computers

MDPI

Article

Hybrid Methods for Automatic Collocation Extraction in Building a Learners' Dictionary of Italian

Damiano Perri [1,*], Osvaldo Gervasi [1], Sergio Tasso [1], Stefania Spina [2], Irene Fioravanti [2], Fabio Zanda [2] and Luciana Forti [3]

[1] Department of Math and Computer Science, University of Perugia, Via Luigi Vanvitelli, 1, 06123 Perugia, Umbria, Italy; osvaldo.gervasi@unipg.it (O.G.); sergio.tasso@unipg.it (S.T.)
[2] Department of Italian Language, Literature and Arts, University for Foreigners of Perugia, Piazza Fortebraccio, 4, 06123 Perugia, Umbria, Italy; stefania.spina@unistrapg.it (S.S.); irene.fioravanti@unistrapg.it (I.F.); fabio.zanda@unistrapg.it (F.Z.)
[3] Department of Languages, Literature and Modern Cultures, University of Chieti 'G. d'Annunzio', Via dei Vestini, 31, 66100 Chieti, Abruzzo, Italy; luciana.forti@unich.it
* Correspondence: damiano.perri@unipg.it

Abstract

The automatic construction of learners' dictionaries requires robust methods for identifying non-literal word combinations, or collocations, which represent a significant challenge for second-language (L2) learners. This paper addresses the critical initial step of accurately extracting collocation candidates from corpora to build a learner's dictionary for Italian. The adopted method and the implemented application are significant for learning the Italian language. We present a comparative study of three methodologies for identifying these candidates within a 41.7-million-word Italian corpus: a Part-Of-Speech-based approach, a syntactic dependency-based approach, and a novel Hybrid method that integrates both. The analysis yielded 2,097,595 potential collocations. Results indicate that the Hybrid method achieves superior performance in terms of Recall and Benchmark Match, identifying the most significant portion of candidates, 42.35% of the total. We conducted an in-depth analysis to refine the extracted dataset, calculating multiple statistical metrics for each candidate, which are described in detail in the paper. Such analysis allows for the classification of collocations by relevance, difficulty, and frequency of use, forming the basis for the future development of a high-quality, web-based dictionary tailored to the proficiency levels of Italian learners.

Keywords: Artificial Intelligence; Natural Language Processing; language learning; collocations; part-of-speech; syntactic dependency; L2 learners

Academic Editor: Paolo Bellavista

Received: 12 November 2025
Revised: 9 December 2025
Accepted: 10 December 2025
Published: 12 December 2025

Citation: Perri, D.; Gervasi, O.; Tasso, S.; Spina, S.; Fioravanti, I.; Zanda, F.; Forti, L. Hybrid Methods for Automatic Collocation Extraction in Building a Learners' Dictionary of Italian. *Computers* **2025**, *14*, 552. https://doi.org/10.3390/computers14120552

1. Introduction

This work aims to create a learner's dictionary by automatically identifying the most appropriate combinations of words that are commonly used in spoken language and whose meanings are not literal. These word combinations, called collocations, are of fundamental importance in learning a language and often represent a significant barrier for learners to overcome. A collocation consists of two or more words within a sentence that have a semantic meaning different from the literal interpretation of the two terms and are used, often unconsciously, in both written and spoken language [1–3].

The Artificial Intelligence (AI) and Natural Language Processing (NLP) communities have extensively studied collocations, particularly in English, due to their significance in

language learning and communication [4–6]. Multi-word expressions (MWEs), encompassing constructs such as collocations, idioms, and lexical bundles, are lexical units formed by two or more words [7].

Significant progress has been made in developing resources, such as collocation dictionaries tailored for second-language (L2) learners, primarily for English [5,6]. At the same time, general collocation dictionaries, which are not explicitly aimed at L2 learners, exist for several languages, including English [4] and Italian [8–10]. The application of language corpora has greatly advanced the investigation of phrases and their lexicographical use. In particular, corpora have been used in identifying recurring word combinations. Such corpora leverage NLP and statistical methods to uncover patterns in vast text datasets [11].

Extracting MWEs from corpora involves two key tasks [12]. The first is the automated identification of candidates based on predetermined grammatical or syntactic criteria. The second step is to filter these candidates to detect phraseologically significant combinations, such as collocations, using measures like frequency and statistical association. This study focuses on identifying potential collocations in Italian corpora, a crucial step in creating a learner-oriented dictionary of collocations. To do this, several syntactic rules have been defined, which, starting from the logical-grammatical analysis obtained from pre-trained models on Italian, such as the libraries *spaCy* [13] and UDpipe [14], make it possible to identify word combinations that comply with the defined rules and classify them by type.

We posit that the success of subsequent steps in creating learner-centric collocation dictionaries heavily relies on the accuracy of candidate identification. Enhanced precision in identifying potential collocations leads to more reliable frequency data, which improves the association measures used to discard irrelevant combinations. Consequently, the overall process becomes more accurate.

This research presents an experiment to assess a Hybrid approach for detecting collocation candidates. We compare the two predominant methods and their combination, evaluating their effectiveness in identifying collocations in Italian corpora. The first one is the Part-Of-Speech-based (P-based) method, while the second is the syntactic dependency-based (S-based) method [15]. We have developed an integrated technique, blending these methods, which is termed the Hybrid approach.

Linguistic pre-processing steps, such as Part-Of-Speech (POS) tagging and dependency parsing, are utilised in collocation extraction to refine candidate identification. Although the P-based approach, rooted in robust NLP tasks such as POS tagging, effectively identifies positional patterns, it has its shortcomings. It cannot capture syntactic relationships or handle non-standard sentence structures. For instance, it might miss the verb-object relationship between 'play' and 'role' in a sentence like Example 1. Similar examples are reported in [12].

Example 1. *People need to observe and understand the reality around them, the social media plays, as numerous studies have shown, a significant role in shaping public opinion.*

Conversely, an S-based approach, which relies on parsed data, is better equipped to detect the verb-direct object relationship, making it particularly effective in identifying candidate collocations with syntactic dependencies. Unlike the P-approach, the S-approach is not constrained by the distance between words, offering greater flexibility in recognising combinations spread across a sentence. However, it suffers from parsing errors, which have been reported to account for 7.85% to 9.7% of the extracted candidate collocations [16,17]. Despite advancements in parsing accuracy [18,19], limitations persist. The S-approach provides minimal insight into how words interact and struggles to distinguish between frequent combinations and idiomatic expressions that share identical syntactic structures [15].

Our Hybrid method has been developed to identify collocation candidates from corpora, with a focus on Italian rather than English. This integrated approach will likely outperform existing methods in candidate detection. Additionally, we aim to explore scenarios where the Hybrid approach proves superior and identify potential areas for further refinement.

The article is composed as follows: Section 2 analyses the current literature and provides the background needed to understand the contents of the manuscript, Section 3 describes how the texts were analysed and how the preliminary dataset of collocations was constructed, Section 4 describes the statistics that were calculated for each collocation and explains how the dictionary was made, Section 6 gives the final considerations regarding the work done and describes possible future developments.

2. Related Work

This section provides a brief review of key methods and NLP techniques employed in detecting or uncovering collocation candidates from corpora [20], while setting aside the measures used to validate phraseological significance, a subsequent step in assembling entries for lexicographic applications.

Early work in NLP focused on identifying collocation candidates, primarily through frequent word sequences, utilising n-gram models to extract these patterns from corpora [21,22]. N-gram models are statistical language models that predict the next word in a sequence based on the previous n-1 words. However, these methods often struggled with the complexity of natural language, particularly in capturing syntactic relationships and semantic nuances. This search, referred to as "looking for needles in a haystack" [22], has increasingly leveraged linguistically pre-processed data, including lemmatised and POS-tagged corpora. This shift has been especially beneficial for languages with rich morphological systems and less rigid word orders [23].

The P-approach, which emerged early due to the widespread availability of POS-tagged corpora, became foundational. Many extraction systems based on this approach rely on predefined combinations of POS tags (e.g., verb-noun, adjective-noun). Early studies demonstrated significant improvements in detection accuracy when POS filtering was applied [24–27]. These advances primarily benefited fixed and adjacent word pairs, where fundamental linguistic analysis suffices to capture elementary grammatical patterns.

In recent years, researchers have advocated for incorporating more refined linguistic analyses to enhance the detection of candidate collocations. Seretan's work highlighted the potential of syntactic dependencies, emphasising their capacity to identify non-contiguous and syntactically flexible collocations based on word relationships, thereby improving the overall accuracy of results [12]. Despite these advances, systems leveraging the S-approach for multi-word expression (MWE) detection have encountered significant challenges due to high parsing error rates. Parsing, although beneficial for initial syntactic analysis [20], has been documented as a persistent source of errors that impact detection quality. Some studies have quantified this problem by reporting an error rate that in some cases exceeds 7% [28], while identifying relative constructions as the cause of nearly half the missed candidate collocations [29].

These limitations underline that while S-based methods show promise, they remain hampered by parsing inaccuracies. To address these issues, researchers have increasingly advocated for Hybrid approaches. These approaches aim to mitigate their shortcomings by integrating the strengths of P- and S-based methods. This perspective is summarised by the observation that *"the two methods seem to be highly complementary rather than competing with one another"* [15].

Recent studies have attempted to operationalise this integration, developing a system combining POS-tagging with dependency parsing to identify single-token and multi-token verbal MWEs [30]. Their systems achieved the best results with verb-particle constructions, accurately identifying approximately 60% of such cases, although success rates for other MWE types remained around 40%. Similarly, Shi and Lee [31] introduced a joint approach that combines scores from POS tagging and dependency parsing to extract headless MWEs. Their findings indicated that tagging surpasses parsing in accuracy for flat-structure MWEs. However, the joint method delivered higher overall precision, primarily due to the combined contributions of the two techniques [32]. Artificial intelligence can also help to analyse language effectively with the help of software such as *spacy* [13] or UDPipe [33] libraries, which can also take advantage of modern graphics accelerators (GPUs) and their computational power [34]. These approaches could be further improved by the possibility of enriching the information in the input datasets, also through the production of synthetic data designed ad hoc or possibly generated by modern Large Language Models (LLMs). These techniques have yielded significant results in various fields, ranging from telerehabilitation to automatic object recognition within images [35,36].

3. Methodology

This section describes the methodology used to analyse the set of sample texts on which the dictionary's structure is based. The set of these texts is also referred to as a Corpus. To achieve the set objectives, we have designed a prototype system based on a small number of sample texts to evaluate the goodness of the adopted method. It must be considered that analysing a large Corpus requires a significant amount of time for classification and investigation on the part of the linguists involved. Therefore, we first validated the method on a restricted sample of texts, and then extended the examination to the complete Corpus.

3.1. The Corpus

The Corpus used for this study is a collection of Italian texts, comprising approximately 41.7 million words, which includes both written and spoken language [37]. The Corpus was balanced between written and spoken registers to ensure a comprehensive representation of the Italian language. To ensure diversity, the written segment included two newspaper articles (a report and an editorial), two school essays, and a tourism blog post. The spoken portion featured conference transcripts, political speeches, and dialogue from television series.

The texts were selected to represent a range of genres and registers, including the following topics: *Academic writing*, *Administrative writing*, *Literary fiction*, *Non-fiction*, *School essays*, *Newspapers*, *Web texts*, *Tv programs*, *Film dialogues* and *Spoken texts*. The distribution of words across these categories is reported in Table 1. To evaluate statistics on the Corpus, it is essential to determine the total number of tokens in the entire dataset. A token is a sequence of characters that is grouped and treated as a single unit in the analysis, such as a portion of words, punctuation marks, or other meaningful elements in the text. From this Corpus, the total number of tokens was calculated to be 47,944,818, because of this the number of tokens exceeds the number of words constituting the Corpus.

We extracted eight texts randomly from the Corpus, comprising approximately 8000 tokens. The selected texts were categorized as follows: *news report* is extracted from *Newspapers*, *editorial* from *Newspapers*, *school essay 1* and *school essay 2* from *School essays*, *blog post* from *Web texts*, *conference* from *Spoken texts*, *political speech* from *Spoken texts*, and *tv series* from *Tv programs*. This selection offers a realistic simulation of candidate collocation

extraction across various genres and registers, providing insights into the robustness of the three approaches in different textual contexts and styles.

Table 1. The composition of the Corpus PEK24 and its sections.

Section	Nr. of Texts	Tokens	Words
Written Sections			
Academic writing	315	2,003,969	1,765,584
Administrative writing	194	1,914,625	1,703,146
Literary fiction	90	6,623,697	5,593,287
Non-fiction	107	3,172,781	2,739,136
School essays	25,137	6,989,768	6,192,514
Newspapers	104,433	6,902,522	5,973,817
Web texts	105,967	11,266,851	9,788,067
Spoken Sections			
Tv programs	196	1,556,099	1,366,442
Film dialogues	116	1,107,484	858,492
Spoken texts	2376	6,407,022	5,717,846
Total	144,931	47,944,818	41,698,331

3.2. The Candidate Collocations

This subsection outlines the process of extracting candidate collocations from the test Corpus, which consists of a limited number of examples, focusing on the logical and grammatical components considered and the logical assumptions devised for their identification.

We structured our experiment to replicate the "natural" processes anticipated for the eventual extraction of entries intended for a learner dictionary of Italian collocations. For this reason, we avoided pre-selecting target words or lemmas during the experiment and instead considered all word pairs generated from the sample text.

The only exception is the preselection of two specific syntactic dependencies: *verb + direct object* (Vdobj) and *adjective modifier* (amod) preceding or following a noun (reflecting Italian's flexible word order).

Previous studies demonstrate that among the eight most frequent syntactic structures forming Italian collocations *verb + direct object*, *amod*, *noun + preposition + noun*, *noun + noun*, *verb + adjective*, *verb + adverb*, *noun + conjunction + noun*, and *adjective + conjunction + adjective*, two of them are particularly relevant: *Vdobj* and *amod* alone account for over 50% of the total [38]. Additionally, these two relations differ notably in how their components interact, particularly in terms of distance.

The distance between components can span several words in *verb + directobject* (Vdobj) combinations. For example, in the phrase «*Non fare* ***promesse*** *che non riuscirai mai a* ***mantenere***», the verb (*mantenere*) and its object (*promesse*) are separated by five words, including a relative pronoun. Conversely, in adjective modifier (amod) relations, the two components are typically adjacent (as in *nuovo libro*, 'new book') or close to each other (e.g., *libro estremamente interessante*, 'a fascinating book'). This distinction makes the chosen relations particularly useful for evaluating the challenges and capabilities of different systems in detecting syntactically diverse collocations.

3.2.1. S-BasedApproach

The S-based approach identified candidate collocations as syntactically related lexical pairs conforming to grammatical constraints (e.g., nouns, adjectives, and verbs). Text parsing was conducted using the Universal Dependencies (UD) framework [39] implemented via *spaCy*, an AI-driven linguistic analysis library. In this specific framework, the standard

dependency label 'obj' identifies the direct object, which corresponds to the *Vdobj* relation targeted in our study.

This procedure yielded more candidate collocations (685), as it captures discontinuous and flexible dependencies, which are particularly relevant for syntactic relations like *Vdobj*.

3.2.2. P-Based Approach

The P-based approach relies on POS tagging, implemented using *TreeTagger* [40], trained with an ad hoc tagset featuring 54 fine-grained POS tags [41]. Queries were formulated for the *Corpus Workbench* [42] and *Corpus Query Processor* (CQP) tools to detect *Vdobj* and *amod* relations. Queries integrate:

1. A POS tag sequences for ADJ, NOUN, and VERB categories.
2. Regex expressions to handle potential intervening constituents like articles, conjunctions, or adverbs.
3. A lemma-based exclusion list to filter common intransitive verbs.

This approach identified 549 candidate collocations.

3.2.3. Hybrid Approach

The Hybrid approach combined the outputs of the P-based and S-based approaches. It integrated common candidates and those uniquely identified by each approach, resulting in 748 candidate collocations. By merging these techniques, the Hybrid approach seeks to maximise precision and recall, addressing the limitations of each standalone method.

3.3. Annotation

To validate system performance, the outputs were compared against human-annotated data. Two expert Italian linguists manually extracted *amod* and *Vdobj* combinations from the sample texts, adhering strictly to the syntactic relations criterion. Thanks to the clearly defined guidelines and the domain expertise of the annotators, a perfect inter-annotator agreement was observed during the independent annotation phase (Cohen's $\kappa = 1.0$). This process generated a benchmark set of 610 candidate collocations for subsequent evaluation.

The comparison with this benchmark enables us to evaluate the accuracy, recall, and precision of each system under study, providing insights into their relative effectiveness for Italian lexicographic applications.

3.4. The Computational Procedure

The computational process comprises three steps to ensure thorough and consistent data processing: preliminary pre-processing of the input texts, sentence parsing with rules for syntactic dependency recognition, and statistical analysis of the results comparing the S-, P-, and Hybrid approaches.

3.4.1. The Pre-Processing of the Input Texts

The initial step focused on standardising the input data format and eliminating irrelevant elements to improve analysis consistency. Pre-processing steps have been applied to the input texts, which were essential for preparing the data for subsequent analysis. First, the Corpus is transformed into a list of sentences, where each sentence is represented as a single row. At the beginning of each sentence, a capital letter is inserted, and at the end, a full stop is added to ensure that each phrase is correctly formatted. To ensure a clean input, all double whitespaces are removed, and any excess whitespace at the end of sentences is eliminated. After normalization, the semantic content of the sentences remains unaltered. This ensures the consistency of the analyzed text and has a negligible impact on the portion of the corpus that does not originate from spoken sources. The sentences were

subsequently inserted into a Tab-Separated Values (TSV) file. Each sentence was associated with the row number related to its position in the original text, the file ID (indicating the source text), and the word count and token count for that sentence.

This preparatory step is critical for obtaining clean and easily understandable data for subsequent.

To ensure that these normalization steps did not introduce artifacts or alter the syntactic interpretation of the text (e.g., by unintentionally merging tokens due to whitespace removal), we performed a parse sanity check. We compared the tokenization and dependency labels of randomly selected sentences before and after pre-processing. Table 2 presents an example demonstrating that the normalization process preserves the integrity of the core syntactic dependencies while ensuring proper sentence segmentation.

Table 2. Parse sanity check: comparison of tokenization and dependency labels before and after pre-processing.

Before Pre-Processing			**After Pre-Processing**		
Raw Input:	`1: "Mi manchi tanto` `2: anche tu mi manchi."`		*Clean Input:*	`1: Mi manchi tanto.` `2: Anche tu mi manchi.`	
Token	**POS**	**Dep**	**Token**	**POS**	**Dep**
Mi	PRON	obj	Mi	PRON	obj
manchi	VERB	ROOT	manchi	VERB	ROOT
tanto	ADV	advmod	tanto	ADV	advmod
			.	PUNCT	punct
anche	ADV	advmod	Anche	ADV	advmod
tu	PRON	nsubj	tu	PRON	nsubj
mi	PRON	obj	mi	PRON	obj
manchi	VERB	ROOT	manchi	VERB	ROOT
.	PUNCT	punct	.	PUNCT	punct

3.4.2. The Analysis of the Corpus

In the subsequent phase, the sentences of the Corpus have been analysed using *spaCy* library with a focus on identifying two syntactic dependencies: verb-direct object (*Vdobj*) and adjective modifier (*amod*).

The software used to analyze the Corpus was entirely custom-developed using the Python 3.13 programming language and the analysis capabilities provided by spaCy library. The analysis was conducted on an HPE ProLiant DL380 Gen10 server equipped with 256 GB of RAM and two Intel® Xeon® Silver 4309Y CPUs @ 2.80 GHz, totaling 16 cores and 32 threads; each CPU features 12 MB of cache and a maximum (Turbo) frequency of 3.60 GHz. The Python-based syntactic analysis employed the pre-trained *it_core_news_lg* pipeline for the Italian language, optimised for CPUs (https://github.com/explosion/spacy-models/releases/tag/it_core_news_lg-3.8.0 (accessed on 30 September 2024).

The Italian Stanford Dependency Treebank (UD Italian ISDT v2.8; [43]) was used to train the model. The language model spans 541 MB of written texts, including media and news.

While various libraries are available for grammatical analysis, we chose *spaCy* library for its state-of-the-art Italian language model, updated as recently as 30 September 2024 (https://spacy.io/models/it (accessed on 30 September 2024).) This ensures compatibility with contemporary linguistic standards and optimised dependency parsing performance.

By employing this approach, we established a syntactic structure that identifies *amod* and *Vdobj* relations with high precision, setting the foundation for comparative evaluation across systems.

Each sentence in our dataset was scrutinised at the level of individual words. For every word, the *spaCy* library generates a structured output comprising the following elements: *wordId*, *Form*, *Lemma*, *UPosTag*, *XPosTag*, *head.i* and *DepRel*. The meanings of these fields are detailed in Table 3.

Table 3. Description of the fields in the dataset generated by *spaCy* library.

Field	Description
wordId	serves as a unique identifier for each word within the sentence, facilitating precise tracking and reference.
Form	Reflects the surface representation of the word as it appears in the text.
Lemma	Captures the canonical dictionary form of the word.
UPosTag	Universal Part of Speech Tag categorises the word grammatically according to the universal POS tag schema.
XPosTag	Extended part-of-speech tag provides additional granular detail about the part-of-speech classification.
head.i	Head index denotes the index position of the word's syntactic parent within the sentence's dependency structure.
DepRel	Specifies the syntactic relationship connecting the word to its governing word in the sentence.
len	Indicates the length of the word in characters.
f	This is a special column added to indicate whether a word has been identified as part of a collocation, in which case it is marked with an asterisk (*).

An example of the output generated by *spaCy* library for the sentence «*Questa cosa funziona se il messaggio è consegnato intatto.*» is shown in Table 4.

Table 4. Example of the output generated by *spaCy* library for the sentence «*Questa cosa funziona se il messaggio è consegnato intatto.*»

wordId	Form	Lemma	UPosTag	XPosTag	head.i	DepRel	len	f
1	Questa	questo	DET	DD	2	det	6	
2	cosa	cosa	NOUN	S	3	nsubj	4	
3	funziona	funzionare	VERB	V	3	ROOT	8	
4	se	se	SCONJ	CS	8	mark	2	
5	il	il	DET	RD	6	det	2	
6	messaggio	messaggio	NOUN	S	8	nsubj:pass	9	*
7	è	essere	AUX	VA	8	aux:pass	1	
8	consegnato	consegnare	VERB	V	3	advcl	10	
9	intatto	intatto	ADV	B	8	advmod	7	
10	.	.	PUNCT	FS	3	punct	1	

While these outputs serve as foundational linguistic features, they do not suffice for a complete understanding of the sentence's syntactic or semantic structure. To fill this gap, we defined custom syntactic rules implemented as Python functions. These rules examine each word and its syntactic parent to determine whether they form an *amod* or *Vdobj* pairing. Leveraging the linguistic data provided by the parser, the rules played a pivotal role in boosting the model's accuracy and recall.

The development of these rules was intricate due to the morphological and syntactic complexity of Italian, a language characterised by rich inflexion and relatively flexible word order. Our approach was iterative: analysing intermediate results, identifying misclassifications, and adding new rules to capture a broader range of patterns. This process enabled the model to identify numerous valid word combinations.

It is noteworthy that the Python rules were meticulously crafted with the nuances of the Italian language in mind.

Among the most significant rules implemented, one function identifies a direct verbal object (*Vdobj*) by checking for the *obj* dependency linked to the root. At the same time, it ensures that the *UPosTag* of the root corresponds to a *VERB* element type.

The rule shown in Listing 1 can recognise the combination of words *messaggio consegnato* in Example 2.

Listing 1. Example of a function to recognize *Vdobj*.

```
def checkFn1(doc, token, line, fileID):
  ack = False
  if(token.pos_=="ADJ" and
     token.dep_ == "xcomp" and
     token.head.pos_ == "VERB" and
     token.head.dep_ == "advcl"):
    ack = True
    addToTSV(1,line,fileID,doc,"Vdobj",token)
  return ack
```

Example 2. *Questa cosa funziona se il* ***messaggio*** *è* ***consegnato*** *intatto.*
This works if the ***message*** *is* ***delivered*** *intact.*

The function serves to identify and record a specific grammatical pattern in which an adjective is used as a modifier of a noun connected by an oblique complement. An oblique complement (oblique modifier or oblique object) is a syntactic constituent that provides accessory information, usually introduced by a preposition.

The rule shown in Listing 2 can recognise the word combination *buon allenatore* in Example 3.

Listing 2. Example of a function to recognize *amod*.

```
def checkFn2(doc, token, line, fileID):
  ack = False
  if(token.dep_ == "amod" and
     token.head.dep_ == "obl" and
     token.pos_ == "ADJ" and
     token.head.pos_ == "NOUN"):
    ack = True
    addDataTSV(2,line,fileID,doc,"Amod")
  return ack
```

Example 3. *L'aver goduto di un* ***buon allenatore*** *ha contribuito a migliorare la nostra squadra.*
Having a ***good coach*** *has helped to improve our team.*

We developed 20 distinct functions to identify syntactic patterns related to *amod* and *Vdobj*. These functions were consolidated into an array for streamlined processing. Each word in the dataset was sequentially evaluated against the functions in this array, as shown in Listing 3.

Listing 3. Example of the function scan for the words in a sentence.

```
functionsList = [
 checkFn1
 checkFn2
 ...
 checkFn20
]
negativeFunctionsList = [
 checkNegFn1
 checkNegFn2
]
for token in line:
  negFound = False
  found = False
  for fun in negativeFunctionsList:
    if fun(token):
      negFound = True
      break
  if negFound:
    continue
  for fun in functionsList:
    if fun(token):
      found = True
      break
  ...
```

The result was promptly recorded within our data structure for further analysis whenever a pattern match occurred.

From our tests, it was also necessary to develop negative controls, that is, rules that exclude certain word combinations that are not valid collocations. These rules were essential for filtering out false positives and ensuring that only valid collocations were retained in the final dataset. The developed code employs a combination of positive and negative rules to identify valid collocations, excluding invalid combinations. To optimise performance, the negative rules are executed before the positive rules, thereby avoiding unnecessary checks on word combinations that should be excluded.

An example of a negative rule is shown in Listing 4, which checks if the word is an ADJ with the morphological feature "NumType=Ord" (indicating it is an ordinal number) and returns false if this condition is met. This rule helps to filter out ordinal adjectives that do not contribute to valid collocations.

Listing 4. Example of a negative function to recognize *amod*.

```
    def checkNegFn1(doc, token, line, fileID):
      ack = False
      if(token.pos_=="ADJ" and "NumType=Ord" in token.morph):
        ack = True
      return ack
```

The rule shown in Listing 4 can recognise the word combination *primo capitolo* in Example 4.

Example 4. *Occorre leggere il* ***primo capitolo*** *del libro per l'esame di domani.*
It is necessary to read the ***first chapter*** *of the book for tomorrow's exam.*

The word combination *primo capitolo* is not an *amod* relation, but rather identifies or specifies which chapter within an ordered sequence and doesn't enhance the meaning of the noun *capitolo*.

To quantify the efficiency of the filtering process, Table 5 summarizes the impact of the implemented negative rules on the benchmark dataset. The analysis verifies the number of excluded candidates and confirms that no valid collocations (True Positives) were erroneously removed.

Table 5. Impact of the negative rules on the candidate extraction process (Benchmark dataset).

Rule ID	Description	Candidates Removed	TP Loss
checkNegFn1	Exclude Ordinal Adjectives	23	0
checkNegFn2	Exclude Possessive Adjectives	90	0

As shown, the rules effectively filtered out 113 non-collocational sequences without affecting the system's recall. This targeted filtering contributed to an overall accuracy improvement for the Hybrid approach (from 67.28% to 68.32% on the entire dataset), confirming the rules's precision in distinguishing noise from valid entries.

After this phase, we constructed a data structure containing all identified *amod* and *Vdobj* word combinations, ensuring no duplicates were present. This refined structure served as the input for the subsequent processing step.

The addToTSV function is used to create a TSV (Tab-Separated Values) file containing information about the discovered collocation, which allows the results to be printed in a format that is easily readable and analyzable. The output of the analysis of the Example 2 is shown in Table 6.

Table 6. Example of the output generated by the function *checkFn1* for the Example 2.

Form	W1	W2	Lemma1	Lemma2	Type	f	d
messaggio consegnato	messaggio	consegnato	messaggio	consegnare	*Vdobj*	1	1

The elements shown in Table 6 are:

***Form*:** the form of the collocation, i.e., the combination of words that form the collocation;

***W1*:** the first word that form the collocation;

***W2*:** the second word that form the collocation;

Lemma1 the first lemma of the two words that form the collocation;

***Lemma2*:** the second lemma of the two words that form the collocation;

***Type*:** the type of collocation, in this case *Vdobj*;

***f*:** the number of the function that identified the collocation, in this case the combination was identified by the function *checkFn1*, that is, the first function in the list of functions;

***d*:** the distance between the two words that form the collocation, in this case 1, since the two words are adjacent.

3.4.3. Validation of the Models

The performance of the three approaches, P-based, S-based, and Hybrid, was compared and evaluated using standard metrics: accuracy, precision, recall, and F1 score.

We also introduced the concept of the benchmark match, which measures the alignment between the model's predictions and the reference class labels from the benchmark dataset. This metric reflects the model's reliability and consistency when evaluated against a validation set. The formula used to calculate the benchmark match [32] is shown in

Equation (1), where TP = True Positive, TN = True Negative, and FN = False Negative, represent the model's performance components.

$$bm = 100 \times \frac{TP + TN}{TP + TN + FN} \quad (1)$$

To demonstrate the stability of our approach, we employed a non-parametric bootstrap resampling procedure, generating 1000 resampled datasets from the original benchmark by sampling with replacement. For each iteration, accuracy, precision, recall, and F1 score were recalculated to estimate their distribution. Consequently, the values reported in Tables 7–9 present the performance metrics alongside their Standard Deviation (SD), which provides a measure of the model's stability across different data compositions. Furthermore, the total candidate count (N) and the number of recognised collocations (Match) have been explicitly shown for each analysed relation to contextualise the statistical significance of the results.

Table 7. Analysis of the performance metrics for all three models in relation to modifier adjectives and verb-object collocations, 620 total entries.

	Match	Accuracy	Recall	Precision	F1 Score	Benchmark Match
S-based	520	67.79% ± 1.14%	86.38% ± 1.40%	75.91% ± 0.79%	80.81% ± 0.95%	86.38% ± 1.40%
P-based	475	70.37% ± 1.52%	78.90% ± 1.69%	86.68% ± 1.05%	82.61% ± 1.24%	78.90% ± 1.69%
Hybrid	548	68.32% ± 0.95%	91.03% ± 1.17%	73.26% ± 0.68%	81.19% ± 0.79%	91.03% ± 1.17%

Table 8. Evaluation of the performance metrics for the three models in relation to modifier adjectives (amod), 332 total entries.

	Match	Accuracy	Recall	Precision	F1 Score	Benchmark Match
S-based	298	69.14% ± 1.84%	89.76% ± 1.66%	75.06% ± 1.68%	81.76% ± 1.51%	89.76% ± 1.66%
P-based	277	76.51% ± 2.34%	83.43% ± 2.10%	90.23% ± 2.01%	86.70% ± 1.81%	83.43% ± 2.10%
Hybrid	315	70.63% ± 1.68%	94.88% ± 1.20%	73.43% ± 1.58%	82.79% ± 1.36%	94.88% ± 1.20%

Table 9. Evaluation of the performance metrics for the three models in relation to verb-object collocations (vdobj), 270 total entries.

	Match	Accuracy	Recall	Precision	F1 Score	Benchmark Match
S-based	222	66.07% ± 2.34%	82.22% ± 2.33%	77.08% ± 2.15%	79.57% ± 1.98%	82.22% ± 2.33%
P-based	198	63.25% ± 2.50%	73.33% ± 2.59%	82.16% ± 2.32%	77.50% ± 2.16%	73.33% ± 2.59%
Hybrid	233	65.44% ± 2.19%	86.30% ± 2.13%	73.04% ± 2.01%	79.12% ± 1.86%	86.30% ± 2.13%

The Hybrid approach consistently achieves superior benchmark match and recall results compared to the P- and S-based approaches. This trend holds across the entire dataset (as shown in Table 7) and within specific syntactic relations analyzed individually (Tables 8 and 9).

Notably, for the *amod* relation, the Hybrid method achieves a 94.88% benchmark match, a highly commendable outcome in candidate collocation identification.

From our results, the P-based approach demonstrates the highest precision but lower recall, indicating fewer false positives while missing more true positives. Conversely, the S-based approach exhibits the opposite trend, characterised by lower precision and higher recall. Precision measures the proportion of identified instances that are correct. Achieving a high precision value implies that the model classifies few items as "false positives." The P-based method excels in precision as it is much more restrictive and specific. It is based on preset queries that search for predefined Part-Of-Speech sequences. Recall is another statistical metric that measures a model's ability to find all relevant instances. This work calculates the number of "true collocations" detected in the analysed Corpus. High recall

means that the system "misses" a few instances by recognising the most significant number of examples. The S-based method achieves higher recall because it is based on dependency parsing, which analyses the sentence's grammatical structure.

Thanks to this flexibility, the S-based method can identify many potential collocations, including structurally complex ones that an approach based on fixed sequences would be unable to capture. This inevitably leads to a larger number of total candidates and, consequently, a higher probability of capturing almost all correct collocations, thus increasing recall. The disadvantage is that, being more "generous," it might include grammatically valid but not idiomatic pairs, thereby lowering precision.

All three approaches encounter challenges in detecting *Vdobj* relations, where word order and distance play a more disruptive role. Among these, the P-based method exhibits the most pronounced decrease in performance for *Vdobj* relations, with a benchmark match decline of 10.10% compared to the *amod* relation.

Figure 1 illustrates these findings by plotting benchmark match values across individual sample files of the Corpus; the discrepancies between approaches are most evident in two spoken texts with formal registers a conference and a political speech where the P-based method performs poorly. A subset of the Corpus was utilized because calculating accuracy statistics, such as Benchmark Match, required annotated data for comparative validation. The annotation was performed by expert linguists, which allowed us to validate our model.

In conclusion, the Hybrid approach consistently aligns more closely with the benchmark, demonstrating robustness and reliability in comparison to the gold standard of human annotation. These results confirm the advantage of integrating positional Part-Of-Speech and syntactic features for effective candidate collocation extraction.

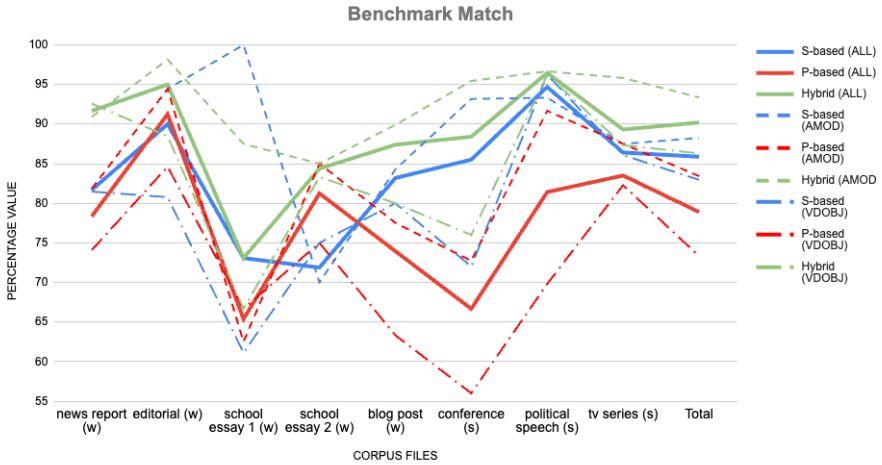


Figure 1. The graph shows the 'Benchmark Match' score analyzing a subset of the Corpus Files, where results are aggregated by individual source text (w = written, s = spoken). Colors are consistent for each method (S-based, P-based, Hybrid), while the curve style distinguishes the relation type: *amod* and *Vdobj* (continuous), *amod* only (dashed), and *Vdobj* only (dash-dot).

4. The Dictionary

The combinations obtained through the Hybrid approach were subsequently processed to be integrated within a MySQL database. Using a database enables the effective manipulation of data and the adoption of additional technologies [44], such as those necessary to display this content to users. Additional metrics have also been calculated that can be used to filter the dictionary according to predetermined thresholds. The threshold values are based on established practices in the field of collocation extraction as documented in the literature [45–47].

The metrics that are summarised below (Mi, Mi^3, $Logdice$, $Log-Likelihood$, D, DP, DP_{norm}, U) were calculated for each combination of the dictionary. The formulas used to calculate Mi, Mi^3, $LogDice$ and D are described in [45,48]. The formula of $Log-Likelihood$ is described in [49]. The formula of U is described in [50]. Finally, the formulas of DP and DP_{norm} are described in [51].

The formulas that are now described require the frequency associated with the combinations to be known and all the used formulas are summarised in Table 10.

The frequency of a combination was calculated, taking into account the context in which it was detected. This approach was adopted for the combination, understood as a combination of lemmas, as well as for the individual lemmas that comprise it. A combination of two lemmas will have a frequency calculated for both the first and the second lemma, and the value of the frequency of the two lemmas may differ.

The first time a combination is detected, its frequency will be 0. Subsequently, each time a previously analysed combination is re-encountered, the context of the sentence is analysed. If the sentences have the same context, the frequency is not increased; however, if the contexts of the sentences are different, the frequency of the combination increases by 1. This analysis produces a triple of values for each combination: the total frequency of the combination, the frequency of the first lemma, and the frequency of the second lemma. The analysis of whether a combination belongs to a context already encountered is based on the similarity between the sentences using the Bag-of-Words technique [52]. For each sentence, a list without repetitions of the words that compose the phrase is created and stored in memory. Each time a new combination is encountered, the sentence's word list is calculated and compared with the lists of previously analysed sentences in which the analysed combination was present. The similarity percentage between two sentences is calculated by dividing the number of common words, obtained from the intersection of the two lists, by the number of words in the shorter list, as described in [53,54].

Table 10. Summary of statistical metrics and mathematical definitions.

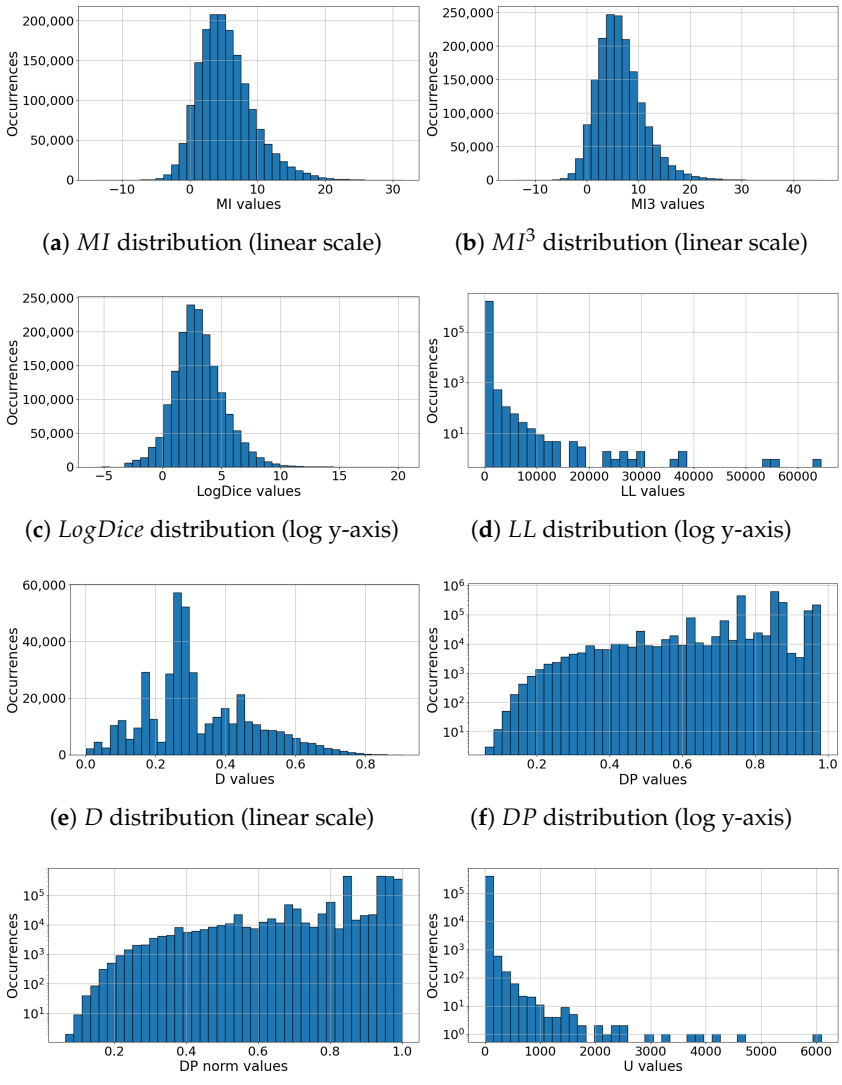
Metric	Formula	
$MI(A,B)$	$\log_2\left(\frac{f(A,B)N}{f(A)f(B)}\right)$	(2)
$MI^3(A,B)$	$\log_2\left(\frac{f(A,B)^3N}{f(A)f(B)}\right)$	(3)
$LogDice$	$14+\log_2\left(\frac{2f(A,B)}{f(A)+f(B)}\right)$	(4)
$Log-Likelihood$	$2\left(O_{11}\ln\frac{O_{11}}{E_{11}}+O_{21}\ln\frac{O_{21}}{E_{21}}+O_{12}\ln\frac{O_{12}}{E_{12}}+O_{22}\ln\frac{O_{22}}{E_{22}}\right)$	(5)
D	$1-\frac{V}{\sqrt{C-1}}=1-\frac{V}{3}$	(6)
DP	$1+\left(\frac{\sum p_i\cdot\log_2(p_i)}{\log_2(C)}\right)$	(7)
DP_{norm}	$\frac{DP}{1-\frac{1}{C}}=\frac{DP}{0.9}$	(8)
U	Df_{total}	(9)

If the similarity percentage exceeds a predetermined threshold, the combination is considered to belong to the same context. A similarity threshold of 0.5 was selected based on empirical tuning. Qualitative assessment by domain experts indicated that stricter thresholds led to data sparsity, negatively affecting the stability of association measures, while 0.5 effectively balanced context retention and noise filtering.

Figure 2 shows the graphs calculated by analysing the database of combinations extracted from the Corpus.

Mutual Information (Mi) is used to identify the strength of association between two words. Given a co-occurrence AB, A and B constituents of the combination, N number of words contained in the Corpus from which it was extracted and f absolute frequency value in that Corpus, the expression of Mi is shown in Equation (2). The graph of the distribution of Mi is shown in Figure 2a.

The majority of the analysed word pairs have a positive Mi score. The graph illustrates this by the histogram's asymmetrical distribution, characterised by a pronounced right tail (right-skewed). The peak of the distribution (the mode) is located at an Mi value of approximately 5. This is a highly favourable result, as it indicates that the extraction method has successfully identified word combinations that co-occur much more frequently than they would if selected by a random algorithm based on pure chance. The centring of the distribution in the positive range indicates that the dataset is rich in candidates with a significant associative link. Generally, an $Mi \geq 3$ is considered a good threshold for identifying significant collocations, though higher values are preferred for learner dictionaries.



(**a**) MI distribution (linear scale) (**b**) MI^3 distribution (linear scale)

(**c**) $LogDice$ distribution (log y-axis) (**d**) LL distribution (log y-axis)

(**e**) D distribution (linear scale) (**f**) DP distribution (log y-axis)

(**g**) DP_{norm} distribution (log y-axis) (**h**) U distribution (log y-axis)

Figure 2. Graphs showing the distribution of the calculated metrics for the dictionary entries. Histograms display the frequency distribution of candidates, grouped into 40 equal-width bins to visualize data density across the value range.

The Cubic Association Ratio (Mi^3) aims to amplify terms involving less frequent associations to determine whether rare associations may be significant. The formula is shown in Equation (3). The graph of the distribution of Mi^3 is shown in Figure 2b.

The distribution of Mi^3 values maintains the general shape observed for Mi. The most notable difference is the extension of the value scale on the x-axis, which now reaches and exceeds 40. Cubing the joint frequency $f(A, B)$ acts as a magnifying glass, "stretching" the distribution to the right. The pairs that already exhibited a strong connection with the standard Mi show an exponential increase in their Mi^3 score, distancing themselves from weaker associations. The primary purpose of this metric is to identify word pairs that, although rare, constitute very strong and fundamental collocations to study for language acquisition. For lexicographic purposes, a threshold of $Mi^3 \geq 12$ is suggested to prioritize frequent and strongly associated pairs over rare combinations.

LogDice, a statistical metric, was calculated to indicate the typicality of the combination using the formula shown in Equation (4).

The graph of the distribution of *LogDice* is shown in Figure 2c.

LogDice focuses on the degree of internal cohesion of a word combination. If the calculated value is high, the presence of one word in the pair strongly indicates the other's presence within the sentence or discourse. Combinations with values above 10 are considered highly cohesive, and students can greatly benefit from studying them to learn the Italian language. The combination with the highest detected *LogDice* value is *capro espiatorio (scapegoat)* with a value of 13.742, followed by *anidride carbonica (carbon dioxide)* with a value of 13.712, and *procreazione assistita (assisted reproduction)* with a value of 13.515. For practical lexicographic filtering, we suggest prioritizing candidates with a $LogDice \geq 10$, as this threshold typically indicates high cohesion and idiomaticity.

$Log - Likelihood$ (LL) was calculated to determine the likelihood of a combination, i.e., how likely it is that two words recur together. LL compares a combination's observed frequency with the expected frequency if the two words were independent. The formula used is shown in Equation (5). The graph of the distribution of LL is shown in Figure 2d. A critical feature of this graph is that the y-axis (Occurrences) is plotted on a logarithmic scale. A logarithmic y-axis is necessary to visualise the data, indicating a vast difference in the number of occurrences across the range of LL values. The distribution is skewed to the right, with most candidate pairs having low LL scores.

Where O_{11} is the observed frequency of the combination, O_{21} is the frequency of word B from which the value of the frequency of combination AB is subtracted, O_{12} is the observed frequency of word A from which the value of the frequency of combination AB is subtracted, O_{22} is the total words minus the observed frequencies of the combinations. E_{11}, E_{12}, E_{21}, E_{22} are the expected frequencies.

LL's primary function is to measure confidence or statistical significance. LL values can be used to filter the data in the dataset. A combination with a very low value, close to 0, is highly likely to be of little use and could be classified as noise. Discarding these combinations can enhance the quality of the dictionary by focusing on those that are more likely to be beneficial for students and removing those that are likely to be false positives or noise. The filter can be applied by setting a minimum threshold for the LL value, which semantically describes the desired confidence level. We suggest filtering for $LL \geq 50$.

We also calculated Juilland's dispersion index D, which refers to how widely a word or combination is distributed in a Corpus. Let us imagine the Corpus structured in several sub-corpora: a word/combination is dispersed within the Corpus if it occurs with equal or similar frequency in all parts of the Corpus; conversely, a word or combination is poorly dispersed if it only happens in one or a few parts of the Corpus. Dispersion applies equally to single words and word combinations. A word combination can be seen as a single word.

D is calculated as shown in Equation (6), where V is the coefficient of variation and C is the number of parts of the Corpus, which in our case is 10. The graph of the distribution of D is shown in Figure 2e.

Unlike the previous metrics, the distribution of the D index is not unimodal. It presents a complex, bimodal (or even multimodal) shape, which provides significant insights into the nature of the collocations found in the Corpus. The most prominent feature is the presence of at least two distinct peaks. This suggests that the extracted collocations fall into different categories based on usage patterns across the Corpus. While fewer in number, the collocations with high dispersion scores ($D > 0.7$) are of the highest value for this project. These are word combinations whose meaning is consistent regardless of the topic of discussion. Examples of combinations with a D value greater than 0.7 are *prendere decisione (make a decision)*, *avere potere (have power)*, *mostrare interesse (show interest)*, and *fare parte (be part of)*. We suggest filtering for $D \geq 0.6$.

Gries' Deviation of Proportions (DP) coefficient was calculated to indicate how widely a word combination is distributed in a Corpus. The formula used is shown in Equation (7), where p_i is the proportion of occurrences of the element in the i-th part of the Corpus and C is the total number of parts of the Corpus. The graph of the distribution of DP is shown in Figure 2f on a logarithmic scale.

A DP score near 1 indicates maximum clustering, while a score near 0 indicates maximum dispersion. These are highly domain-specific collocations, found in only one or a few parts of the Corpus. Several distinct and sharp peaks (e.g., around 0.75, 0.85, 0.95) suggest specific, recurring clustering patterns. This could be related to the size and content of the 10 sub-corpora, indicating that large amounts of topic-specific vocabulary dominate certain sections of the source material. When filtering the combinations, it is recommended to select those with values below a threshold (e.g., 0.5). The collocations with the lowest DP values are the most fundamental for a learner. For example, among those with the smallest values are *valere pena (be worth)*, *spiegare meglio (explain better)*, and *avere senso (make sense)*.

The normalised Deviation of Proportions coefficient (DP_{norm}) was calculated to indicate how widely a word combination is distributed in a Corpus. The formula used is shown in Equation (8). The graph of the distribution of DP_{norm} is shown in Figure 2g.

DP_{norm} is a scaled version of DP, making it easier to interpret and compare across different datasets. The purpose of this normalisation is to adjust the metric's range so that it measures the same concept as the un-normalised DP. The data shown in the figure presents results similar to those of DP. Similar to DP, a threshold of $DP_{norm} \leq 0.5$ is recommended.

Finally, the Usage Coefficient index (U) [50] was calculated according to the formula shown in Equation (9). The graph of the distribution of U is shown in Figure 2h on a logarithmic scale.

U is a composite metric that combines the dispersion index D that describes how widely and evenly a collocation is used across the different parts of the Corpus, with the total frequency of the collocation f_{total}, which indicates how often the collocation appears in the Corpus. A high U score can only be achieved by a frequent and widely distributed collocation in the Corpus. This distribution shape proves that the overwhelming majority of candidate collocations have a very low Usage Coefficient. This is an expected result because the previous analysis of dispersion metrics D showed that many items were poorly dispersed, which would inherently lead to a low U score. The most important information is in the sparse tail to the right of the distribution. The collocations that achieve a high U score are the best candidates for the dictionary, as they are frequently used and appear in various contexts. To ensure candidates are both frequent and widely used, we recommend setting a minimum threshold of $U \geq 10$.

The graphs illustrate the relative distribution of calculated values. The X-axis represents the values of the variable, while the Y-axis represents the frequency for each corresponding X-axis value. The graphs provide a visual representation of the distribution of the calculated metrics, offering insights into the characteristics of the dictionary entries.

The analysed metrics, for which values have been calculated, are fundamental for identifying the collocations representing high-yield learning for the student.

These metrics are utilized during the filtering phase and are combined to select statistically relevant collocations. The filtering process was implemented by establishing thresholds derived from a rigorous analysis of prior studies on the English language, which guided the application of these filters to Italian collocations.

5. The System Architecture

The dictionary's system architecture is built on a cloud infrastructure, adopting a pay-per-use model. This approach eliminates the need for physical servers by utilising the provider's virtual servers, with costs determined by actual service consumption. It does not require physical hardware maintenance, resulting in consequent advantages in terms of service reliability for our use case. The infrastructure is structured into two distinct subnets: the first one is for development and testing, while the second one is dedicated to the production environment that serves end-users.

Figure 3 shows the overall architecture of the system.

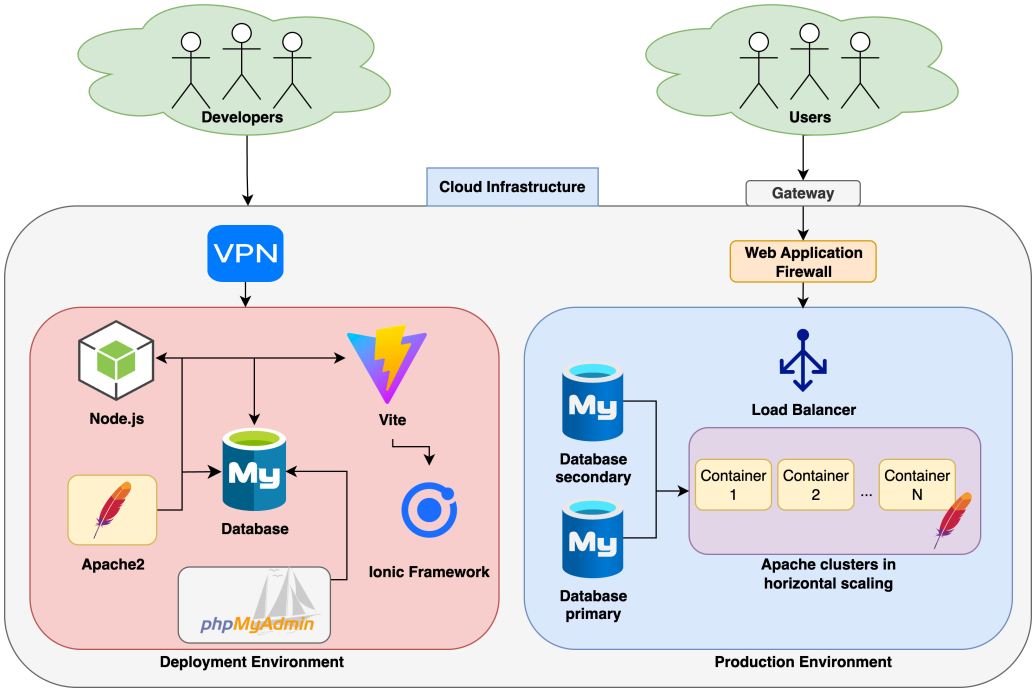


Figure 3. Dictionary infrastructure architecture.

The dictionary development environment is now analysed. The purpose of this area of the cloud infrastructure is to enable developers to manage content used by users without slowing down or causing issues for the public during testing and implementation of new code functions. Access to this portion of the infrastructure is via a VPN connection, ensuring that only developers can connect to the machines and services in this area. Various Docker containers have been configured within it. The decision to use Docker was made due to software maintenance, system package updates, and the intrinsic security of the containers themselves, which, for example, are only able to communicate with the outside world if the developers decide to do so and only to the ports and IP addresses agreed

during the development phase. A first container contains a Node.js server that establishes communication with a Vite service, within which the Ionic framework has been set up. This setup is necessary to create the dictionary using the Hybrid App technique. The Vite server is capable of serving HTML, CSS, and JavaScript code. It automatically updates the content displayed on developers' test devices as soon as the source code is changed, in a fully automated manner.

Development takes place using the Ionic framework, which allows the creation of a web page that can then be exported for an Android or iOS project, drastically reducing the development and programming time for mobile operating systems, as two separate projects are automatically generated from a single code base, which can be immediately compiled and sent to the official stores (Google Play Store and App Store), ensuring that the graphical interface also maintains the styles of the respective operating systems to which they belong. This framework and the `Vite` server are supported by `Node.js`, which enables JavaScript code to be executed on the server side and allows packages and libraries to be managed via `npm`.

A second container contains a MySQL server that hosts the dictionary database. This database is used only for the development and testing phase. However, it can be synchronised with the environment actually used by users. The database can also be managed via a graphical interface using a second container containing `phpMyAdmin`. Queries to the dictionary are handled using PHP hosted within an Apache2 container.

The production environment, on the other hand, is used solely by end users. The access point is the public Gateway, which corresponds to the dictionary's domain name.

Immediately after contacting the Gateway, requests pass through a WAF, i.e., a Web Application Firewall designed to filter out cyberattacks, such as SQL injection, Cross-site scripting (XSS), and distributed denial-of-service (DDoS) attacks.

If the request is deemed legitimate, it is redirected to the load balancer of the Apache2 node cluster. The load balancer sorts user requests using the Round-Robin algorithm, evenly distributing the number of requests across the various nodes. Apache2 containers are used to manage queries to the database. Each container exposes APIs written in PHP that query the database and return the results to the client.

The container cluster was designed to ensure that if one of these containers malfunctions, the others can still execute requests. They also enable the distribution of computational load, making the system more responsive. Metrics have also been implemented to analyse the number of requests made by users in real time. Based on the calculated value, nodes are added or removed to ensure the best possible experience. This technique is also called auto-scaling and is one of the key features of cloud computing.

For example, suppose there is a spike in users using the dictionary. In that case, the system can react automatically and create new Apache2 containers to handle the additional workload, releasing these resources as soon as the number of requests decreases. This ensures that the system is always responsive and that there are no slowdowns or service interruptions, paying only for the resources actually used.

The database is also built using High Availability techniques. There are two databases: a primary database, which can both read and write data, and a secondary database, which has read-only permissions. The secondary database is used to distribute the workload and improve system performance. This ensures that even if the primary database fails, the system can continue to function using the secondary database.

The infrastructure is based on a production database cluster accessed by end-users, and a second database dedicated to application development and maintenance. These databases can be synchronized to propagate changes from the development environment to the production one. The primary advantage of this approach is the ability to test and

implement updates and modifications within an isolated and secure testing environment. This prevents any potential disruption to the users accessing the service. Once modifications have been tested and deemed reliable, they can be made available to the public.

6. Conclusions and Future Work

This work aims to create a dictionary for international students learning Italian. Unlike conventional dictionaries, the one we aim to create will feature word combinations, also called collocations. The primary characteristic of these collocations is that their semantic meaning differs from the syntactic meaning derived from analysing the individual words composing them. This characteristic represents one of the significant obstacles for language learners, as a proper grasp of collocations requires a deep understanding of the ongoing discourse and the idiomatic expressions of the target language. Consequently, having suitable tools to enhance study and improve the quality of learning is crucial. In this work, we analysed a Corpus, a collection of texts, comprising 41.7 million words, using three distinct methodologies.

The project's initial phase involved testing our approach on a small sample of the Corpus to validate our working methodology. The results indicate that the Hybrid method outperforms the other two in metrics such as benchmark match and recall, supporting our initial assumptions. Nevertheless, the P-based approach outperforms in precision, accuracy, and F1 score, highlighting areas where the Hybrid method requires further refinement. Following encouraging results, we proceeded with the study by implementing and refining grammatical analysis rules. These rules, based on the output from the *spaCy* NLP library, enabled the creation of Python scripts for automated Corpus analysis, producing a dataset of collocations. Enhancing the model's precision could involve developing additional Python rules, such as negative rules, to systematically eliminate false positives. While post-tagging provides a degree of robustness that compensates for the relative imprecision of syntactic parsing, the detection rules used after parsing require additional refinement to mitigate errors stemming from false positives. The findings underscore the value of a Hybrid approach, demonstrating that it leads to improved performance and higher-quality results.

At the end of the Corpus analysis and combination extraction phase, we obtained a total of 2,097,595 combinations, of which 40.05% were identified using the S-based method, 17.60% with the P-based method and 42.35% with the Hybrid method. After developing the dataset, we conducted a detailed analysis to gather statistical metrics, as outlined in Section 4. This analysis aimed to filter collocations in the dictionary, eliminate potential false positives. Additionally, we sought to identify the collocations that are most frequently used in the Italian language. The labeling of collocations according to CEFR levels (A1, A2, B1, B2, C1, C2) is currently in progress and will be the subject of future work. In the future, we will utilise the adopted metrics to enhance the quality of the dictionary, which will be accessed through a web interface. This will allow students to select their current proficiency level and the target level they wish to achieve, based on frameworks such as the Common European Framework of Reference for Languages. The collocations can also be ordered by combining frequency with the measure of dispersion through the different textual genres represented in the Corpus. Thanks to these features, our goal is to enable students and teachers to use this dictionary with the ultimate aim of improving the quality of Italian language study and learning.

Author Contributions: Conceptualization, D.P., O.G., S.S., I.F. and F.Z.; Data Curation, D.P., O.G., S.S., I.F. and F.Z.; Formal Analysis, D.P., O.G., S.S., I.F. and F.Z.; Funding acquisition, D.P., O.G., S.S., I.F. and F.Z.; Investigation, D.P., O.G., S.S., I.F. and F.Z.; Methodology, D.P., O.G., S.S., I.F. and F.Z.; Project administration, O.G. and S.S.; Resources, O.G. and S.S.; Software, D.P., O.G. and S.T.; Supervision,

O.G. and S.S.; Validation, D.P., O.G., S.T., S.S., I.F., F.Z. and L.F.; Visualization, D.P., O.G., S.T., S.S., I.F., F.Z. and L.F.; Writing—original draft, D.P., O.G., S.T., S.S., I.F., F.Z. and L.F.; Writing—review and editing, D.P., O.G., I.F. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financed by the Italian Ministry of University and Research under the PRIN 2022 project "Dici-A: Dictionary of Italian Collocations for Learners". Project code: 2022HXZR5E, CUP: J53D23008060006.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The source code and specific extraction scripts are not publicly available due to intellectual property restrictions associated with the funded project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
Amod	Adjective Modifier
API	Application Programming Interface
CEFR	Common European Framework of Reference for Languages
CQP	Corpus Query Processor
D	Juilland's Dispersion Index
DDoS	Distributed Denial-of-Service
DP	Gries' Deviation of Proportions
FN	False Negative
GPU	Graphics Processing Unit
ISDT	Italian Stanford Dependency Treebank
L2	Second Language
LL	Log-Likelihood
LLM	Large Language Model
LogDice	Log-Dice Coefficient
Mi	Mutual Information
MWE	Multi-Word Expression
NLP	Natural Language Processing
POS	Part-Of-Speech
SD	Standard Deviation
SQL	Structured Query Language
TN	True Negative
TP	True Positive
U	Usage Coefficient Index
UD	Universal Dependencies
Vdobj	Verb-Direct Object
VPN	Virtual Private Network
WAF	Web Application Firewall
XSS	Cross-Site Scripting

References

1. Venkatapathy, S.; Joshi, A.K. Relative Compositionality of Multi-word Expressions: A Study of Verb-Noun (V-N) Collocations. In *Natural Language Processing—IJCNLP 2005*; Dale, R., Wong, K.F., Su, J., Kwong, O.Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 553–564.
2. Paquot, M. Lexicographyand phraseology. In *The Cambridge Handbook of English Corpus Linguistics*; Biber, D., Reppen, R., Eds.; Cambridge Handbooks in Language and Linguistics; Cambridge University Press: Cambridge, UK, 2015; pp. 460–477.
3. Spina, S. The role of Learner Corpus Research in the study of L2 phraseology: Main contributions and future directions. *Rivista Psicolinguistica Appl.—J. Appl. Psycholinguist.* **2020**, *XX*, 35–52.

4. Benson, M.; Benson, E.; Ilson, R. *The BBI Dictionary of English Word Combinations*; Benjamins: Amsterdam, The Netherlands, 1986.
5. McIntosh, C.; Francis, B.; Poole, R. *Oxford Collocations Dictionary for Students of English*; Oxford University Press: Oxford, UK, 2002.
6. Rundell, M. *Macmillan Collocations Dictionary for Learners of English*; Macmillan Education: London, UK, 2010.
7. Bandyopadhyay, S.; Naskar, S.K.; Ekbal, A. *Emerging Applications of Natural Language Processing: Concepts and New Research*; IGI Global Scientific Publishing: Hershey, PA, USA, 2013; pp. 1–388. [CrossRef]
8. Urzì, F. *Dizionario delle Combinazioni Lessicali*; Convivium: Luxemburg, 2009.
9. Tiberii, P. *Dizionario Delle Collocazioni*; Zanichelli: Bologna, Italy, 2012.
10. Lo Cascio, V. *Dizionario Combinatorio Italiano*; Benjamins: Amsterdam, The Netherlands, 2013.
11. Hanks, P. Corpus evidence and Electronic Lexicography. In *Electronic Lexicography*; Granger, S., Paquot, M., Eds.; Oxford University Press: Oxford, UK, 2012; pp. 57–82.
12. Seretan, V. *Syntax-Based Collocation Extraction*; Springer: Dordrecht, The Netherlands, 2011.
13. dos Reis, E.C.; Caneppa, S.; Vasconcelos, P.; de Lima Santos, P.C.J. Advancing pharmacogenomics research: Automated extraction of insights from PubMed using SpaCy NLP framework. *Pharmacogenomics* **2024**, *25*, 573–578. [CrossRef] [PubMed]
14. Straka, M.; Hajič, J.; Straková, J. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., et al., Eds.; European Language Resources Association: Paris, France, 2016; pp. 4290–4297.
15. Castagnoli, S.; Lebani, G.E.; Lenci, A.; Masini, F.; Nissim, M.; Passaro, L.C. POS-patterns or Syntax? Comparing methods for extracting Word Combinations. In *Computerised and Corpus-Based Approaches to Phraseology: Monolingual and Multilingual Perspectives*; Pastor, G.C., Ed.; Tradulex: Geneva, Switzerland, 2016; pp. 116–128.
16. Wu, H.; Zhou, M. Synonymous collocation extraction using translation information. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2003), Sapporo, Japan, 7–12 July 2003; pp. 120–127.
17. Lin, D. Automatic identification of non-compositional phrases. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, Morristown, NJ, USA, 20–26 June 1999; pp. 317–324. [CrossRef]
18. Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; Manning, C.D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Online, 5–10 July 2020; Celikyilmaz, A., Wen, T.H., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 101–108. [CrossRef]
19. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual String Embeddings for Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; Bender, E.M., Derczynski, L., Isabelle, P., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 1638–1649.
20. Constant, M.; Eryiğit, G.; Monti, J.; van der Plas, L.; Ramisch, C.; Rosner, M.; Todirascu, A. Survey: Multiword Expression Processing: A Survey. *Comput. Linguist.* **2017**, *43*, 837–892. [CrossRef]
21. Smadja, F. Retrieving collocations from text: Xtract. *Comput. Linguist.* **1993**, *19*, 143–177.
22. Choueka, Y. Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. In Proceedings of the User-Oriented Content-Based Text and Image Handling, Paris, France, 21–24 March 1988; pp. 609–623.
23. Evert, S. The Statistics of Word Cooccurrences: Word Pairs and Collocations. Ph.D. Thesis, University of Stuttgart, Stuttgart, Germany, 2004.
24. Krenn, B. Collocation mining: Exploiting corpora for collocation idenfication and representation. In Proceedings of the KONVENS 2000, Ilmenau, Germany, 9–12 October 2000.
25. Breidt, E. Extraction of V-N-collocations from text corpora: A feasibility study for German. In Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives, Columbus, OH, USA, 22 June 1993; pp. 74–83. [CrossRef]
26. Ritz, J. Collocation Extraction: Needs, Feeds and Results of an Extraction System for German. In Proceedings of the Workshop on Multi-Word-Expressions in a Multilingual Context at the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3–7 April 2006; pp. 41–48.
27. Su, K.Y.; Wu, M.W.; Chang, J.S. A corpus-based approach to automatic compound extraction. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM, USA, 27–30 June 1994; pp. 242–247.
28. Lü, Y.; Zhou, M. Collocation Translation Acquisition Using Monolingual Corpora. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004. [CrossRef]
29. Orliac, B.; Dillinger, M. Collocation extraction for machine translation. In Proceedings of the Machine Translation Summit IX: Papers, New Orleans, LA, USA, 18–22 September 2003.

30. Simkó, K.I.; Kovács, V.; Vincze, V. USzeged: Identifying Verbal Multiword Expressions with POS Tagging and Parsing Techniques. In Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), Valencia, Spain, 4 April 2017; Markantonatou, S., Ramisch, C., Savary, A., Vincze, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 48–53. [CrossRef]
31. Shi, T.; Lee, L. Extracting Headless MWEs from Dependency Parse Trees: Parsing, Tagging, and Joint Modeling Approaches. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 8780–8794. [CrossRef]
32. Perri, D.; Fioravanti, I.; Gervasi, O.; Spina, S. Combining Grammatical and Relational Approaches. A Hybrid Method for the Identification of Candidate Collocations from Corpora. In Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies, MWE-UD at LREC-COLING 2024, Torino, Italy, 25 May 2024; Bhatia, A., Bouma, G., Doğruöz, A.S., Evang, K., Garcia, M., Giouli, V., Han, L., Nivre, J., Rademaker, A., Eds.; ELRA and ICCL: Paris, France; New York, NY, USA, 2024; pp. 138–146.
33. Vorvilas, G.; Pantazi, D.; Paxinou, E.; Feretzakis, G.; Kalles, D.; Kameas, A.; Karousos, N.; Verykios, V.S. An Automated Text Summarization Approach for Open-ended Responses in Student Online Surveys. In Proceedings of the 2024 15th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 17–19 July 2024; pp. 1–7. [CrossRef]
34. Labini, P.S.; Cianfriglia, M.; Perri, D.; Gervasi, O.; Fursin, G.; Lokhmotov, A.; Nugteren, C.; Carpentieri, B.; Zollo, F.; Vella, F. On the Anatomy of Predictive Models for Accelerating GPU Convolution Kernels and beyond. *ACM Trans. Archit. Code Optim.* **2021**, *18*, 1–24. [CrossRef]
35. Perri, D.; Simonetti, M.; Gervasi, O. Deploying Serious Games for Cognitive Rehabilitation. *Computers* **2022**, *11*. [CrossRef]
36. Perri, D.; Gervasi, O. A Novel Computational Framework for Visual Snow Syndrome. *IEEE Access* **2025**, *13*, 23877–23887. [CrossRef]
37. Spina, S.; Zanda, F.; Fioravanti, I. FROM PEC TO PEC24: A NEW REFERENCE CORPUS FOR ITALIAN. *Ital. Linguadue* **2025**, *17*, 745–768. [CrossRef] [PubMed]
38. Spina, S. Learner corpus research and phraseology in Italian as a second language: The case of the DICI-A, a learner dictionary of Italian collocations. In *Collocations Cross-Linguistically. Corpora, Dictionaries and Language Teaching*; Vilas, B.S., Ed.; Memoires de la Societe Neophilologique de Helsinki: Helsinki, Finland, 2016; pp. 219–244.
39. de Marneffe, M.C.; Manning, C.D.; Nivre, J.; Zeman, D. Universal Dependencies. *Comput. Linguist.* **2021**, *47*, 255–308. [CrossRef]
40. Schmid, H. Probabilistic part-of-speech tagging using decision trees. In *New Methods in Language Processing*; Routledge: London, UK, 2013; pp. 154–164.
41. Spina, S. Il Perugia Corpus: Una risorsa di riferimento per l'italiano. Composizione, annotazione e valutazione. In Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014, Pisa, Italy, 9–10 December 2014; Basili, R., Lenci, A., Magnini, B., Eds.; Pisa University Press: Pisa, Italy, 2014; Volume 1, pp. 354–359.
42. Hardie, A. CQPweb combining power, flexibility and usability in a corpus analysis tool. *Int. J. Corpus Linguist.* **2012**, *17*, 380–409. [CrossRef]
43. Attardi, G.; Saletti, S.; Simi, M. Evolution of Italian Treebank and Dependency Parsing towards Universal Dependencies. In Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015, Trento, Italy, 3–4 December 2015.
44. Perri, D.; Gervasi, O.; Tasso, S.; Fioravanti, I.; Zanda, F.; Spina, S.; Forti, L. Dockerized Architecture for a Progressive Web App: An Italian Collocations Dictionary. In Proceedings of the Computational Science and Its Applications, ICCSA 2025 Workshops, Istanbul, Turkey, 30 June–3 July 2025; pp. 33–43.
45. Gablasova, D.; Brezina, V.; McEnery, T. Collocations in Corpus-Based Language Learning Research: Identifying, Comparing, and Interpreting the Evidence. *Lang. Learn.* **2017**, *67*, 155–179. [CrossRef]
46. Evert, S.; Uhrig, P.; Bartsch, S.; Proisl, T. E-VIEW-affilation—A large-scale evaluation study of association measures for collocation identification. In Proceedings of the Electronic Lexicography in the 21st Century. Proceedings of the eLex 2017 Conference, Brno, Czech Republic, 27–29 June 2017; pp. 531–549.
47. Deng, Y.; Liu, D. A multi-dimensional comparison of the effectiveness and efficiency of association measures in collocation extraction. *Int. J. Corpus Linguist.* **2022**, *27*, 191–219. [CrossRef]
48. Rychlỳ, P. A Lexicographer-Friendly Association Score. In Proceedings of the RASLAN, Karlova Studanka, Czech Republic, 5–7 December 2008; pp. 6–9.
49. Pojanapunya, P.; Todd, R.W. Log-likelihood and odds ratio: Keyness statistics for different purposes of keyword analysis. *Corpus Linguist. Linguist. Theory* **2018**, *14*, 133–167. [CrossRef]
50. Juilland, A.; Brodin, D.; Davidovitch, C. *Frequency Dictionary of French Words*; Romance Languages and Their Structures; Mouton: Hawthorne, NY, USA, 1971.
51. Gries, S.T. Dispersions and adjusted frequencies in corpora. *Int. J. Corpus Linguist.* **2008**, *13*, 403–437. [CrossRef]

52. Juluru, K.; Shih, H.H.; Keshava Murthy, K.N.; Elnajjar, P. Bag-of-Words Technique in Natural Language Processing: A Primer for Radiologists. *RadioGraphics* **2021**, *41*, 1420–1426. [CrossRef] [PubMed]
53. McGill, M.; Koll, M.; Noreault, T. *An Evaluation of Factors Affecting Document Ranking by Information Retrieval Systems*; Research Report to National Science Foundation; School of Information Studies, Syracuse University: Syracuse, NY, USA, 1979. Available online: https://eric.ed.gov/?id=ED188587 (accessed on 11 June 2025).
54. Vijaymeena, M.; Kavitha, K. A Survey on Similarity Measures in Text Mining. *Mach. Learn. Appl. Int. J.* **2016**, *3*, 19–28. [CrossRef]