divided by the total number of elements in both sets. This method works well for small sets but may introduce bias when the sets differ significantly in size;
  - **Overlap similarity** (Manning et al., 2008), which measures common elements relative to the smaller set, is effective for asymmetric comparisons;
  - **Tversky index** (Tversky, 1977), which generalizes the previous measures by introducing weighting parameters for shared and distinct elements, offering greater flexibility.

Corpus-based similarity measures use information derived from a corpus to calculate similarity, which can involve textual features or co-occurrence probabilities. These measures are divided into (1) frequency-based methods and (2) shallow window-based methods.

- **Frequency-based methods** convert texts into numerical vectors, where each dimension reflects a specific feature of the text, such as word frequency. These approaches focus mainly on the quantitative aspects of words in a text without considering word order, context, or semantic relations between them.
  The most common methods in this category are as follows:
  - **BoW (Bag of Words)** (Harris, 1954): This method represents each document as a word frequency vector without considering word order.
  - **TF–IDF (Term Frequency–Inverse Document Frequency)** (Jones, 1972): A method that weights words on the basis of their frequency in a document relative to their frequency in the entire corpus, reducing the importance of common words.
  - **BM25 (Best Match 25)** (Robertson et al., 1994): An information retrieval model that improves TF–IDF by adjusting term relevance on the basis of frequency and document length, making search results more accurate.
- **Shallow window-based methods** transform texts into dense vectors, dividing them into word-based and sentence-based representations.

[1] An observable property is the description of something observed or derived.

[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
Examples of source variables. Each name is a combination of the variable name and its unit of measurement, with the unit appearing after the final underscore.

Names	Descriptions	Units of Measurement
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 of the trial	%
DM_end_weed_kg/m^2	Weed aerial dry mass at the end of the trial	kg/m^2
…	…	…

- **Word-based representations**: Words are transformed into dense vectors, which can be static or dynamic. Static vectors, such as those produced by Word2Vec (Mikolov et al., 2013), FastText (Joulin et al., 2016), and GloVe (Pennington et al., 2014), are context independent. In contrast, dynamic vectors, such as those produced by Pretrained Language Models (PLMs) like BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019), consider the context of words within a given window, although they may sometimes fail to capture word semantics in contexts requiring a broader contextual window.
- **Sentence-based representations**: These methods represent entire sentences as vectors, such as in SBERT (Reimers and Gurevych, 2019), USE (Cer et al., 2018) and InferSent (Conneau et al., 2018).

Knowledge-based similarity measures rely on structured networks of semantically connected concepts to evaluate word similarity and can be extended to sentence-level analysis. These networks are often domain-specific (e.g., biomedicine or legal studies) or general networks such as WordNet (Miller et al., 1995), which is widely used in measuring knowledge-based similarity (Gomaa and Fahmy, 2013).

Hybrid-based similarity measures combine two or more text similarity measures, such as string-based, corpus-based, and knowledge-based measures, to leverage their strengths and improve similarity measurement accuracy.

The originality of our work lies in the combination of well-known techniques (PLMs, multi-head attention, TF–IDF), tailored to the specific challenges of aligning source and candidate variables in the agroecological sector. The main contributions of our proposed hybrid approach called Matching Agroecological Experiment Variables (MAEVa) to align source and candidate variables are as follows: (1) Our key innovation, which consists of extending PLMs (i.e., BERT, SBERT, SimCSE) with an external multi-head attention layer for matching variable names; (2) An analysis of the relevance and impact of various data collection techniques (snippet extraction, scientific articles) and prompt-based data augmentation on TF–IDF for matching variable descriptions; (3) A linear combination of components (1) and (2); (4) A voting-based method for selecting the final matching results; and (5) The generated corpora, source code, and a guide to reproduce all our results are available at: https://github.com/OussamaMECHHOUR/MAEVa-v1.

The remainder of this paper is structured as follows: Section 2 provides a review of related works on text similarity measures, highlighting their limitations and what we propose in this paper. In Section 3, we introduce our hybrid MAEVa pipeline and details its components. Section 4 presents the experiments and analysis related to each component of MAEVa (i.e., matching of variable names, the matching of descriptions, the combination method, and the voting-based final evaluation). Section 5 discusses the limitations of MAEVa. Finally, Section 6 concludes the paper with closing remarks and an outline of future work.

2. Related work

In this section, we present related work for each category mentioned in Section 1, excluding knowledge-based similarity measures that were not applied in this study.

2.1. String-based similarity measures

String-based similarity measures have been widely applied in various works. For example, (Akhmedovich and Sattarova, 2024) employed the Jaccard Index (Jaccard, 1901) to recommend books that align with the intellectual abilities of schoolchildren. For this purpose, a specialized corpus was constructed using high-level literature textbooks (i.e., suitable books for students), which were then compared with literary works (i.e., books to recommend). The books with the highest similarity scores were suggested for reading. This approach was successfully applied via literature textbooks for students in grades 5 to 11, along with literary works in the Uzbek language. Additionally, (Susanto et al., 2023) employed the Levenshtein distance (Levenshtein, 1966), Hamming distance (Hamming, 1950), Jaro–Winkler similarity (Winkler, 1990), and Sorensen–Dice index (Sorensen and Dice, 1948) to compare each word extracted from noisy images via optical characteristic recognition (OCR) with words in a database of correct terms. This approach aimed at correcting errors in the extracted text. The experiments demonstrated that the Sorensen–Dice index achieved the highest performance, with an F1 score, precision, and recall of 0.88, although it required more processing time than the other methods. Furthermore, (Po, 2020) proposed a system to help users retrieve relevant conference papers based on title searches. The system tokenizes both the user's query title and the titles in a database containing more than 300 previous paper titles while removing insignificant words. It then applies the Levenshtein distance to measure the similarity between the tokenized user title and each paper title in the database. The system ranks the results from most to least similar, providing additional relevant information such as similar titles, author names, conference names, and publication dates.

2.2. Corpus-based similarity measures

Corpus-based similarity measures have been widely applied in various works. For example, (Patil et al., 2023) compared TF–IDF (Baseline) (Jones, 1972), FastText (Joulin et al., 2016), doc2vec (Le and Mikolov, 2014), BERT (Devlin et al., 2018), and ADA (Jadon and Kumar, 2023) in the task of recommending parent or unique bug reports for a given child bug report via the Software Defects Datasets (Jadon et al., 2022; Jadon and Jadon, 2023). A child bug report is recognized as a duplicate of a previously submitted bug report (parent report), whereas unique reports are those that have not been linked as child or parent to any other report. The Software Defects Data included approximately 480,000 bug reports from EclipsePlatform, MozillaCore, Firefox, JDT, and Thunderbird. Patil et al. (2023) demonstrated that BERT generally outperformed the other models in terms of recall, followed by ADA, doc2vec, FastText, and TF–IDF. In another study, (Hassan and Ahmed, 2023) compared TF–IDF, doc2vec, BERT, and sentence-BERT (SBERT) (Reimers and Gurevych, 2019) with cosine similarity to measure the similarity between academic theses and dissertations of graduate students. Two datasets were used: (1) the Duhok Polytechnic University collection, which contains 27 original English theses and dissertations, and (2) the ProQuest collection from "proquest.com", which includes 100 original English theses and dissertations. After preprocessing the documents (text extraction, character removal, stopword elimination, stemming, and lemmatization), the

Table 2
Examples of candidate variables. Each name is a combination of the variable name and its scale, with the scale appearing after the final underscore.

Names	Descriptions (Traits)	Scales	Methods of calculation
leaf_plant_dm_kg	Measurement of foliar dry biomass at the individual level (plant scale)	kg	Common measurement method
abv_om_dm_content_%	Organic matter concentration of the WAB	%	Organic matter concentration defined based on dry matter concentration and mineral concentration
plant_ground_cover_%	Measurement of plant (or species) recovery by ceptometry	%	Common measurement method
fruit_plant_fm_kg	Measurement of fresh fruit biomass at the individual level (plant scale)	kg	Common measurement method
…	…	…	…

four methods were applied separately to compute similarity within each collection. The results showed that TF–IDF outperformed the other methods in terms of accuracy, precision, recall, F1 score, and processing time, followed by doc2vec, BERT, and SBERT. Furthermore, (Romualdo et al., 2021) compared several methods for measuring e-commerce product title similarity in Brazilian Portuguese, using the Americanas corpus,[6] which contains approximately 7.490 million products. The authors explored both specific-domain (Word2Vec, FastText, and GloVe) and general-domain (Word2Vec, FastText, GloVe, and BERT models) word embeddings with cosine similarity. For specific-domain methods, Word2Vec (CBOW and skip-gram), FastText (CBOW and skip-gram), and GloVe were trained on the Americanas corpus. In the case of general-domain methods, the authors used pretrained Word2Vec (CBOW and skip-gram), FastText (CBOW and skip-gram), and GloVe models by NILC,[7] and two BERT models, multilingual BERT[8] and BERTimbau (large and base) (Souza et al., 2020), which are tailored for Brazilian Portuguese. The authors applied each method separately to measure e-commerce product title similarity in Brazilian Portuguese via a test corpus constructed with four different techniques (Romualdo et al., 2021). The results demonstrated that multilingual BERT combined with cosine similarity yielded the best results for calculating product title similarity.

2.3. Hybrid-based similarity measures

Two or more text similarity measures, such as string-based, corpus-based, and knowledge-based methods, are combined to harness the advantages of each and improve the accuracy of similarity measurements. Several studies have explored these approaches, as cited below:

- Fellah et al. (2024) proposed a hybrid approach called the aggregated semantic similarity measure (ASSM) for comparing two words, where the semantic similarity between two words (or word pairs) is the maximum similarity derived from two methods: (1) the cosine similarity between word embeddings obtained from Google's pretrained Word2Vec, which was trained on the Google News dataset containing approximately 100 billion words, and (2) a linear function combining the cosine similarity from Word2Vec with the WordNet Wu & Palmer similarity between the two words (Wu and Palmer, 1994). This approach was evaluated on several benchmark datasets (RG-65, WS353-all, WS353-sim, MC-30, and AG-203) via Spearman's and Pearson's correlation coefficients to compare the similarity values computed by using Word2Vec and ASSM with the corresponding human judgment scores. The experimental results demonstrated that combining WordNet and Word2Vec for measuring semantic similarity between words was more effective than using Word2Vec alone, outperforming some other approaches applied for the same purpose.
- Dieudonat et al. (2020) compared ELMo (Peters et al., 2018), knowledge base embeddings (KBs) via ComplEx (Trouillon et al., 2016), and the concatenation of these methods in entity typing and relation typing tasks, both of which are multiclass classification tasks. The authors utilized the official ELMo pretrained model,[9] and for KB embeddings, they selected PyTorch-BigGraph[10] implementation of ComplEx (Lerer et al., 2019), trained on the Freebase 15 K (FB15K) subset. The comparison was conducted on two datasets: the Freebase-NewYorkTimes dataset ("FB-NYT") (Riedel et al., 2010) and Freebase 15 K (FB15K) (Bordes et al., 2013). For the experimental tasks, they used precision@n, mean average precision@k (MAP@k), and mean reciprocal rank (MRR) as evaluation metrics. The results showed that (1) the concatenation of KB embeddings and ELMo yielded the best performance in the entity typing task, followed by KB embeddings alone and ELMo alone, and (2) KB embeddings outperformed other methods in the relation typing task, followed by ELMo alone and the concatenation of KB embeddings and ELMo.
- Finally, Toshevska et al. (2020) analyzed the effectiveness of cosine similarity, which was applied to several word embedding methods, in capturing word similarity across pairs of words from five different human-generated datasets. The authors evaluated Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Joulin et al., 2016), LexVec (Salle et al., 2016), and ConceptNetNumberbatch (a hybrid method that combines GloVe and Word2Vec embeddings with additional structured knowledge from semantic networks such as ConceptNet Speer and Havasi, 2012 and PPDB Ganitkevitch et al., 2013) to measure the similarity between word pairs on the following datasets: WordSim353 (Finkelstein et al., 2001), SimLex999 (Hill et al., 2014), SimVerb3500 (Gerz et al., 2016), RG65 (Rubenstein and Goodenough, 1965), and RW2034 (Luong et al., 2013). The analysis was conducted in two parts: (1) an average similarity analysis, where the authors computed the average similarity between word pairs in each dataset and compared it to the average human ratings, and (2) a correlation analysis, where Spearman, Pearson, and Kendall's tau correlation coefficients were calculated

[6] http://www.americanas.com.br
[7] http://www.nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc
[8] https://github.com/google-research/bert
[9] https://github.com/allenai/allennlp-models
[10] https://github.com/facebookresearch/PyTorch-BigGraph

between the ground truth similarities and the cosine similarities of the word embeddings. The results indicated that GloVe and FastText outperformed the other word embeddings on four out of the five datasets in terms of average similarity. However, ConceptNetNumberbatch achieved the highest results in terms of correlation analysis, despite not having the highest average similarity values, except for the RW2034 dataset.

2.4. Comparative overview

We have presented various existing methods and works for measuring text similarity at both the lexical and semantic levels. Each method has its own advantages and disadvantages. *String-based similarity measures* are relatively simple, but they have significant limitations: they do not account for synonyms, context, or the semantics of words. For example, two synonymous words are not considered to be similar by these methods. For *corpus-based similarity measures*, we have shown that some of them do not correctly account for context. For example, *frequency-based or statistical approaches* treat two words with similar meanings independently and do not take order into account. Owing to their purely statistical nature, they can present limitations such as the curse of dimensionality, where vectors become too large to manage efficiently, or out-of-vocabulary (OOV) issues when words in new texts do not appear in the corpus. *Shallow window-based similarity measures* such as Word2Vec, FastText, and GloVe, which use a limited context window and assign a single dense vector to each word, are also limited: regardless of the word's context, its vector representation remains the same. In contrast, methods such as BERT, RoBERTa, and XLNet consider context by generating multiple vectors for the same word on the basis of its immediate textual environment, improving semantic context representation. However, these methods are less effective when a broad context is required to fully understand the meaning of the text or word. To leverage the strengths of each method while overcoming their limitations, hybrid-based similarity measures have been introduced. The hybrid approaches cited in this section typically combine established similarity measures without introducing significant methodological innovations. In contrast, this paper proposes a novel hybrid approach, MAEVa, designed to match source and candidate variables based on two key innovations: (a) extending PLMs (i.e., BERT, SBERT, SimCSE) with an external multi-head attention layer to enhance the matching of variable names, and (b) analyzing the relevance and impact of various data collection techniques (snippet extraction, scientific articles) and prompt-based data augmentation on TF–IDF for variable description matching. The combination method has been used in previous studies and has demonstrated its effectiveness (Fellah et al., 2024). For this reason, we adopted it for its efficiency and proven performance. However, in our approach, the combination is not simply the use of multiple similarity measures. It involves integrating complex and enriched data together with these methods to build a coherent and complete pipeline.

3. Proposed approach

First, we provide a summary of our proposed pipeline, MAEVa, in the following Section 3.1. Each step of MAEVa's pipeline is detailed in Sections 3.2, 3.3, 3.4, and 3.5.

3.1. MAEVa pipeline

The matching of agroecological experiment variables pipeline (MAEVa) outlines the methodology used to match source and candidate variables. In this work, we utilized samples of variables selected by agroecological experts that illustrate different complexities to measure the efficiency of MAEVa. For each of the 84 source variables, we calculate its similarity with all 170 candidate variables based on each step of the MAEVa's pipeline (Fig. 1), then we rank the results from highest to lowest similarity candidates for each source variable. For evaluation purposes, we rely on a manually curated reference file created by domain experts, which indicates the correct match for each source variable. Auzoux et al. (2023) and Mechhour et al. (2025) detailed the used approach for constructing the evaluation dataset based on a rigorous protocol.

As shown in Fig. 1, data preprocessing is common to the first two steps. We detail this process below, and in the following Subsections, we outline each of our four steps.

Data preprocessing Data preprocessing plays a crucial role. It aims to transform the corpus, variable names and descriptions into a suitable form to facilitate their matching. The data preprocessing used in this paper includes several steps, which are applied in the order of their appearance:

1. `clean_text()`: this function is used to clean the source and candidate variables as well as the corpus. It removes numbers, parentheses, and their contents and converts the names, descriptions and the corpus to lowercase, ensuring that the variable matching process is case-insensitive.
2. `remove_stopwords()`: stopwords are commonly used words in languages that do not carry specific meanings in the context of agroecology. This function removes these functional words, such as prepositions and conjunctions. By eliminating stopwords, the key terms that are more meaningful for matching variables have been focused upon.
3. `lemmatize()`: lemmatization is a linguistic technique that involves reducing words to their canonical form, or lemma. It transforms plural nouns into singular forms and verbs into their infinitive form. For example, it can reduce the words *running*, *runs*, and *ran* to their base form *run*. It improves the accuracy of similarity calculations by comparing lemmas rather than different word forms.
4. `remove_punctuation()`: this function removes punctuation from the variable descriptions and the corpus. By eliminating special characteristics such as periods, commas, and quotation marks, we avoid unwanted interference during variable matching.
5. `replace_synonyms()`: to facilitate matching variables, this technique allows for the replacement of certain words with their synonyms via WordNet. For instance, the word *level* can be replaced with *degree*.

All the functions above were applied sequentially, with each function taking the output of the previous function to the corpus and the descriptions of both the source and the candidate variables. However, only the first three functions were used to preprocess the variable names. After explaining the data preprocessing, we detail each step of the MAEVa's pipeline (Fig. 1) in the subsections below.

3.2. Matching variable names

Variable names are often acronyms and lack sufficient context, making them challenging to interpret using PLMs. Our intuition is that PLMs are capable of effectively embedding such variable names. For this reason, our innovation consists of extending existing PLMs with an external multi-head attention layer applied to their frozen embeddings. We compare the performance of this extended architecture with that of the original PLMs to evaluate their effectiveness in representing variable names. Additionally, string-based similarity measures are effective for tasks that do not require contextual understanding, such as spelling correction and duplicate detection. Given that our variable names are often acronyms and lack clear semantics and context, we included string-based similarity measures for comparison purposes.

Prior to similarity computation, we applied three preprocessing functions to the variable names, as discussed earlier. The processed names were then used as inputs for both string-based and corpus-based similarity measures, as described below.

3.2.1. String-based similarity measures

We applied several widely-used string-based similarity measures from the literature, including the Jaro–Winkler similarity (Winkler, 1990), the Sorensen–Dice index (Sorensen and Dice, 1948), and the Overlap Coefficient (Manning et al., 2008), to match variable names for comparison purposes. Each of these measures computes the similarity between a source variable name and all candidate variable names. The candidate names are then ranked in descending order of similarity for each source variable name.

3.2.2. Corpus-based similarity measures

We focused on PLMs within the corpus-based similarity methods. We selected three of the most widely used models in the literature:

- **BERT** (Devlin et al., 2018) is a bidirectional transformer-based model trained through self-supervised learning on two main objectives: Masked Language Modeling (MLM), where 15% of the input tokens are randomly masked and the model tries to predict them, and Next Sentence Prediction (NSP), which trains the model to determine whether a given sentence logically follows another.
- **SBERT** (Reimers and Gurevych, 2019) is a modification of BERT that enables efficient sentence-level embeddings by introducing a siamese network structure. It is fine-tuned using a contrastive objective so that semantically similar sentence pairs are mapped to nearby points in the embedding space, making it more effective for semantic similarity tasks.
- **SimCSE** (Gao et al., 2021) (Simple Contrastive Learning of Sentence Embeddings) improves sentence representations by using contrastive learning on natural language inference data. It builds embeddings that distinguish between similar and dissimilar sentences through dropout-based augmentation or supervised pair labeling, depending on the version (unsupervised or supervised).

For each of these models, we extracted the frozen embeddings of each variable name. These embeddings were then passed through an external multi-head attention layer to produce enhanced representations. The goal was to assess whether this attention-based extension could generate embeddings that more effectively represent our variable names. Cosine similarity was then computed between each source and candidate variable name based on these new embeddings, and the candidates were ranked from highest to lowest similarity for each source variable name.

3.3. Matching variable descriptions

The objective is to analyze the relevance and impact of various data collection techniques (snippet extraction, scientific articles) and prompt-based data augmentation on TF–IDF (Jones, 1972) for matching variable descriptions.

Data augmentation is a technique used in low-resource settings to generate diverse data from the original data without additional data collection. This task is often used to reduce overfitting. It has been applied in previous works, ranging from traditional methods (before the appearance of LLMs) to advanced ones (after the appearance of LLMs). The former includes, in a non-exhaustive list: (1) thesaurus-based substitution (Zhang et al., 2015; Mueller and Thyagarajan, 2016), (2) MLM-based generation (Garg and Ramakrishnan, 2020), and (3) back-translation (Xie et al., 2020); for more details, see Chaudhary (2020), Feng et al. (2021). After the emergence of LLMs, they have been applied in several works for data augmentation. As mentioned in this survey (Chai et al., 2025), LLM-based data augmentation can be divided into (1) prompt-based, (2) retrieval-based, and (3) hybrid methods. Our work is cited under the prompt-based methods, where prompts are used to guide LLMs to generate more controlled data based on a given input (our descriptions). After explaining data augmentation, we will deal in the next Subsection with the details of how these corpora are obtained.

3.3.1. Corpora construction

- **Articles:** Fifteen representative English-language articles from the field of agroecology were selected by three experts, which we refer to as the *Corpus (15 articles)*.
- **Snippets:** Refer to small portions of text or summaries extracted from a larger document or collection of documents. They are specifically designed to provide a quick overview or insight into the content, enabling users to understand the context without reading the entire document. Given that the descriptions are often brief, using snippets provides more detailed information than the variable descriptions do. In this work, we utilized the Custom Search JSON API provided by Google and the Bing Search v7 API provided by Microsoft. For each API, we input our source and candidate variable descriptions and receive a corpus containing snippets that provide concise summaries for each variable description. We refer to the corpus obtained via the Google API as the *Corpus (Google)* and the corpus obtained via the Microsoft API as the *Corpus (Microsoft)*.
- **GPT-3.5 Turbo:** We used the GPT-3.5 Turbo API with six prompts, which are described below. Each prompt is used to generate a new contextualized description of each source and candidate variable on the basis of its original description. As output, each prompt generates a corpus that contains all the newly generated descriptions of both types of variables. We refer to the corpus obtained via the first prompt as *Corpus (GPT-prompt 1)*, the second as *Corpus (GPT-prompt 2)*, and so on, with *Corpus (GPT-prompt 6)* for the sixth prompt. Collectively, all these corpora are referred to as *Corpus (GPT)*. The six prompts used in this work are listed below:

  1. 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.
  2. You are a farmer with knowledge of sugarcane cultivation and other relevant information. Surround [variable's description] with useful information to assist TF–IDF in better vectorizing it. Provide the results in a complete paragraph, with a maximum of 500 words.
  3. You are a farmer. You have knowledge about sugarcane cultivation and other relevant information. Please provide with all the information you have about [variable's description]. Keep in mind that I am a data scientist.
  4. You are an agronomist working in sugarcane and other crops. Can you describe the variable [variable's description] in a paragraph with a of maximum 500 words.
  5. You are an agronomist working on sugarcane and other crops. Can you contextualize the variable [variable description] with relevant close terms in a paragraph with a of maximum 500 words for a good vectorization of the variable.
  6. You are an agronomist working in sugarcane and other crops. Can you give 5 examples of 100-word snippets of text mentioning the variable [variable's description].

  The six prompts used to generate *Corpus (GPT)* were carefully designed to simulate realistic descriptions of variables from complementary expert viewpoints. Specifically, Prompts 1–3 were crafted to emulate the language and perspective of a farmer, focusing on practical and observational insights, while Prompts 4–6 simulate the more technical and structured descriptions expected from agronomists. This dual perspective is intentional: it enables the TF–IDF (Jones, 1972) model to capture both informal, real-world terminology and scientifically structured expressions.

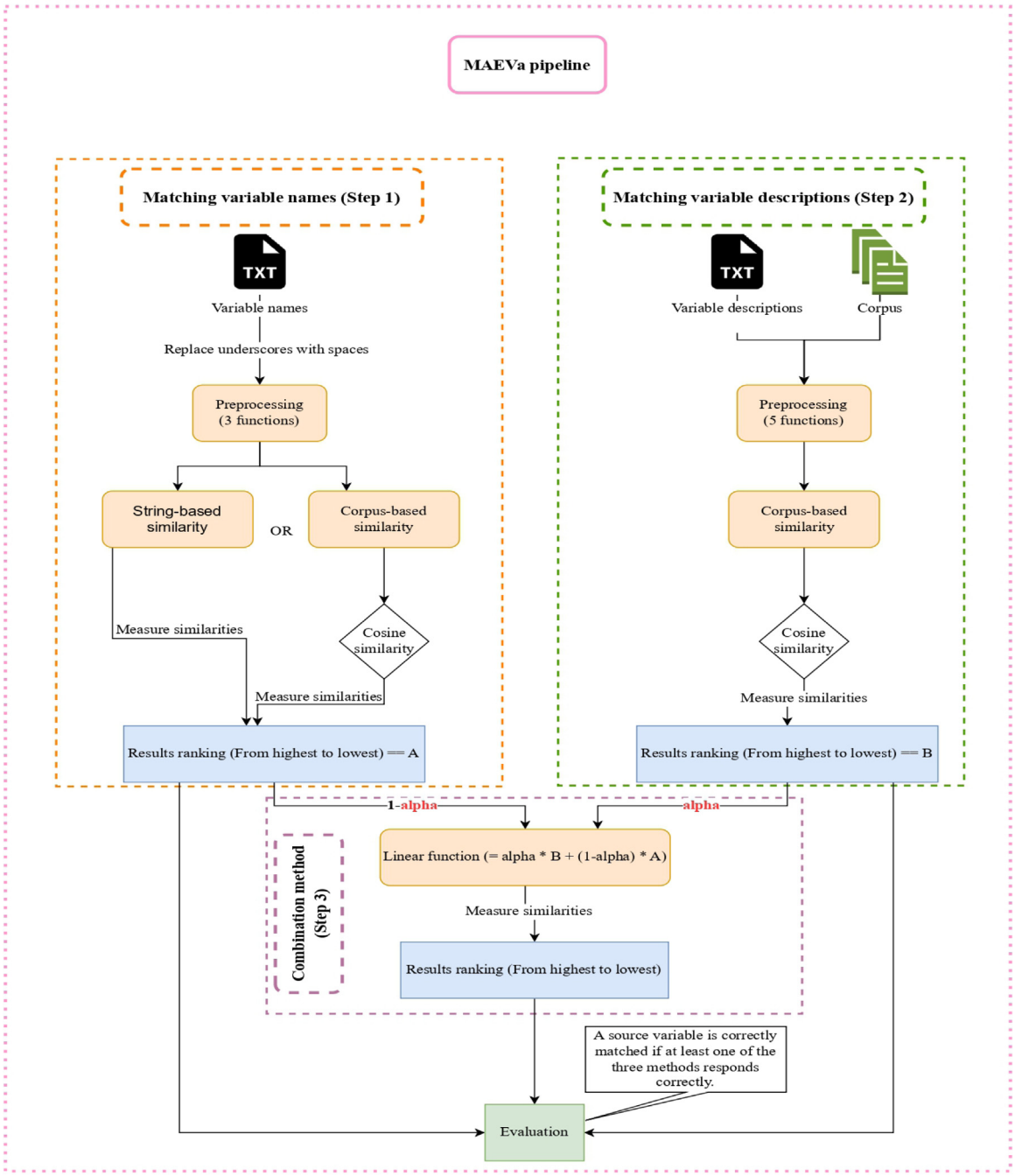

Fig. 1. MAEVa Pipeline.

Prompts 1 and 2 explicitly request that the variable description be enriched with surrounding lexical context to improve TF–IDF vectorization. This aligns with prompt engineering principles for text augmentation used in information retrieval tasks (Liu et al., 2023; Sun et al., 2023). Prompt 3 asks for exhaustive knowledge relevant to the variable, assuming a data scientist audience, encouraging the generation of rich but coherent content. Prompts 4 and 5 simulate scientific communication, while Prompt 6 produces 5 × 100-word snippet-style descriptions inspired by search engine summaries, which are typically used in retrieval-based NLP applications.

We selected GPT-3.5 Turbo as the generation API due to its well-documented ability to generate controllable, high-quality text. The API's balance between fluency, factuality, and style made it particularly suitable for our goal of generating multiple realistic, domain-relevant corpora. Moreover, the generation process was bounded using fixed word constraints-500 words for prompts 1–5 and 100-word snippets for prompt 6-based on the belief that generating large volumes of text increases the risk of hallucinations, which can introduce noisy or irrelevant tokens. Therefore, we fixed the number of words in each prompt output to minimize this risk and ensure the quality and relevance of the generated content.

Table 3 presents the constructed corpora along with the number of words in each corpus.

Table 3
Corpora constructed and the number of words in each corpus.

Corpus name	Word count
Corpus (15 articles)	1,037,564
Corpus (GPT-prompt 1)	96,755
Corpus (GPT-prompt 2)	102,296
Corpus (GPT-prompt 3)	83,709
Corpus (GPT-prompt 4)	110,933
Corpus (GPT-prompt 5)	101,953
Corpus (GPT-prompt 6)	58,109
Corpus (Microsoft)	110,958
Corpus (Google)	26,260

3.3.2. Corpus-based similarity measures

To analyze the impact of different corpus construction techniques, as described in Section 3.3.1, on corpus-based methods, we selected TF–IDF due to its simplicity, despite its context-independence. Our objective is not to implement the most advanced state-of-the-art models for matching source and candidate descriptions, but rather to examine, in a straightforward manner, how various corpus construction strategies influence performance. To this end, we used TF–IDF to vectorize the source and candidate descriptions: (a) using each corpus individually, and (b) using different combinations of these corpora. Subsequently, cosine similarity was computed between each source description and all candidate descriptions, and the results were ranked in descending order of similarity.

3.4. Combination method

In Step 3 of our pipeline (Fig. 1), we combine the similarity scores obtained from name-based and description-based matching using a linear function to improve overall matching precision.

Let $S = \{\text{Var}_s^1, \ldots, \text{Var}_s^N\}$ be the set of N source variables and $C = \{\text{Var}_c^1, \ldots, \text{Var}_c^M\}$ the set of M candidate variables. For each pair $(\text{Var}_s^i, \text{Var}_c^j)$, we compute the final matching score as:

$$\text{comb}(\text{Var}_s^i, \text{Var}_c^j) = \alpha \cdot \text{sim}_{\text{desc}}(\text{Var}_s^i, \text{Var}_c^j) + (1-\alpha) \cdot \text{sim}_{\text{name}}(\text{Var}_s^i, \text{Var}_c^j) \quad (1)$$

where:

- $\text{sim}_{\text{name}}(\text{Var}_s^i, \text{Var}_c^j)$ is the similarity score between variable names.
- $\text{sim}_{\text{desc}}(\text{Var}_s^i, \text{Var}_c^j)$ is the similarity score between variable descriptions.
- $\alpha \in]0, 1[$ is a tunable weighting parameter that controls the contribution of each component.

This linear combination reflects the intuition that variable names and descriptions provide complementary signals. When α approaches 1, the method prioritizes description-based similarity; conversely, when α approaches 0, it favors name-based similarity. The parameter α is sensitive to changes, and even slight variations can significantly impact the final ranking of candidates. Therefore, we empirically evaluate several values of α to identify the most effective balance between the two components (Section 4.1).

In accordance with Fig. 1, the ranked results from name similarity (Step 1) are denoted as A, and those from description similarity (Step 2) are denoted as B. These ranked scores are linearly combined using Eq. (1) to produce the final ranking used in Step 3.

3.5. Evaluation method

To assess the overall effectiveness of our approach, we propose a voting-based evaluation strategy that leverages the complementary strengths of all our matching methods. Instead of relying solely on a single technique, we consider a match to be correct if it is successfully retrieved by at least one of the following three approaches: (i) matching variable names, (ii) matching variable descriptions, or (iii) their linear combination. This strategy is designed to increase the likelihood of retrieving the correct candidate variable by leveraging the complementary signals from these three methods. Our objective is not to identify the best individual method, but rather to evaluate the collective performance of the ensemble, aiming to maximize matching coverage.

Let Var_s^i denote the ith source variable, and let c^i be the unique correct candidate match for Var_s^i as specified in the ground truth. Let $R_k^{(\text{name})}(i)$, $R_k^{(\text{desc})}(i)$, and $R_k^{(\text{comb})}(i)$ denote the sets of the top-k candidate variables returned for Var_s^i by the name-based, description-based, and combination methods, respectively. The prediction for Var_s^i is considered correct under the voting strategy if:

$$c^i \in R_k^{(\text{name})}(i) \cup R_k^{(\text{desc})}(i) \cup R_k^{(\text{comb})}(i) \quad (2)$$

This evaluation strategy (Eq. (2)) assumes the availability of a ground truth file specifying a unique correct match c^i for each ith source variable Var_s^i. Without this information, the voting-based strategy cannot be applied.

4. Experiments and discussion

After detailing our proposed MAEVa pipeline (Section 3), the methods used, and how they were applied in this study, we now present the results.

4.1. MAEVa parameters

We begin by describing the experimental setup. For methods requiring parameter tuning, we explain how the optimal parameters were determined. While we conducted extensive experiments, we do not display all intermediate results. Instead, for each method with adjustable parameters, we report only the best-performing configuration. Below, we summarize each method, the optimal parameters identified, and the checkpoint used for each PLM.

- **BERT-base (case-insensitive):** We used the checkpoint ‘‘`bert-base-uncased`’’ from Hugging Face.[11] We experimented with different numbers of hidden layers (HLs), and the best results were achieved using 2 HLs.
- **SBERT:** We used the checkpoint ‘‘`all-MiniLM-L6-v2`’’ from Hugging Face.[12]
- **SimCSE:** We used the checkpoint ‘‘`sup-simcse-bert-base-uncased`’’ from Hugging Face.[13]
- **Multi-head Attention:** We initialized the weights using a uniform distribution in the range $[-1, 1]$ and explored different numbers of attention heads to identify the most effective configuration. The best results were achieved using dropout regularization with a rate of 0.1 (DT=0.1), uniform weight initialization (UDW), and 256 attention heads (Hs). Furthermore, our code is fully reproducible.
- **Combination Method:** This method relies on a single weighting parameter, α, which controls the balance between name-based and description-based similarity. Given the sensitivity of this parameter, we selected the combination of BERT-base and TF–IDF applied on *Corpus (GPT-prompt 1)* as a reference configuration to determine the optimal value of α. We conducted an evaluation by varying α from 0.01 to 0.99 with a step size of 0.01.
  Fig. 2 shows the precision curves for each individual metric (P@1, P@3, P@5, P@10), annotated with the best-performing α for

[11] https://huggingface.co/bert-base-uncased
[12] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[13] https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased

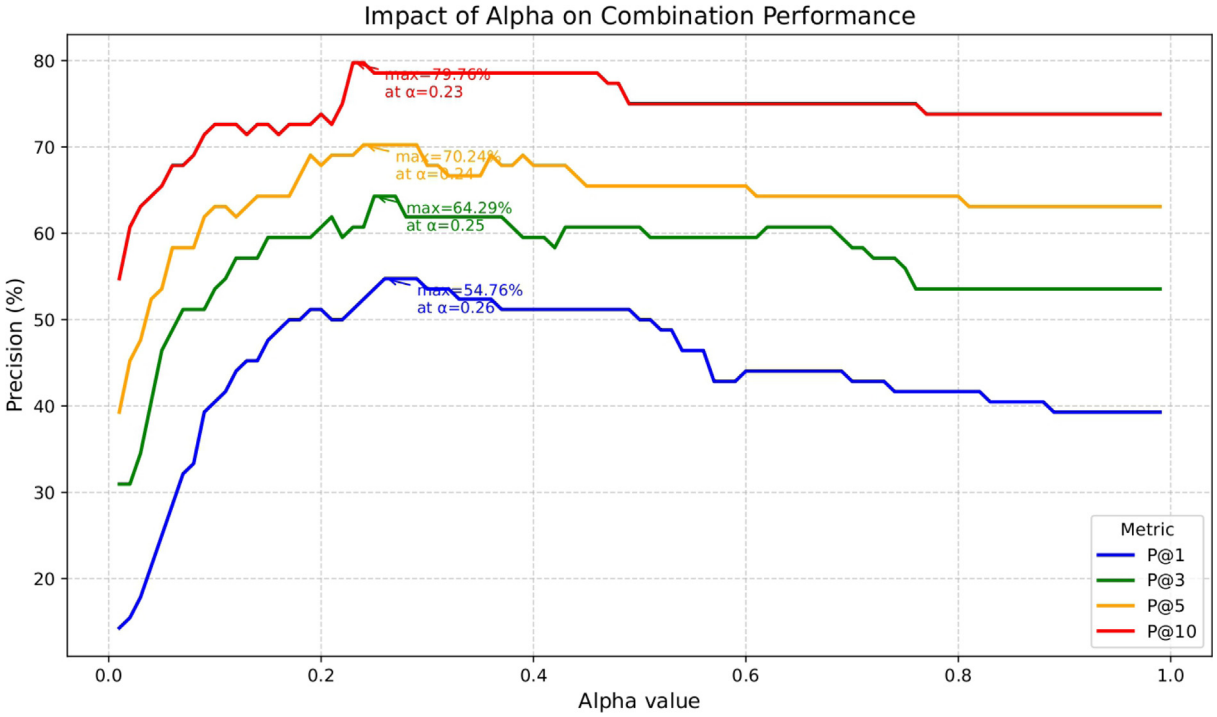

Fig. 2. Impact of α on Combination Performance.

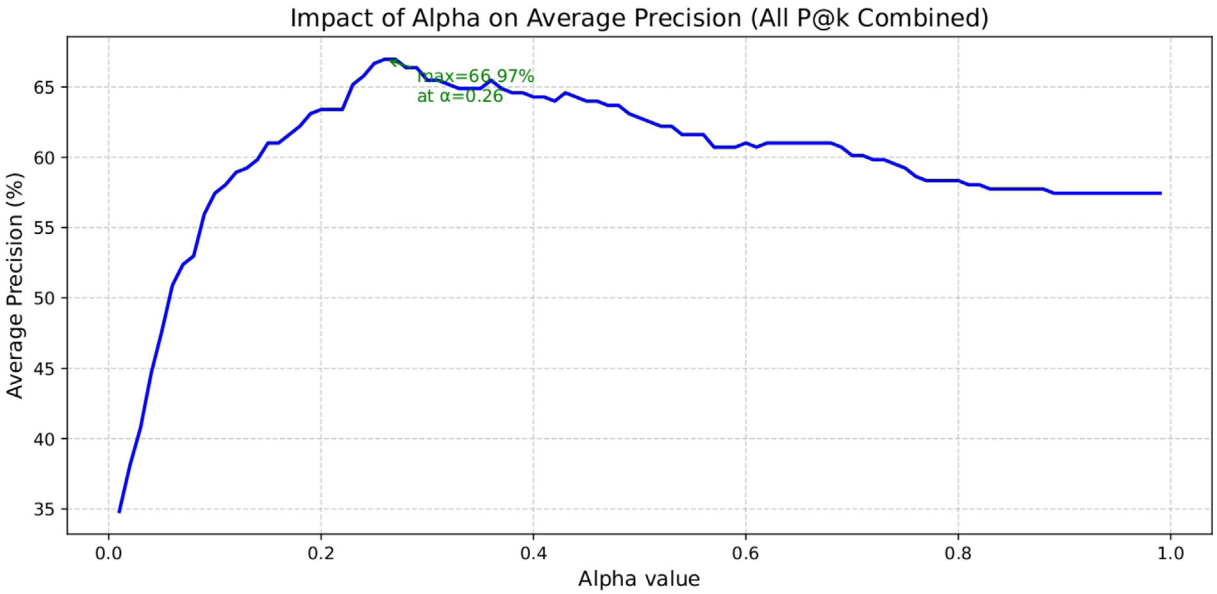

Fig. 3. Impact of α on Average Precision (All P@k Combined).

each. The optimal values lie within a narrow window around $\alpha = 0.25$, with P@3 and P@5 achieving their maximum exactly at this point.

Fig. 3 presents the average precision computed across all P@k levels. Although the global maximum is reached at $\alpha = 0.26$ (66.97%), the score at $\alpha = 0.25$ (66.67%) is nearly identical, differing by only 0.3 percentage points. Furthermore, the curve is relatively flat in this region, indicating stability and robustness. Based on these observations, we consistently adopt $\alpha = 0.25$ as the default setting in all combination-based methods.

Given that each source variable has exactly one correct candidate variable, we use precision at position K ($P@K$) as our evaluation metric. This metric measures the proportion of source variables for which the correct matching candidate appears within the top K selections, divided by the total number of source variables. We use this evaluation metric to assess all our results.

4.2. Matching variable names results

As shown in Table 4, the results demonstrate that our core innovation, extending the selected PLMs with an external multi-head attention

Table 4
Matching variable names results: Similarity was calculated between each of the 84 source variable names and each of the 170 candidate variable names. The results show the proportion of source variables for which the correct matching candidate variable appears within the top $K \in [1, 10]$ selected candidates, divided by the total number of source variables.

Method	P@1	P@3	P@5	P@10
Jaro–Winkler Similarity	8.33%	17.86%	22.62%	30.95%
Overlap Coefficient	8.33%	21.43%	32.14%	42.86%
Sorensen–Dice Index	2.38%	3.57%	3.57%	5.95%
BERT-base (2 HLs) + cosine similarity	11.90%	28.57%	36.90%	53.57%
BERT-base (2 HLs) + Multi-Head attention (256 Hs, DT=0.1 and UDW) + cosine similarity	**20.24%**	**35.71%**	**41.67%**	**54.76%**
SBERT + cosine similarity	9.52%	20.24%	23.81%	27.38%
SBERT + Multi-Head attention (256 Hs, DT=0.1 and UDW) + cosine similarity	11.90%	20.24%	27.38%	34.52%
SimCSE + cosine similarity	9.52%	10.71%	14.29%	21.43%
SimCSE + Multi-Head attention (256 Hs, DT=0.1 and UDW) + cosine similarity	8.33%	10.71%	16.67%	28.57%

HLs: Hidden Layers, Hs: Heads, DT: Dropout regularization, and UDW: Uniform Distribution of Weights

layer, leads to improved representations of variable names compared to the original models. This enhancement consistently increases precision across most evaluated values of k. The best overall performance was achieved by our extended BERT-base model, with improvements ranging from +1 to +8 percentage points over the original BERT-base. String-based similarity measures are effective for tasks that do not require contextual understanding, such as spelling correction and duplicate detection. Given that our variable names are often acronyms and lack clear semantics, we included string-based similarity measures for comparison purposes. The results show that string-based methods can be competitive with PLMs such as SBERT and SimCSE. Although these methods were initially included for comparison purposes, we observed that their performance warrants further investigation. In particular, we aim to explore whether lexical overlap between source and candidate variable names may bias the results in favor of string-based similarity measures. To investigate this, we conducted a systematic analysis of lexical overlaps between each source variable name and all candidate names.

For each source variable name, we computed the number of overlapping tokens with each candidate name. The correct candidate match was identified based on the human annotation file. We then observed whether any incorrect candidate names shared an equal or greater number of overlapping tokens than the true match. A source variable was defined as *biased* if the number of incorrect candidates with higher or equal token overlap exceeded that of the correct match. Analysis shows that only 9 out of 84 source names (10.7%) were identified as lexically biased. In these cases, incorrect candidate names had equal or greater token-level overlap with the source than the ground truth match. Conversely, 75 source variables (89.3%) were not affected by such bias.

4.3. Matching variable description results

In Table 5, the results show that the performance is similar across all corpora generated via the GPT-3.5 Turbo API, as well as Microsoft snippets. The corpus composed of scientific articles selected by domain experts (i.e., *Corpus (15 articles)*) achieves the best results at P@3 and P@5 and the google snippets (i.e., *Corpus (Google)*) achieved the best P@10. However, when these corpora are combined, the final performance remains the same.

To further assess the relevance and potential noise within each corpus, we conducted a detailed domain keyword density analysis across all corpora used in our TF–IDF + Cosine similarity experiments. We defined a list of domain-specific keywords.[14] and expressions frequently used in agroecology. Keywords were extracted from our variable descriptions and supplemented by terms provided by an agroecology expert. The analysis revealed significant differences in keyword density across corpora, as shown in Fig. 4

Among the enriched corpora, *Corpus (GPT-prompt 4)*, *Corpus (GPT-prompt 6)*, and hybrid corpora such as *Corpus (GPT-prompt 6) + Corpus (Google)* and *Corpus (GPT-prompt 4) + Corpus (Google)* also demonstrate relatively high keyword densities. This highlights that certain prompts and retrieval-based methods are able to generate domain-relevant content that complements the original descriptions. In contrast, corpora such as *Corpus (GPT-prompt 1)* and *Corpus (GPT-prompt 2)*, despite providing contextualization, tend to contain more generic agricultural language with less dense domain vocabulary. This analysis provides lexical evidence that the enriched corpora are not only diverse in linguistic structure but also differ in domain specificity. By integrating multiple such corpora, we aimed to strike a balance between thematic relevance and textual variability to support more robust vectorization. Although there are differences in domain keyword density among the corpora, their performance results remain largely similar.

4.4. Combination method results

Table 6 presents the results of the combination method. The best results for P@1, P@3, and P@5 were obtained by combining BERT-base with TF–IDF applied to the fusion of all corpora. The highest performance for P@10 was achieved by combining BERT-base with TF–IDF applied to *Corpus (Google)*. Our objective was not to evaluate all possible combinations of methods and corpora, but rather to focus on a subset of experiments and use the final line of Table 6 for statistical significance analysis. This was done to assess whether the improvements observed with the combination method are indeed meaningful and not due to chance. We selected the last row for analysis because it represents the combination of the best-performing results for name matching-which includes our innovation of extending BERT-base with a multi-head attention layer, as highlighted in Table 4. Moreover, as shown in Table 5, all augmented corpora produced closely similar results in the TF–IDF experiments. Therefore, we selected the first prompt (i.e., *Corpus (GPT-prompt 1)*) as the representative corpus for this evaluation.

To assess whether the performance improvements of the *Combination method* are statistically significant, we conducted Wilcoxon signed-rank tests (Wilcoxon, 1945), Friedman tests (Friedman, 1940), and Nemenyi post-hoc analyses (Nemenyi, 1963) on all experiments presented in Table 6, excluding string-based methods, which are included solely for comparison purposes. The full results of the Wilcoxon signed-rank tests and the Friedman tests followed by Nemenyi post-hoc analyses are provided in Appendix, in Tables A.11 and A.12, respectively. In this subsection, we focus on the statistical analyses conducted on the final row of Table 6, which corresponds to the best-performing configuration in the final MAEVa results (Table 9). These tests compare the Combination method against two baselines: Name-based and Description-based similarity. We focus on these comparisons to validate whether the observed gains arise from a meaningful fusion of signals, rather than chance. We compared the Combination method separately against the Name-based and Description-based baselines across four evaluation

[14] https://github.com/OussamaMECHHOUR/MAEVa-v1.5/blob/main/datasets/keywords/default_keywords.txt

Table 5
Matching variable descriptions results: Similarity was calculated between each of the 84 source variable descriptions and each of the 170 candidate variable descriptions. The results show the proportion of source variables for which the correct matching candidate variable appears within the top $K \in [1, 10]$ selected candidates, divided by the total number of source variables.

Method	Corpus	P@1	P@3	P@5	P@10
TF-IDF + Cosine similarity	✗	33.33%	42.86%	51.19%	60.71%
	Corpus (15 articles)	35.71%	**55.95%**	**63.10%**	65.48%
	Corpus (15 articles) + variable names and descriptions	36.90%	47.62%	59.52%	66.67%
	Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	36.90%	47.62%	59.52%	66.67%
	Corpus (Google) + Corpus (Microsoft) + Corpus (GPT) + Corpus (15 articles) + variable names and descriptions	36.90%	47.62%	59.52%	66.67%
	Corpus (Google)	35.71%	48.81%	59.52%	**67.86%**
	Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 1)	**36.90%**	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 2)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 3)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 4)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 5)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 6)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 1) + Corpus (Google)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 1) + Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 2) + Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 3) + Corpus (Google)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 3) + Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 4) + Corpus (Google)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 4) + Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 5) + Corpus (Google)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 5) + Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 6) + Corpus (Google)	36.90%	47.62%	59.52%	66.67%
	Corpus (GPT-prompt 6) + Corpus (Microsoft)	36.90%	47.62%	59.52%	66.67%

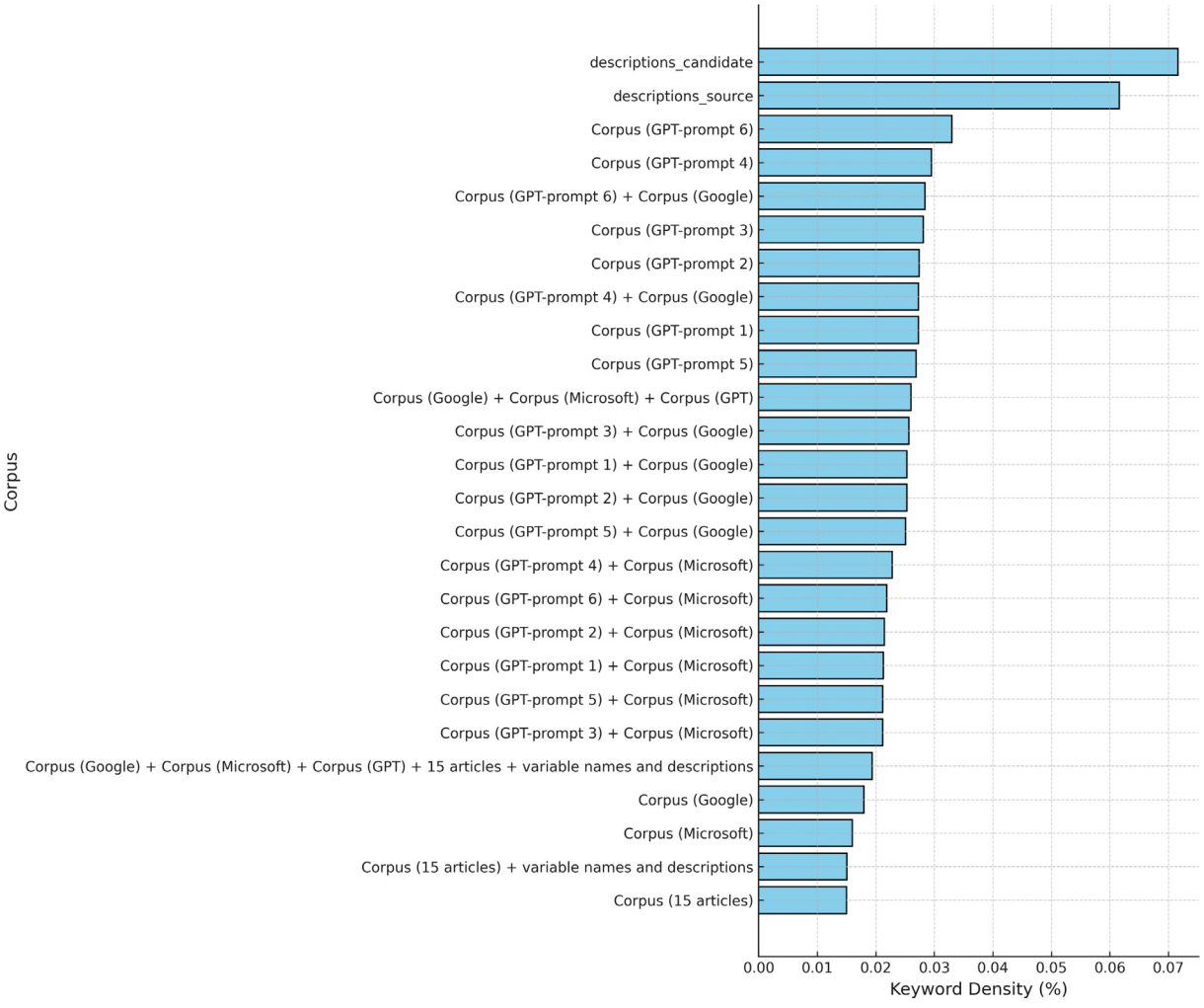

Fig. 4. Keyword density across all corpora used in matching variable descriptions.

Table 6
Results of the combination method. Similarity was calculated between each of the 84 source variable and each of the 170 candidate variable. The results show the proportion of source variables for which the correct matching candidate variable appears within the top $K \in [1, 10]$ selected candidates, divided by the total number of source variables.

Corpus	P@1	P@3	P@5	P@10	Matching names	Matching descriptions
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	30.95%	45.24%	55.95%	67.86%	Jaro–Winkler Similarity	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	38.10%	57.14%	64.29%	71.43%	Overlap Coefficient	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	50.00%	60.71%	65.48%	76.19%	Sorensen–Dice Index	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT) + Corpus (15 articles) + variable names and descriptions	**54.76%**	**64.29%**	**70.24%**	78.57%	BERT-base (2 HL) + Cos	TF–IDF + Cos
Corpus (Google)	54.76%	63.10%	70.24%	**79.76%**	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 2)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 3)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 4)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 5)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 6)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1) + Corpus (Google)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1) + Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 2) + Corpus (Google)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 2) + Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 3) + Corpus (Google)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 3) + Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 4) + Corpus (Google)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 4) + Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 5) + Corpus (Google)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 5) + Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 6) + Corpus (Google)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 6) + Corpus (Microsoft)	53.57%	64.29%	70.24%	78.57%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	32.14%	45.24%	51.19%	63.1%	SBERT + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	27.38%	40.48%	52.38%	59.52%	SBERT + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	38.1%	42.86%	46.43%	52.38%	SimCSE + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	38.1%	45.24%	48.81%	54.76%	SimCSE + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	46.43%	59.52%	64.29%	77.38%	BERT-base (2 HLs) + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF + Cos

HLs: Hidden Layers, Hs: Heads, DT: Dropout regularization, UDW: Uniform Distribution of Weights and Cos: Cosine similarity

Table 7
Wilcoxon signed-rank test p-values comparing the Combination method against the Name-based and Description-based baselines, based on the last line of Table 6. Significant differences ($p < 0.05$) are highlighted in bold.

P-value (Combination vs Name)				P-value (Combination vs Description)			
P@1	P@3	P@5	P@10	P@1	P@3	P@5	P@10
0.000059	**0.000407**	**0.000941**	**0.000074**	0.059346	**0.025347**	0.317310	**0.029049**

Table 8
Nemenyi post-hoc test p-values comparing the Combination method against the Name-based and Description-based baselines, based on the last line of Table 6. Significant differences ($p < 0.05$) are highlighted in bold.

P-value (Combination vs Name)				P-value (Combination vs Description)			
P@1	P@3	P@5	P@10	P@1	P@3	P@5	P@10
0.0293	0.0538	0.0713	0.0713	0.6239	0.4788	0.8886	0.5506

levels (P@1, P@3, P@5, P@10). Table 7 reports the p-values. Notably, the Combination method significantly outperforms the Name-based method at all P@k levels, and the Description-based method at P@3 and P@10.

We also applied the Friedman test followed by the Nemenyi post-hoc test to assess ranking differences across all methods. Table 8 shows that the only statistically significant difference is between the Combination method and the Name-based method at P@1 ($p = 0.029$).

4.5. MAEVa pipeline results

In this subsection, we apply the voting-based method described in Section 3.5, which considers a match correct if at least one of the three methods (i.e., name matching, description matching, or combination method) correctly identifies the true candidate variable for a given source variable. Table 9 presents the results obtained using this evaluation strategy.

The best result at P@1 was achieved by combining BERT-base with TF–IDF applied to the fusion of all corpora and *Corpus (Google)*. For all other P@k values, the highest performance was obtained by combining our key innovation-BERT-base extended with a multi-head attention layer-with TF–IDF applied to *Corpus (GPT-prompt 1)*. In general, the final MAEVa pipeline improved performance by more than 26 percentage points with respect to name matching, more than 16 percentage points over description matching, and more than 2 percentage points over the combination method alone across all P@k metrics. All MAEVa's codes are available on GitHub and fully reproducible. Each code runs entirely on a CPU and produces complete results in less than 10 s, demonstrating the approach's efficiency and suitability for real-world use cases.

Table 9
MAEVa pipeline results. Similarity was calculated between each of the 84 source variable and each of the 170 candidate variable. The results show the proportion of source variables for which the correct matching candidate variable appears within the top $K \in [1, 10]$ selected candidates, divided by the total number of source variables.

Corpus	P@1	P@3	P@5	P@10	Matching names	Matching descriptions
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	41.66%	65.47%	71.42%	80.95%	Jaro–Winkler Similarity	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	51.19%	69.04%	73.80%	80.95%	Overlap Coefficient	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	58.33%	70.23%	76.19%	80.95%	Sorensen–Dice Index	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (Google) + Corpus (Microsoft) + Corpus (GPT) + Corpus (15 articles) + variable names and descriptions	**60.71%**	71.43%	77.38%	84.52%	BERT-base (2 HL) + Cos	TF–IDF + Cos
Corpus (Google)	60.71%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 2)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 3)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 4)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 5)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 6)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1) + Corpus (Google)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1) + Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 2) + Corpus (Google)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 2) + Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 3) + Corpus (Google)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 3) + Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 4) + Corpus (Google)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 4) + Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 5) + Corpus (Google)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 5) + Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 6) + Corpus (Google)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 6) + Corpus (Microsoft)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	44.05%	55.95%	67.86%	79.76%	SBERT + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	45.24%	59.52%	75.00%	83.33%	SBERT + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	44.05%	53.57%	66.67%	76.19%	SimCSE + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	47.62%	55.95%	66.67%	78.57%	SimCSE + + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	59.52%	71.43%	77.38%	84.52%	BERT-base (2 HLs) + Cos	TF–IDF + Cos
Corpus (GPT-prompt 1)	57.14%	**72.62%**	**79.76%**	**84.52%**	BERT-base (2 HLs) + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF-IDF + Cos

HLs: Hidden Layers, Hs: Heads, DT: Dropout regularization, UDW: Uniform Distribution of Weights and Cos: Cosine similarity

4.6. Error analysis

Although we obtained encouraging results, our methods still fail to correctly match a significant number of source variables. To analyze the reasons behind these errors, we focused on the $P@10$ metric. This means that we examined all source variables for which the correct candidate was not retrieved within the top 10 proposed candidates.

Across all our matching methods (i.e., matching names, matching descriptions, combination method) we observed that the mismatched candidate variables tended to be either semantically distant or semantically close to the source variable. In both cases, the candidate variables could share the same unit of measurement as the source or not. These combinations highlight two main sources of error: semantic mismatch and unit mismatch.

- **Row 1** in Table 10 illustrates a case where the proposed candidate is semantically distant and does not share the same unit of measurement as the source variable.
- **Row 2** shows a candidate that is semantically distant but shares the same unit of measurement as the source.
- **Row 3** presents a semantically close candidate[15] which does not share the same unit.
- **Row 4** provides an example where the candidate is semantically close and shares the same unit of measurement as the source.
- **Rows 1–2 and 3–4** highlight that within the top 10 results, both types of errors-semantic and unit-based-can occur simultaneously for a single source variable.

These observations underline the importance of considering units of measurement alongside textual descriptions when matching variable descriptions. One possible improvement for future work is to apply a unit-of-measurement-based filtering mechanism, comparing source and candidate variables only when their units are dimensionally compatible. This approach could simplify the matching task and increase the likelihood of correct matches. However, it also introduces a limitation: Two variables may share the same unit but still be semantically unrelated.

[15] These variables are semantically close because cane yield is a specific part of the total fresh biomass. Both are measured in fresh weight and refer to sugarcane productivity.

Table 10
Examples illustrating various types of errors made by our methods (i.e., matching names, matching descriptions, combination method).

Source Variables		Candidate Variables	
Description	Unit	Description	Unit
Weed coverage index over the trial	%	Height of the apex of the sugar stem sample	cm
Weed coverage index over the trial	%	Soluble sugar concentration of the leaves	%
Cane yield (in fresh machinable stem)	$t.ha^{-1}$	Total fresh biomass	kg
Cane yield (in fresh machinable stem)	$t.ha^{-1}$	Aerial biomass of straws at harvest per unit area measured on the plot	$t.ha^{-1}$
…	…	…	…

To overcome this, we plan to explore a hybrid strategy that integrates unit-based filtering with semantic similarity between descriptions.

5. Discussion

Despite the promising results obtained by MAEVa, several limitations remain that open up avenues for future research. First, the combination strategy employed is a simple linear fusion of name- and description-based similarity scores. While effective, it does not leverage more sophisticated fusion techniques such as trainable neural similarity functions or learning-to-rank models, which could dynamically assign weights to each component. Second, MAEVa does not yet incorporate external knowledge sources such as ontologies or thesauri. This limits its ability to capture deeper semantic relationships, particularly in cases where lexical similarity is low but conceptual similarity is high. To address this, we have initiated work on normalizing units of measurement using ontologies such as QUDT (Quantities, Units, Dimensions, and Types), TO (Plant Trait Ontology), PO (Plant Ontology), UO (Units of measurement Ontology), and OM (Ontology of units of Measure). These resources provide structured semantic information (e.g., labels, URIs) that can be integrated as features to improve matching precision. In parallel, we aim to retrieve and exploit relevant ontologies and thesauri from AgroPortal (Jonquet et al., 2018). These semantic resources offer valuable domain knowledge that can support the adaptation of PLMs, such as RoBERTa (Liu et al., 2019), BERT (Devlin et al., 2018), and AgriBERT (Rezayi et al., 2022), as well as large language models (LLMs), including GPT-4 (OpenAI, 2023), LLaMA (Touvron et al., 2023), and Mistral (Jiang et al., 2023). Furthermore, these resources can be incorporated into retrieval-augmented generation (RAG) pipelines (Lewis et al., 2020), allowing the system to retrieve domain-specific definitions, concept hierarchies, or standard descriptions for each variable prior to generation or comparison. This retrieved context could serve as a basis for generating enriched, semantically grounded, and domain-adapted representations, thereby enhancing matching precision.

Multilinguality is another important limitation of the current approach. Many agroecological datasets include variable names and descriptions in multiple languages, which may reduce MAEVa's generalizability. In future work, we aim to incorporate multilingual modeling through token- or sentence-level language detection and apply cross-lingual embeddings from models such as XLM-R (Conneau et al., 2020) or LaBSE (Feng et al., 2022) to align multilingual variables without requiring explicit translation.

Another important limitation concerns the size and characteristics of our evaluation dataset. The ground truth contains only 84 correct matched pairs between source and candidate variables. This is a typical constraint in the agroecological domain, where expert-annotated data is scarce and costly to produce. Based on the ground truth file, we have 84 correct matches and 6,972 non-matching pairs, resulting in a highly imbalanced dataset. This imbalance makes it difficult to conduct fair comparisons with recent supervised or semi-supervised entity matching systems such as DeepMatcher (Xie et al., 2024) or Ditto Li et al. (2020), which generally require large and balanced datasets and are prone to overfitting in low-resource settings. To address these limitations, we plan to incorporate additional ground truth matchings from ongoing projects such as IntercropVALUES,[16] and to explore data augmentation and rebalancing strategies in future work.

6. Conclusion and future work

In this work, we introduced a hybrid approach called *Matching Agroecological Experiment Variables* (MAEVa), designed to align source and candidate variables based on three components: their names, their descriptions, and the combination of both.

For matching variable names, our key innovation involved extending some PLMs (i.e., BERT-base, SBERT, SimCSE) by integrating an external multi-head attention layer on top of their frozen embeddings. This architectural extension improved the precision of name matching, outperforming the original PLMs across most P@k metrics.

For matching variable descriptions, our objective was to assess the relevance and impact of various data collection techniques (snippet extraction, scientific articles) and prompt-based data augmentation on TF–IDF. The results show that performance is relatively consistent across all GPT-3.5 Turbo-generated corpora and Microsoft snippets. The expert-curated article corpus achieved the best results at P@3 and P@5, while the Google snippets yielded the highest score at P@10. However, combining all corpora did not result in further performance gains. A detailed keyword density analysis confirmed that the corpora are not only diverse in linguistic structure but also relevant and aligned with the agroecological domain.

Our experiments highlight that although the combination method itself is not our innovation, the best results for P@1, P@3, and P@5 were obtained by combining BERT-base with TF–IDF applied to the fusion of all corpora. The highest performance at P@10 was achieved using the same combination, but with TF–IDF applied to *Corpus (Google)*. These results demonstrate that the combination method consistently outperforms using variable names or descriptions alone. To assess whether the improvements from the *Combination method* were statistically significant, we conducted Wilcoxon signed-rank tests (Wilcoxon, 1945), Friedman tests (Friedman, 1940), and Nemenyi post-hoc analyses (Nemenyi, 1963). The combination method, which integrates our extended BERT-base model with multi-head attention for name matching and TF–IDF using *Corpus (GPT-prompt 1)* for description matching, achieved statistically significant improvements over name matching alone at all P@k levels according to the Wilcoxon signed-rank test, and over description matching at P@3 and P@10 based on the Friedman test followed by the Nemenyi post-hoc test. Overall, the final MAEVa pipeline improved performance by more than 26 percentage points over name matching, more than 16 percentage points over description matching, and more than 2 percentage points over the combination method alone across all P@k metrics.

From a generative perspective, while we manually designed prompt strategies for GPT-3.5 to enrich sparse descriptions, future work could explore learning-based or adaptive prompt optimization. Recent studies have shown that prompts can be fine-tuned to improve downstream

[16] https://intercropvalues.eu/

Table A.11
Wilcoxon signed-rank test p-values comparing the Combination method against the Name-based and Description-based baselines across all experiments. Significant differences ($p < 0.05$) are highlighted in bold.

Combination Method		Corpus	P-value (combination vs name)				P-value (combination vs description)			
Name	Description		p@1	p@3	p@5	p@10	p@1	p@3	p@5	p@10
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Google) + Corpus (Microsoft) + Corpus (GPT) + Corpus (15 articles) + variable names and descriptions	**5.22×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0017**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Google)	**5.22×10^{-9}**	**1.86×10^{-6}**	**3.06×10^{-6}**	**5.90×10^{-5}**	**0.0010**	**0.0046**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 2)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 3)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 4)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 5)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 6)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 1) + Corpus (Google)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 1) + Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 2) + Corpus (Google)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 2) + Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 3) + Corpus (Google)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 3) + Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 4) + Corpus (Google)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 4) + Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 5) + Corpus (Google)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 5) + Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 6) + Corpus (Google)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 6) + Corpus (Microsoft)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
SBERT + Cos	TF–IDF	Corpus (GPT-prompt 1)	**3.38×10^{-5}**	**4.59×10^{-6}**	**1.62×10^{-6}**	**1.13×10^{-7}**	0.2850	0.6170	0.1266	0.5316
SBERT + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**5.90×10^{-5}**	**0.0004**	**0.0009**	**7.43×10^{-5}**	0.0593	**0.0253**	0.3173	**0.0290**
SimCSE + Cos	TF–IDF	Corpus (GPT-prompt 1)	**2.51×10^{-6}**	**1.23×10^{-6}**	**2.60×10^{-6}**	**2.06×10^{-6}**	0.7388	0.2059	**0.0076**	**0.0143**
SimCSE + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**1.49×10^{-6}**	**4.45×10^{-7}**	**5.33×10^{-7}**	**0.0001**	0.7962	0.5637	**0.0389**	**0.0253**
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**8.71×10^{-9}**	**1.13×10^{-6}**	**3.06×10^{-6}**	**0.0001**	**0.0028**	**0.0009**	**0.0125**	**0.0015**
BERT-base (2 HL) + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**5.9×10^{-5}**	**0.0004**	**0.0009**	**7.43×10^{-5}**	0.0593	**0.0253**	0.3173	**0.0290**

HLs: Hidden Layers, Hs: Heads, DT: Dropout regularization, UDW: Uniform Distribution of Weights and Cos: Cosine similarity

task performance (Zhou et al., 2022). Adaptive prompting during inference could further enhance the quality and relevance of the generated corpora (Diao et al., 2023). All of these proposed developments, along with those outlined in the Discussion section, aim to make MAEVa more robust, multilingual, semantically informed, and adaptable to the complexities of real-world agroecological data integration.

CRediT authorship contribution statement

Oussama Mechhour: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Sandrine Auzoux:** Validation, Supervision. **Clément Jonquet:** Validation, Supervision. **Mathieu Roche:** Validation, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Oussama Mechhour reports a relationship with Digital Agriculture Convergence Institute Digitag that includes: funding grants. Oussama Mechhour reports a relationship with Horizon Europe project IntercropVALUES that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially funded by the French National Research Agency (ANR) as part of the France 2030 program, under the reference ANR-16-CONV-0004 (#DigitAg), and by the Horizon Europe research and innovation program under grant agreement 101081973 (IntercropValuES). This research received support from the Regional Council of La Réunion, the French Ministry of Agriculture and Food, and the European Union (Feder program, grant AG/974/DAAF/2016-00096 and Feder program, grant GURTDI 20151501-0000735).

Table A.12
Nemenyi post-hoc test p-values comparing the Combination method against the Name-based and Description-based baselines across all experiments. Significant differences ($p < 0.05$) are highlighted in bold.

Combination Method		Corpus	P-value (combination vs name)				P-value (combination vs description)			
Name	Description		p@1	p@3	p@5	p@10	p@1	p@3	p@5	p@10
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Google) + Corpus (Microsoft) + Corpus (GPT)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4789
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Google) + Corpus (Microsoft) + Corpus (GPT) + Corpus (15 articles) + variable names and descriptions	$\mathbf{9.18 \times 10^{-5}}$	**0.0015**	**0.0034**	**0.0400**	0.1918	0.2370	0.5506	0.4789
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Google)	$\mathbf{9.18 \times 10^{-5}}$	**0.0022**	**0.0034**	**0.0293**	0.1530	0.3467	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 2)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 3)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 4)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 5)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 6)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 1) + Corpus (Google)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 1) + Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 2) + Corpus (Google)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 2) + Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 3) + Corpus (Google)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 3) + Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 4) + Corpus (Google)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 4) + Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 5) + Corpus (Google)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 5) + Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 6) + Corpus (Google)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 6) + Corpus (Microsoft)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
SBERT + Cos	TF–IDF	Corpus (GPT-prompt 1)	0.0713	**0.0400**	**0.0212**	**0.0015**	0.8886	0.9708	0.6967	0.9357
SBERT + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**0.0293**	0.0538	0.0713	0.0713	0.6239	0.4788	0.8886	0.5506
SimCSE + Cos	TF–IDF	Corpus (GPT-prompt 1)	**0.0151**	**0.0050**	**0.0050**	**0.0073**	0.9926	0.8886	0.4104	0.3467
SimCSE + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**0.0106**	**0.0022**	**0.0050**	**0.0293**	0.9926	0.9708	0.5506	0.4788
BERT-base (2 HL) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**0.0001**	**0.0015**	**0.0034**	**0.0400**	0.2370	0.2370	0.5506	0.4788
BERT-base (2 HL) + Multi-Head attention (256 Hs, DT=0.1 and UDW) + Cos	TF–IDF	Corpus (GPT-prompt 1)	**0.0293**	0.0538	0.0713	0.0713	0.6239	0.4788	0.8886	0.5506

HLs: Hidden Layers, Hs: Heads, DT: Dropout regularization, UDW: Uniform Distribution of Weights and Cos: Cosine similarity

Appendix. Statistical significance tests comparing the combination method separately to the name- and description-based baselines

Table A.11 presents the Wilcoxon signed-rank test results for all experiments reported in Table 6, excluding the string-based methods, which were included only for comparison purposes.

The results in Table A.11 show that the Combination method significantly outperforms:

- the Name-based method at all P@k levels, regardless of the corpus or model used.
- the Description-based method at all P@k levels, except in the following cases:
  - P@1 for SBERT and SimCSE (with and without the extended multi-head attention), and for BERT with extended multi-head attention;
  - P@3 for SBERT without extended multi-head attention, and for SimCSE (with and without extended multi-head attention);
  - P@5 for SBERT (with and without extended multi-head attention), and BERT with extended multi-head attention;
  - P@10 for SBERT without extended multi-head attention.

Table A.12 presents the Friedman test followed by the Nemenyi post-hoc test conducted on all experiments reported in Table 6, excluding the string-based methods, which were included only for comparison purposes.

The results in Table A.12 show that the Combination method significantly outperforms only the Name-based methods at all P@k levels, except in the following cases:

- P@1 for SBERT without extended multi-head attention;
- P@3, P@5, and P@10 for SBERT and BERT (with extended multi-head attention).

References

Akhmedovich, K., Sattarova, S., 2024. Using the jaccard similarity method for recommendation system of books. 5, http://dx.doi.org/10.47689/2181-1415-vol5-iss1-pp59-69.

Arnaud, E., Laporte, M.A., Kim, S., Aubert, C., Leonelli, S., Cooper, L., Jaiswal, P., Kruseman, G., Shrestha, R., Buttigieg, P.L., Mungall, C., Pietragalla, J., Agbona, A., Muliro, J., Detras, J., Hualla, V., Rathore, A., Das, R., Dieng, I., King, B., 2020. The ontologies community of practice: An initiative by the cgiar platform for big data in agriculture. SSRN Electron. J. http://dx.doi.org/10.2139/ssrn.3565982.

Auzoux, S., Christina, M., Goebel, F.R., Mansuy, A., Marion, D., 2018. A Dictionary of Variables to Harmonize Data from Agro-Ecological Experiments on Sugarcane. ISSCT.

Auzoux, S., Ngaba, B., Christina, M., Heuclin, B., Roche, M., 2023. Experimental variables in sugarcane intercropping in reunion island for data matching. Data Brief 46, 108869. http://dx.doi.org/10.1016/j.dib.2022.108869, URL: https://www.sciencedirect.com/science/article/pii/S2352340922010721.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems. URL: https://api.semanticscholar.org/CorpusID:14941970.

Caquet, T., Gascuel, C., Tixier-Boichard, M., Dedieu, B., Détang-Dessendre, C., Dupraz, P., Faverdin, P., Hazard, L., Hinsinger, P., Litrico-Chiarelli, I., Médale, F., Monod, H., Petit, S., Reboud, X., Thomas, A., Lescourret, F., Roques, L., de Vries, H., Soussana, J.-F., 2019. Réflexion prospective interdisciplinaire pour l'agroécologie. Rapport de synthèse. p. 108. http://dx.doi.org/10.15454/heimwa, il s'agit d'un type de produit dont les métadonnées ne correspondent pas aux métadonnées attendues dans les autres types de produit : REPORT. URL: https://hal.science/hal-02154433.

Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., Kurzweil, R., 2018. Universal sentence encoder. arXiv:1803.11175.

Chai, Y., Xie, H., Qin, J.S., 2025. Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities. arXiv:2501.18845.

Chaudhary, A., 2020. A Visual Survey of Data Augmentation in NLP. URL: https://amitness.com/posts/data-augmentation-for-nlp.html.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V., 2020. Unsupervised cross-lingual representation learning at scale. In: Proceedings of ACL.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., 2018. Supervised learning of universal sentence representations from natural language inference data. arXiv:1705.02364.

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. pp. 4171–4186.

Diao, Y., Qin, L., Liu, T., 2023. Active prompting with chain-of-thought for large language models. arXiv preprint arXiv:2302.12246.

Dieudonat, L., Han, K., Leavitt, P., Marquer, E., 2020. Exploring the combination of contextual word embeddings and knowledge graph embeddings. ArXiv, arXiv:2004.08371.URL: https://api.semanticscholar.org/CorpusID:215814244.

Fellah, A., Zahaf, A., Elçi, A., 2024. Semantic similarity measure using a combination of word2vec and wordnet models. Indones. J. Electr. Eng. Informatics (IJEEI) 12 (2), 455–464.

Feng, S.Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., Hovy, E., 2021. A survey of data augmentation approaches for NLP. arXiv:2105.03075.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W., 2022. Language-agnostic BERT sentence embedding. Trans. Assoc. Comput. Linguist. 10, 194–206.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E., 2001. Placing search in context: The concept revisited. 20, pp. 406–414. http://dx.doi.org/10.1145/503104.503110,

Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. 11 (1), 86–92.

Ganitkevitch, J., Van Durme, B., Callison-Burch, C., 2013. PPDB: The paraphrase database. In: Vanderwende, L., Daumé III, H., Kirchhoff, K. (Eds.), Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Atlanta, Georgia, pp. 758–764, URL: https://aclanthology.org/N13-1092.

Gao, T., Yao, X., Chen, D., 2021. SimCSE: Simple contrastive learning of sentence embeddings. In: Moens, M.-F., Huang, X., Specia, L., Yih, S.W.-t. (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 6894–6910. http://dx.doi.org/10.18653/v1/2021.emnlp-main.552, URL: https://aclanthology.org/2021.emnlp-main.552/.

Garg, S., Ramakrishnan, G., 2020. BAE: BERT-based adversarial examples for text classification. In: Webber, B., Cohn, T., He, Y., Liu, Y. (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. EMNLP, Association for Computational Linguistics, Online, pp. 6174–6181. http://dx.doi.org/10.18653/v1/2020.emnlp-main.498, URL: https://aclanthology.org/2020.emnlp-main.498/.

Gerz, D., Vulić, I., Hill, F., Reichart, R., Korhonen, A., 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In: Su, J., Duh, K., Carreras, X. (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, pp. 2173–2182. http://dx.doi.org/10.18653/v1/D16-1235, URL: https://aclanthology.org/D16-1235.

Gomaa, W.H., Fahmy, A.A., 2013. A survey of text similarity approaches. Int. J. Comput. Appl. 68, 13–18, URL: https://api.semanticscholar.org/CorpusID:2703920.

Hamming, R.W., 1950. Error detecting and error correcting codes. Bell Syst. Tech. J. 29 (2), 147–160.

Harris, Z.S., 1954. Distributional structure. 10, (2–3), Taylor & Francis, pp. 146–162,

Hassan, R.T., Ahmed, N.S., 2023. Evaluating of efficacy semantic similarity methods for comparison of academic thesis and dissertation texts. Sci. J. Univ. Zakho 11 (3), 396. http://dx.doi.org/10.25271/sjuoz.2023.11.3.1120, URL: https://sjuoz.uoz.edu.krd/index.php/sjuoz/article/view/1120.

Hill, F., Reichart, R., Korhonen, A., 2014. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. arXiv:1408.3456.

Jaccard, P., 1901. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines.. Bull. de la Soc. Vaudoise Des Sci. Nat. 37, 241–272. http://dx.doi.org/10.5169/seals-266440.

Jadon, S., Jadon, A., 2023. An overview of deep learning architectures in few-shot learning domain. arXiv:2008.06365.

Jadon, A., Kumar, S., 2023. 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). IEEE, http://dx.doi.org/10.1109/smartnets58706.2023.10215825.

Jadon, A., Patil, A., Jadon, S., 2022. A comprehensive survey of regression based loss functions for time series forecasting. arXiv:2211.02989.

Jaro, M.A., 1989. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. J. Amer. Statist. Assoc. 84 (406), 414–420. http://dx.doi.org/10.1080/01621459.1989.10478785.

Jiang, N., de Marneffe, M.C., 2019. 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. pp. 4208–4213.

Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.-A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E., 2023. Mistral 7B. arXiv:2310.06825.

Jones, K.S., 1972. A statistical interpretation of term specificity and its application in retrieval. J. Doc. 28 (1), 11–21.

Jonquet, C., Toulet, A., Arnaud, E., Aubin, S., Dzalé Yeumo, E., Emonet, V., Graybeal, J., Laporte, M.-A., Musen, M.A., Pesce, V., Larmande, P., 2018. AgroPortal: A vocabulary and ontology repository for agronomy. Comput. Electron. Agric. 144, 126–143. http://dx.doi.org/10.1016/j.compag.2017.10.012, URL: https://www.sciencedirect.com/science/article/pii/S0168169916309541.

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T., 2016. FastText: Efficient learning of word representations and sentence classification. arXiv preprint arXiv:1607.01759.

Le, Q.V., Mikolov, T., 2014. Distributed representations of sentences and documents. arXiv:1405.4053.

Lerer, A., Wu, L.Y., Shen, J., Lacroix, T., Wehrstedt, L., Bose, A., Peysakhovich, A., 2019. Pytorch-BigGraph: A large-scale graph embedding system. ArXiv.arXiv:1903.12287.URL: https://api.semanticscholar.org/CorpusID:88523916.

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Dokl. 10 (8), 707–710.

Lewis, M., et al., 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 37th International Conference on Machine Learning.

Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.-C., 2020. Deep entity matching with pre-trained language models. 14 (1), 50–60. http://dx.doi.org/10.14778/3421424.3421431, URL: https://doi.org/10.14778/3421424.3421431.

Li, H., Xu, J., et al., 2014. Semantic matching in search. Found. Trends® Inf. Retr. 7 (5), 343–469.

Lin, D., 1998. An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning. ICML, pp. 296–304.

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.

Liu, Y., Xiong, C., Callan, J., 2023. Pre-train prompt tuning for low-resource retrieval. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. ACL.

Luong, T., Socher, R., Manning, C., 2013. Better word representations with recursive neural networks for morphology. In: Hockenmaier, J., Riedel, S. (Eds.), Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, Sofia, Bulgaria, pp. 104–113, URL: https://aclanthology.org/W13-3512.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press.

Mechhour, O., Auzoux, S., Jonquet, C., Roche, M., 2025. Corpus and list of agroecological experimental variables. CIRAD Dataverse, http://dx.doi.org/10.18167/DVN1/9X3IVR.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K., 1995. WordNet: An online lexical database. Commun. ACM 38 (11), 39–41. http://dx.doi.org/10.1145/219717.219748.

Mueller, J., Thyagarajan, A., 2016. Siamese recurrent architectures for learning sentence similarity. 30, http://dx.doi.org/10.1609/aaai.v30i1.10350, URL: https://ojs.aaai.org/index.php/AAAI/article/view/10350.

Nemenyi, P., 1963. Distribution-free multiple comparisons (Ph.D. thesis). Princeton University.

OpenAI, 2023. GPT-4. Technical Report, URL: https://openai.com/research/gpt-4.

Patil, A., Han, K., Jadon, A., 2023. A comparative study of text embedding models for semantic text similarity in bug reports. arXiv:2308.09193.

Pennington, J., et al., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP, pp. 1532–1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. arXiv:1802.05365.

Pham, H., Luong, M.T., Manning, C.D., 2015. Learning distributed representations for multilingual text sequences. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 88–94.

Po, D., 2020. Similarity based information retrieval using levenshtein distance algorithm. Int. J. Adv. Sci. Res. Eng. 06, 06–17. http://dx.doi.org/10.31695/IJASRE.2020.33780.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing. EMNLP.

Rezayi, S., Liu, Z.L., Wu, Z., Dhakal, C., Ge, B., Zhen, C., Liu, T., Li, S., 2022. AgriBERT: Knowledge-infused agricultural language models for matching food and nutrition. In: International Joint Conference on Artificial Intelligence. URL: https://api.semanticscholar.org/CorpusID:250635911.

Riedel, S., Yao, L., McCallum, A., 2010. Modeling relations and their mentions without labeled text. In: ECML/PKDD. URL: https://api.semanticscholar.org/CorpusID:2386383.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M., 1994. Okapi at TREC-3. In: Proceedings of the Third Text Retrieval Conference (TREC-3).

Romualdo, A., Real, L., Caseli, H., 2021. Measuring Brazilian portuguese product titles similarity using embeddings. In: Proceedings of the 13th Brazilian Symposium in Information and Human Language Technology. SBC, Porto Alegre, RS, Brasil, pp. 121–132. http://dx.doi.org/10.5753/stil.2021.17791, URL: https://sol.sbc.org.br/index.php/stil/article/view/17791.

Rubenstein, H., Goodenough, J.B., 1965. Contextual correlates of synonymy. Commun. ACM 8 (10), 627–633. http://dx.doi.org/10.1145/365628.365657, URL: https://doi.org/10.1145/365628.365657.

Salle, A., Villavicencio, A., Idiart, M., 2016. Matrix factorization using window sampling and negative sampling for improved word representations. In: Erk, K., Smith, N.A. (Eds.), Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Berlin, Germany, pp. 419–424. http://dx.doi.org/10.18653/v1/P16-2068, URL: https://aclanthology.org/P16-2068.

Sorensen, T., Dice, L.R., 1948. The comparison of two samples. Amer. Nat. 50 (594), 159–175.

Souza, F., Nogueira, R., Lotufo, R., 2020. Bertimbau: Pretrained BERT models for Brazilian portuguese. pp. 403–417. http://dx.doi.org/10.1007/978-3-030-61377-8_28.

Speer, R., Havasi, C., 2012. Representing general relational knowledge in ConceptNet 5. In: Calzolari, N., Choukri, K., Declerck, T., Do˘ gan, M.U.g., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (Eds.), Proceedings of the Eighth International Conference on Language Resources and Evaluation. LREC'12, European Language Resources Association (ELRA), Istanbul, Turkey, pp. 3679–3686, URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/1072_Paper.pdf.

Sun, C., Zhou, Y., Lin, Y., Yu, P., 2023. Augmenting text with large language models for low-resource tasks. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. EMNLP.

Susanto, A., Muliadi, N., Nugroho, B., Muljono, M., 2023. Comparison of string similarity algorithm in post-processing OCR. J. Appl. Intell. Syst. URL: https://api.semanticscholar.org/CorpusID:259463513.

Toshevska, M., Stojanovska, F., Kalajdjieski, J., 2020. The ability of word embeddings to capture word similarities. Int. J. Nat. Lang. Comput. 9, 25–42. http://dx.doi.org/10.5121/ijnlc.2020.9302.

Touvron, H., et al., 2023. LLaMA: Open and efficient foundation language models. ArXiv Preprint arXiv:2302.13971.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G., 2016. Complex embeddings for simple link prediction. ArXiv. arXiv:1606.06357.URL: https://api.semanticscholar.org/CorpusID:15150247.

Tversky, A., 1977. Features of similarity. Psychol Rev 84 (4), 327–352. http://dx.doi.org/10.1037/0033-295X.84.4.327, URL: http://psycnet.apa.org/journals/rev/84/4/327/.

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D.F., Chao, L.S., 2019. Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787.

Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biom. Bull. 1 (6), 80–83.

Winkler, W.E., 1990. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. Proc. Sect. Surv. Res. Methods 354–359.

Wu, Z., Palmer, M., 1994. Verb semantics and lexical selection. In: 32nd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Las Cruces, New Mexico, USA, pp. 133–138. http://dx.doi.org/10.3115/981732.981751, URL: https://aclanthology.org/P94-1019.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., Le, Q.V., 2020. Unsupervised data augmentation for consistency training. arXiv:1904.12848.

Xie, T., Dai, K., Wang, K., Li, R., Zhao, L., 2024. DeepMatcher: A deep transformer-based network for robust and accurate local feature matching. Expert Syst. Appl. 237, 121361. http://dx.doi.org/10.1016/j.eswa.2023.121361, URL: https://www.sciencedirect.com/science/article/pii/S0957417423018638.

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.

Zhang, X., Zhao, J., LeCun, Y., 2015. Character-level convolutional networks for text classification. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 28, Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Zhou, X., Wu, T., Xu, X., Yan, W., Yin, W.-t., 2022. Large language models are human-level prompt engineers. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.