T-Extractor: A Hybrid Unsupervised Approach for Term and Named Entity Extraction Using Rules, Statistical, and Semantic Methods

Aliya Kalykulova*, Aliya Nugumanova
"Big Data and Blockchain Technologies" Science and Innovation Center, Astana IT University,
Astana, Kazakhstan
E-mail: aliyakalykulova01@gmail.com, a.nugumanova@astanait.edu.kz
* Corresponding author

Keywords: automatic term extraction, unsupervised annotator, T-Extractor, phrase extraction, semantic analysis

Received: January 27, 2025

Automatic term extraction is a key technology for optimizing natural language processing tasks such as machine translation, sentiment analysis, knowledge graph construction, and/or ontology population. This study presents the T-Extractor approach for unsupervised term extraction. The research goal is to develop an efficient method that does not require labeled data, and to analyze its applicability on scientific texts. T-Extractor combines rule-based, statistical, and semantic analysis, treating unigram and phrase extraction as two subtasks. Part-of-speech templates are used in the candidate selection phase, while a filter based on raw and rectified frequencies refines phrase boundaries. TopicScore is applied for final term filtering, improving extraction precision. Additionally, simple rules help identify abbreviations and named entities, improving recall. T-Extractor was tested on the ACTER (three languages, four domains) and ACL RD-TEC 2.0 datasets. In English, the best result was achieved in the equi domain, with an F1-measure of 48.5%, precision of 41.6%, and recall of 58.2%. On the ACTER dataset, the approach outperformed existing unsupervised methods and performed better than the supervised GPT-3.5-Turbo and BERT models in the corp and wind domains. Specifically, in the corp domain, T-Extractor's F1-measure approached that of the HAMLET model, lagging by 3.7%. In addition, the method showed results comparable to those of promtATE and TALN-LS2N.

Povzetek: T-Extractor je hibridna, nenadzorovana metoda za izvleček izrazov in imen, ki z združitvijo pravil, statistik in semantike presega tudi nadzorovane modele na določenih domenah.

1 Introduction

Knowledge-driven digital products are the cornerstone of Industry 4.0, and the most popular format for representing knowledge is knowledge graphs. Developing large knowledge graphs manually is an expensive process, for which natural language processing techniques, including automatic terminology extraction methods, have proven useful in speeding up and scaling. In this paper, we focus on applying an unsupervised approach for automatic term extraction, and we intend to show that with the proper strategy, they can be competitive even when training data is scarce. In 2018, Gartner included knowledge graphs in its famous "Hype Cycle for Emerging Technologies" and 4 years later, industry leaders such as Siemens [1], Bosch [2], and Mitsubishi Electric [3] have proven first-hand that knowledge graphs have successfully moved from the realm of hype to the real economy and even reached a productivity plateau in some cases.

Today, there is a renewed interest in knowledge graphs among researchers due to the human-centered challenges of Industry 5.0 and the growing realization that artificial intelligence alone, without human input, cannot be the basis for building robust systems [4-6]. Expectations related to knowledge graphs center mainly around the idea of combining them with machine/deep learning models to produce better and more explainable cognitive solutions and serve as groundwork for the next generation of systems based on human-machine synergy [6].

These expectations, in turn, stimulate research in the field of computational terminology, a science at the intersection of knowledge engineering and natural language processing dealing with automatic term extraction. After all, it is terms, as expressors of domain concepts, that are the basic building blocks for knowledge graphs. However, even without additional motivation, computational terminology is currently at an important transition stage due to the appearance of transformers, and this stage is no less significant in terms of expected results than the one caused by the arrival of statistical methods in the industry in the 1990s [7].

According to [8], the growth rate of new published articles continues to increase. This makes manual text processing almost impossible, which necessitates a transition to digital data processing and knowledge graph construction for more efficient information analysis. The construction of knowledge graphs requires the selection of informative text units and the establishment of links between them [9]. Terms, named entities, and other text elements are considered as such units. In addition, term extraction plays a key role in machine translation tasks,

search engines, text abstracting and other applications [10, 11]. Some studies also suggest using key phrases to improve information retrieval [12] or text classification [13], illustrating the broader significance of phrase extraction across various domains. Since terminology often includes multi-word expressions, phrase extraction is closely related to term extraction, as both aim to identify meaningful linguistic units that structure domain knowledge. This study focuses on the extraction of terms and named entities, including multi-word terms, which naturally intersects with phrase extraction methods.

Existing approaches for automatic term extraction can be roughly categorized into supervised and unsupervised approaches [10]. Supervised methods provide high accuracy but require significant amounts of labeled data for training, which makes them difficult to adapt to new domains and reduces their effectiveness when processing texts from different subject areas. Unsupervised methods, on the other hand, have greater versatility but show less accuracy when recognizing terms and named entities.

Term extraction faces several challenges, including the presence of noise in the data, difficulty in identifying the boundaries of multi-word expressions, and polysemy. The accuracy of extraction also depends on the quality of the models used, such as embedding generation models and algorithms for part-of-speech detection. Therefore, the development of methods capable of solving textual data analysis problems while considering the existing limitations of tools and resources is still ongoing.

The research aims to develop a more efficient unsupervised approach for extracting terms and named entities. This paper presents an unsupervised annotator, T-Extractor, which extracts terms and named entities using rules, statistical and semantic analysis. The proposed annotator demonstrated an average F1-measure of 40% on the ACTER and ACL RD-TEC 2.0 datasets.

The following research questions were posed:

1. What is the impact of combining statistical, rule-based, and semantic techniques in term extraction?

2. How does the proposed T-Extractor compare to existing unsupervised and supervised methods in extracting terms across multiple domains and languages?

The paper is structured as follows: first, an overview of existing methods and approaches is presented, which allows us to determine the current state of research in this area. Then, the datasets used for annotator quality assessment are presented.

Next, the principles and algorithms of the proposed approach are detailed, including key aspects of its implementation. Then, the developed methodology for quality assessment of the proposed approach is described, which ensures the objectivity and reproducibility of the obtained results.

We focus on analyzing the results, discussing them, and identifying the strengths and weaknesses of the method. The paper is concluded with conclusions and recommendations aimed at practical application of the developed annotator in various text processing tasks.

2 Related work

A term, according to a common definition, is a word or phrase used to refer to a specific concept, subject or phenomenon in a particular field of knowledge. As [14] points out, terms are a valuable linguistic resource that contributes to linguistic coherence. With the development of digital resources and natural language processing (NLP) tools, terms have gained a key role in knowledge graph creation, electronic document management, and data analysis. This has greatly expanded their use beyond traditional tasks such as translation [15]. Terms play a crucial role in structuring and categorizing information, becoming the basis for organizing data in various information systems.

According to common standards, terms are classified into simple (single word) and complex (multi-word) terms [16, 17]. Studies by [17] show that 99% of technical multi-word terms take the form of noun phrases (NPs), whose key element is a noun or its grammatical equivalent.

Named Entities (NEs) are real-world entities that can be identified, such as people, places, organizations, dates, or products. According to the MUC-7 classification, Named Entities are categorized as follows: persons, organizations, locations, dates, times, monetary amounts, and percentages.

Thus, terms and named entities are key components in representing and analyzing information, making them an important research subject in natural language processing and data management. Table 1 presents a comparative analysis of existing methods. Next, an overview of unsupervised and supervised approaches to term extraction is given.

2.1 Unsupervised approaches

Unsupervised information extraction methods are based on rules, frequency features, semantics or their hybrid combinations [18, 19]. The standard unsupervised

Table 1: Summary of related work

Method name	**Data sets**	**F1 score, %**	**Limitations**
Unsupervised			
NMF [20]	ACTER	corp_en: 25,7 equi_en: 33,3 wind_en: 26,1 htfl_en: 33,7	Does not consider word semantics. The method relies on statistical metrics based on word frequency, making it vulnerable to noise in the data. It is also sensitive to parameter settings, including the number of topics, the number of terms to extract, and term length.
UA [21]	ACL-RD TEC 2.0., GENIA, ScienceIE	ACL: 49,95 GENIA: 45,65 ScienceIE: 39,7	Extracts only noun phrases and does not always correctly determine term boundaries, leading to term splitting or merging with irrelevant parts. No semantic filtering for unigrams.
UA1 [22]	ACTER, ACL-RD TEC 2.0.	corp_en: 24,3 equi_en: 28,9 wind_en: 29,5 htfl_en: 32,7 ACL: 44,8	A reimplementation of the UA approach, achieving an F1 score 5.15% lower than the original UA method on the ACL dataset.
Supervised			
HAMLET [27]	ACTER	corp_en: 43,8 equi_en: 60,1 wind_en: 50,1 htfl_en: 55,4	Requires a large number of features. The method computes 152 features per candidate, making training more complex and computationally expensive. It also depends on the quality of training data and has limited adaptability to new domains.
TALN-LS2N [28]	ACTER	htfl_en: 46,66 htfl_fr: 48,15	Sensitive to the n-gram parameter, limiting term length to 4-grams for English and 5-grams for French, which makes extracting longer terms difficult. Also, it requires a large amount of labeled data for training.
GPT-3.5-Turbo [29]	ACTER	corp_en: 31,4 equi_en: 49,7 wind_en: 32,5 htfl_en: 55,6	Token limit constraints reduce the amount of text available for analysis. In broad subject areas (e.g., renewable energy), the method may include irrelevant terms.
promptATE [30]	ACTER	htfl_en: 51,4 htfl_fr: 47,8 htfl_nl: 55,4	Trained on general data and does not account for domain specificity, leading to over-extraction of terms (high recall but low precision).

approach algorithm includes the following steps: 1) candidate extraction, 2) ranking based on certain features, and 3) selection of top candidates using a threshold [20].

Article [20] proposed an unsupervised annotator based on matrix decomposition using the Non-Negative Matrix Factorization (NMF) method. This approach was tested in the TermEval 2020 competition, where it achieved an average F1 score of 27.2% on the ACTER dataset. The NMF method demonstrates its versatility as it can be adapted to handle different languages and subject areas.

Another term extraction approach, called Unsupervised Annotator (UA), is proposed by [21]. It is based on the use of part-of-speech rules, morphological analysis, and two metrics, Topic Score and Specific Score, computed based on cosine similarity of contextual embeddings. The model was tested on the ACL, GENIA and ScienceIE datasets, achieving an average performance of 45.11% on the F1 metric.

The Unsupervised Annotator (UA) approach from [21] was reimplemented by another research group and tested on the ACTER dataset, as the original model code was not publicly available [22]. The implemented UA1 annotator achieved F1=44.8% on the ACL dataset, which is 5.15% lower than the result of the original model (F1=50%). UA1 achieved an average F1=28.9% on the ACTER (English) dataset. The lower performance is attributed to the high variability of part-of-speech combinations in terms of the ACTER dataset. Additionally, multi-word expressions could overlap with

true terms, indicating an incorrect definition of phrase boundaries. Despite this, overall, the UA1 annotator shows high performance when applied to the ACL RD-TEC 2.0 corpus.

Term and keyword extraction share similar objectives, as a term can function as a keyword. The main task of terminology is to generate conceptual descriptions, whereas keywords are intended to reflect the content of the text [23]. Nevertheless, similar techniques can be used to extract informative words.

An example of an unsupervised approach to keyword extraction is the YAKE! model [24]. This method uses different frequency metrics, considers the position of the word in the text as well as its case. The authors note that relevant words are more likely to occur at the beginning of the text or headings, and capitalized words can be significant.

A review of keyword extraction methods presented in [23] emphasizes that most key phrases are noun groups (noun phrases), making their extraction an important step to improve accuracy.

In [13], an unsupervised Subword-Phrase extraction method based on frequency analysis is proposed to improve text classification. The approach demonstrated that a supervised classification model achieves better results when using phrases as one of its features. This confirms the importance of leveraging lexical units that convey the main topic of the text. Since terms are more specific, their application may be even more effective for classification.

Most unsupervised approaches rely on frequency-based features. According to [25], the distributions of noise and quality phrases have similar patterns, making it difficult to extract relevant terms. One of the key challenges is the correct definition of phrase boundaries. Additionally, extracting rare terms is challenging due to their low frequency, which makes it difficult to establish accurate boundaries.

2.2 Supervised approaches

Supervised approaches to term extraction significantly outperform unsupervised approaches [26]. This is due to the complexity of textual data processing, where factors such as case, context, parts of speech, and special punctuation marks influence term extraction. Accounting for all these aspects simultaneously is challenging, as exceptions may exist for each factor. However, machine learning can address this problem comprehensively, making it an effective tool in this field.

One such approach, HAMLET, utilizes over 160 features, including statistical, variational, linguistic, and contextual features, for model training [27]. A random forest-based algorithm demonstrated the best performance in this approach.

The TALN-LS2N approach, presented in [28], was trained on both true terms and false examples. After training, additional filtering is applied: candidate terms starting with conjunctions and pronouns are excluded, and duplicate or common words are removed.

[29] presents an approach using the GPT-3.5-Turbo model with few-shot scripts. To extract terms, a prompt is generated that contains instructions (e.g., “find a term”), a sentence to analyze, and an example of terms. This method stands out for its simple implementation and minimal reliance on labeled data.

Another state-of-the-art approach, promptATE [30], is also based on the use of prompts for term extraction. This approach uses two models, ChatGPT (gpt-3.5-turbo) and Llama 2-Chat, and implements three result output formats. The sequence-labeling approach achieves high precision but low recall. The text-extractive response format uses partial markup with skips, which allows more terms to be extracted but reduces precision. In turn, text generative response, which uses labeled cues, provides an optimal balance between precision and recall.

Traditionally, term extraction was treated as a binary classification task (term/non-term), requiring a large amount of labeled data. However, a new approach has emerged, leveraging prompt-based methods for term generation. This approach requires significantly less labeled data, making it a promising direction for further research.

3 Dataset

The effectiveness of the proposed approach was evaluated using the ACTER and ACL RD-TEC 2.0 corpora, both of which contain texts with labeled terms.

The ACTER (Annotated Corpora for Term Extraction Research) corpus consists of manually annotated texts spanning four topic areas: corruption (corp), training (equi), heart failure (htfl), and wind energy (wind). This corpus contains texts in three languages: English (en), French (fr), and Dutch (nl) [19]. In this study, extracted term candidates were compared against a reference list of true terms, which includes named entities.

The ACL Reference Dataset for Terminology Extraction and Classification (ACL RD-TEC 2.0) corpus, released in 2016, contains annotated abstracts of scientific articles in computational linguistics. A distinctive feature of this corpus is its double annotation by two independent annotators [31], which reduces potential bias and enhances the accuracy of method evaluation.

4 Approach description

T-Extractor is an unsupervised tool for extracting terms and named entities through rule-based, frequency-based, and semantic analysis. At the initial stage, rule-based and frequency analysis help identify potential candidates, while semantic analysis further filters them, selecting the most relevant and domain-specific units.

Extracting multi-word expressions is more challenging than extracting single-word terms. This is due to several challenges, including term boundary definition, nested terms, and syntactic variations across languages. Therefore, term extraction is performed in two stages: unigram extraction and phrase extraction. Abbreviations and named entities exhibit distinct features that facilitate their identification in text (e.g., capitalization patterns).

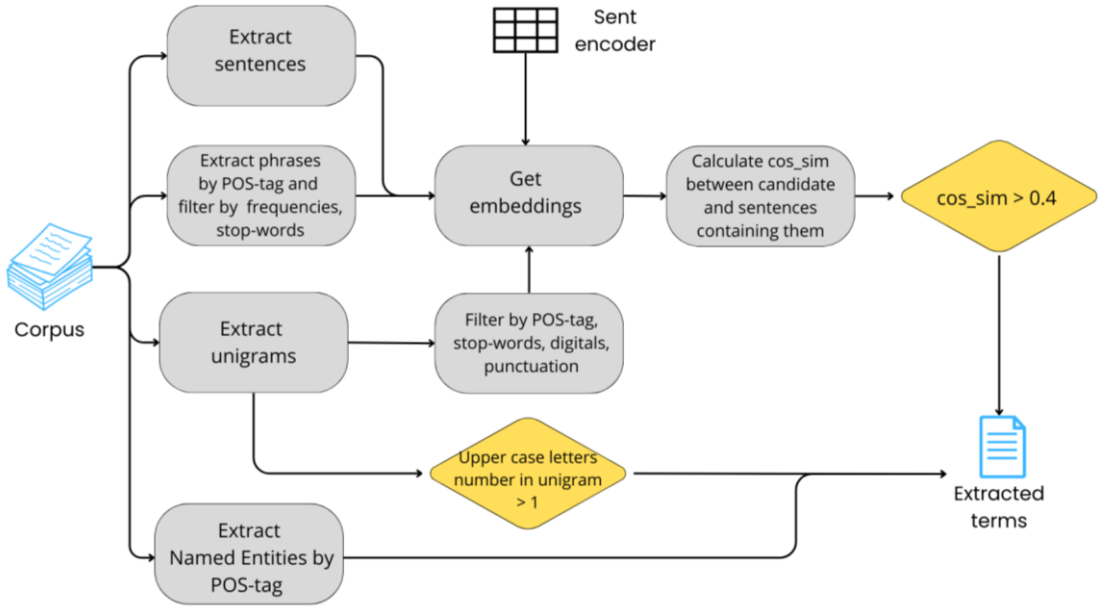

Figure 1: Generalized algorithm for extracting terms, named entities and abbreviations. The input of the algorithm is a text or a corpus of texts, and the output is a final list of extracted information units.

Simple rules were applied for their extraction. While extracting unigrams and phrases, abbreviations and named entities can also be extracted. The tool does not classify extracted information but instead consolidates key units, such as terms, named entities, and abbreviations, into a single list.

Unigram extraction involves selecting candidates based on part-of-speech tagging and pre-filtering them. Multi-word expression extraction relies on part-of-speech patterns combined with filtering based on two frequency metrics. Low-frequency candidates are filtered using a phrase-grouping approach based on word position matching, followed by selecting the most frequent candidate. This approach mitigates ambiguity in term boundaries by selecting the most likely candidates.

The extracted unigrams and multi-word expressions are filtered using the Topic Score metric proposed in [21]. This metric allows selecting candidates that are most relevant to the thematic area of the analyzed text.

Figures 1 and 2 illustrate the term extraction process. A detailed algorithm description is also provided in these figures. A detailed description of the algorithm's key steps is provided in the following section. The first section covers data preparation and model setup for term extraction. The second section describes the unigram extraction algorithm. The third section explains the phrase extraction technique. The fourth section details semantic filtering of extracted candidates using Topic Score. The fifth section introduces an abbreviation extraction approach. The sixth section describes named entity extraction techniques. Finally, the seventh section discusses the models used and threshold tuning parameters.

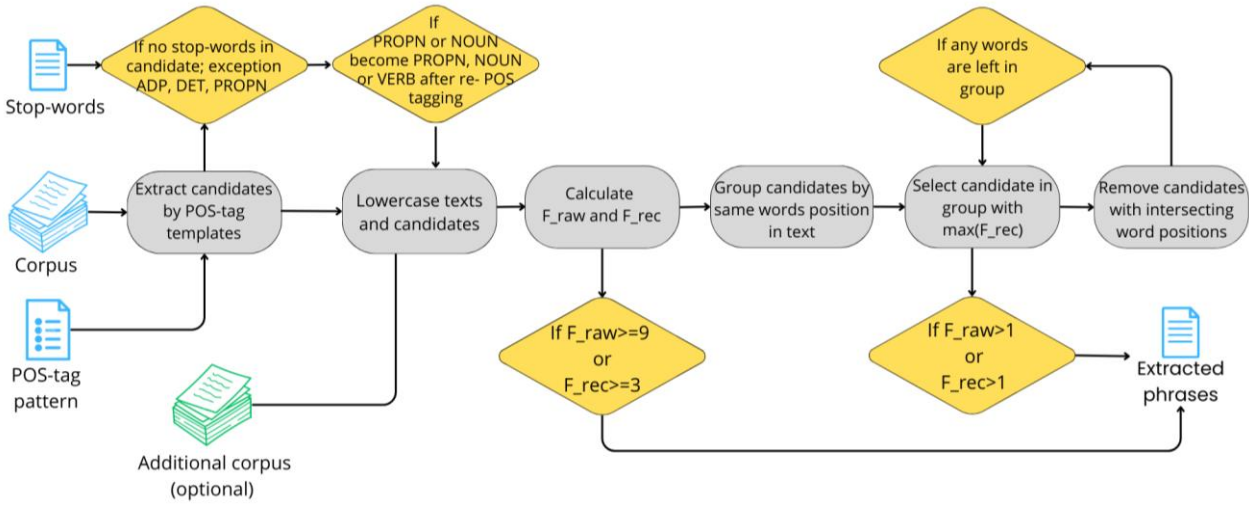

Figure 2: Phrase term extraction algorithm.

Table 2: Patterns of part-of-speech combinations for extracting multi-word expressions.

Templates for English		
1) N*	5) N*, ADJ*, N*	9) N, ADP, N
2) ADJ*, N*	6) ADJ, VERB, N*	10) K*, ADP, N*
3) ADJ*	7) VERB*, N*	11) M (if there is a hyphen)
4) VERB, ADJ, N*	8) ADV*, ADJ*	
N - [PROPN, NOUN]	K - [ADJ, PROPN, NOUN]	M - [VERB, ADV, X]
Templates for French		
1) K*	4) NOUN, VERB	7) N*, ADP, N*, ADP, N*
2) N*, ADP, N*	5) VERB, ADJ	8) N*, ADP, N*, ADP, N*, ADP, N*
3) N*, ADP, DET, N*	6) ADJ, VERB	
N - [NOUN, ADJ]	K - [NOUN, ADJ, PROPN]	
Templates for Dutch		
1) K*	3) NOUN, ADP, NOUN, ADP, NOUN	5) VERB, ADP, DET, NOUN
2) N*, ADP, M*	4) VERB, ADP, NOUN	6) VERB,NOUN
N - [NOUN, ADJ]	K - [NOUN, ADJ, PROPN, SYM]	M - [NOUN, ADJ, CCONJ]

* - asterisk marks those part-of-speech that can be one or more in a sequence [25].

N, K, M - means that their position can be any part of speech from the specified list

If *N**, *K** or *M**, then there can be one or more of any part-of-speech from the list.
Example N* (for English):

- NOUN, NOUN, NOUN
- NOUN, PROPN,
- PROPN,PROPN, NOUN

4.1 Data and model preparation

For text tokenization and part-of-speech partitioning, the SpaCy model was used, which was preconfigured so that hyphenated words were not separated into separate tokens. This setting avoided additional complexity in developing part-of-speech patterns for phrase extraction and prevented the extraction of incomplete words. For example, the word *"anti-corruption"* in this case should be treated as a whole, since the component *"anti"* clearly refers to *"corruption"* and cannot be treated as an independent term. This processing minimizes noise in the data by ensuring that hyphenated words are extracted in their integral form.

In the unigram extraction stage, words with a hyphen are extracted as single tokens. However, in the performance evaluation presented in Table 8, this category is classified as multi-word expressions because it consists of two tokens joined by a hyphen.

Before part-of-speech tagging, the text is not converted to lowercase, which allows the SpaCy model to identify proper names (*PROPN*) more accurately. After extracting candidates based on part-of-speech tagging, the extracted units are converted to lower case. This step is necessary to perform subsequent tasks correctly and to obtain a more accurate evaluation of the extracted terms.

4.2 Unigram extraction

Extraction of candidate unigrams is performed based on part-of-speech partitioning. Nouns (*NOUN*), proper names (*PROPN*) and adjectives (*ADJ*) are considered as candidates. To reduce noise, prefiltering removes stop words and unigrams consisting only of digits and/or punctuation marks. Additionally, words containing punctuation marks, except hyphen ("-") and apostrophe (" ' "), as long as they appear within a word rather than at its boundary, are filtered out.

4.3 Extracting multi-word expressions

The phrase extraction algorithm presented in Figure 2 includes five main steps: (1) candidate extraction using part-of-speech templates, (2) pre-filtering of candidates, (3) raw and rectified frequency calculation, (4) phrase grouping by position followed by extraction of the most frequent phrases, and (5) phrase extraction based on thresholds.

4.3.1 Candidate extraction using part-of-speech templates

The templates used in this approach are given in Table 2. Phrases are selected as candidates if they are longer than one word or if they contain a hyphen. All phrases containing punctuation marks except for the hyphen and the apostrophe are excluded.

Special symbols were used to optimize the selection of part-of-speech patterns and avoid unnecessary combination of variants. The asterisk (*) indicates that a given Part-of-Speech can be followed by the same Part-of-Speech. This approach was adapted from [25] to create flexible templates.

Table 3: Comparative analysis of multi-word term and NE candidate extraction using different sets of part-of-speech patterns in the corp(en) domain from ACTER.

POS-tag patterns for phrases	N	P	R	F1
SpaCy Noun chunks	5950	12,5	48,88	19,91
Ngrams (from 2 to 5 tokens), with pre-filtering from punctuation and digits	92817	1,2	73,39	2,37
[25]	7354	14,65	70,76	24,27
[32]	3398	14,18	69,25	23,55
[33]	2722	14,36	56,18	22,88
Proposed templates	13404	9,94	87,52	17,85

Letters *N, K, M* represent sets of parts of speech. In a template, any part of speech from the corresponding list can be specified at their position. For example, in the *ADJ+NOUN* and *ADJ+PROPN* templates, the position of *NOUN* can be *PROPN*. Thus, the *ADJ+N* template allows extracting both *ADJ+PROPN* and *ADJ+NOUN*.

If the characters *N, K, M* are followed by an asterisk sign, the part-of-speech sequence may contain any part-of-speech from the specified list. For example, the formula *N** can result in combinations such as *PROPN+PROPN*, *NOUN+NOUN+NOUN*, *NOUN+PROPN+PROPN*, etc.

The part-of-speech templates were computed and selected based on the analysis of terms from the ACTER dataset. This template format achieves an average recall of approximately 88% for English, 75% for French, and 75% for Dutch when extracting phrase terms from the ACTER dataset. This approach provides flexibility to create different template variations, making it significantly easier to capture all possible term combinations.

Table 3 presents the results of multi-word term and named entity extraction using part-of-speech patterns in the corp(en) domain of the ACTER dataset. The proposed approach is compared with Noun Chunks (a SpaCy method designed to extract noun phrases), n-gram extraction (ranging in size from 2 to 5 tokens with pre-filtering) and three sets of part-of-speech templates described in [25, 32, 33].

The Noun Chunks method is limited to the extraction of noun phrases, resulting in a low recall. The extraction of n-grams exhibits high recall, but the precision remains extremely low. Moreover, increasing the length of n-grams results in even lower precision since longer terms are less frequent.

The parts-of-speech templates from [25, 32, 33] provide higher precision and F1-measures compared to the proposed templates. However, the templates proposed in this paper show the highest recall value in extracting phrase terms. The main goal at this stage is to maximize the coverage of potential candidates, since subsequent filtering may lead to the removal of terms themselves, negatively impacting recall.

Table 4 presents the evaluation of the recall of phrase term extraction for each part-of-speech template. The analysis of the results shows that some patterns in some subject areas almost fail to identify true terms. This may indicate that the structure of phrasal terms depends not only on the general patterns of morphosyntactic design, but also on the subject specificity of texts.

This finding highlights the need to adapt patterns to specific domains or employ dynamic term extraction methods that consider context and semantic characteristics of the subject domain.

4.3.2 Preliminary filtration

The extracted phrases undergo a prefiltering stage, which includes two main steps: cleaning by adjusting POS tags and removing stop words.

Table 4: Recall results of phrase candidate extraction for each part-of-speech template. The number corresponds to the part-of-speech template number from Table 2.

Data set	Lang	Domain	№ POS-tag template											Total Recall
			1	2	3	4	5	6	7	8	9	10	11	
ACL-RD-TEC 2.0	**en**	**annotator 1**	38,14	35,7	0,75	0,75	1,13	0,19	3,83	0,06	1	1,25	0,13	**82,93**
		annotator 2	39,38	37,82	0,69	0,5	1,15	0,14	3,4	0	0,69	0,96	0,14	**84,87**
ACTER	**en**	**corp**	34,91	35,78	1,15	0,14	0,29	0,14	1,01	0,14	8,76	4,17	0,29	**86,78**
		equi	52,82	23,66	1,88	0	0,13	0,13	5,65	0	2,15	0,94	0,54	**87,9**
		wind	53,67	26,69	2,18	0,4	0,6	0,3	4,46	0	1,49	0,99	0,2	**90,98**
		htfl	33,51	43,76	1,97	1,25	2,76	0,2	2,04	0	0,39	1,18	0,46	**87,52**
	fr	**corp**	51,04	18,99	2,37	1,04	0	0	0,89	0	-	-	-	**74,33**
		equi	45,13	15,71	3,38	2,39	0,4	0,2	0,6	0	-	-	-	**67,81**
		wind	44,55	23,48	1,85	2,4	0,18	0,74	1,85	0,37	-	-	-	**75,42**
		htfl	69,2	8,2	0,7	2,27	0,44	0	0,7	0	-	-	-	**81,51**
	nl	**corp**	59,54	14,64	1,54	0,77	0,39	3,28	-	-	-	-	-	**80,16**
		equi	57,54	0,25	0	1,01	0,75	4,27	-	-	-	-	-	**63,82**
		wind	69,16	4,34	0	0	0	2,41	-	-	-	-	-	**75,91**
		htfl	78,13	0,79	0	0,13	0	2,24	-	-	-	-	-	**81,29**

Cleanup by adjusting POS tags. This step removes phrases ending in *PROPN* (proper nouns) or *NOUN* (noun) if re-tagging changes their part of speech to something other than *NOUN*, *PROPN*, or *VERB*. The SpaCy model considers the context in which a word occurs when marking it up, which may cause its tag to change. For example, the phrase "European Central" may be initially tagged as *PROPN* + *PROPN*, but after re-evaluation, it could be reclassified *PROPN* + *ADJ*. Since the templates in Table 2 do not provide for a combination ending in *ADJ*, such a phrase is considered incomplete and is deleted. Repeated tagging helps identify incomplete expressions and reduce the number of candidates. This filter is applied to English texts only.

Filtering by stop-words. This step removes phrases containing stop words unless the stop word is *PROPN* (proper name), *ADP* (preposition), or *DET* (article). Although the patterns in Table 2 allow prepositions and particles in phrases, such elements may occur only in the middle of a phrase (when the *ADP* is not at the beginning or end of the phrase). As for *PROPN* proper nouns, some names may contain common words. For example, in *"New York"*, the word *"New"* is a stop-word, but it represents part of the city name and should not be removed. This filter helps eliminate phrases containing common words, such as *"other illegal activities"*, *"possible cases"*, *"more effective"*, which reduces the noise in the data.

This filtering helps to remove certain candidate categories, which minimizes the noise in the data and improves the accuracy of term extraction.

4.3.3 Raw and rectified frequency calculation

It is assumed that if neighboring words frequently co-occur in the text, there is a high probability that they form a stable collocation. For a more accurate analysis, two types of frequency are calculated: raw and rectified, as described in [25].

Raw frequency (*F_raw*) represents the total count of a phrase's occurrences in the text. This frequency indicates how often a given combination of words occurs in the source text.

Rectified frequency (*F_rec*) represents how often the target phrase appears in the text, excluding instances where it is part of longer phrase. To compute the rectified frequency, one must consider the sum of the rectified frequencies (*F_l*) of longer phrases that contain the target phrase. The rectified frequency is computed using the following formula:

$$F\_rec = F\_raw - F\_l \quad (1)$$

At this stage, phrases are sorted in descending order by length, as longer phrases containing the target phrase must be considered to compute the rectified frequency. To enhance the accuracy of rectified frequency calculations, an additional corpus of texts related to the target domain can be used, facilitating a more precise detection of true phrase boundaries.

The phrase extraction process is divided into two stages. In the first stage, phrases with the highest frequencies are selected based on thresholds: raw frequency (*F_raw*) greater than 9 or rectified frequency (*F_rec*) greater than 3. These thresholds were optimized via experimental tuning to achieve optimal results In the second step, the extracted phrases are grouped by word position, and additional filtering is performed based on frequency comparison. These steps are necessary to identify the most strongly related words that form stable phrases.

4.3.4 Grouping phrases by common word positions

Phrases are grouped based on the presence of common word positions. In the grouping process, it is possible that phrases that do not directly share common positions can be grouped together if there is an intermediate phrase that shares common positions with two other phrases.

An example of such grouping is shown in Table 5, where one of the groups contains phrases with overlapping word positions. For example, the candidate phrase *"Austrian-led network"* does not share word positions with the phrase *"European partners against corruption"* but overlaps with the phrase *"Network European*

Table 5: Example of a group of candidate phrases containing words with common positions. The underlined candidate will be categorized as a phrase because it has the highest rectified and raw frequency values. The overlapping positions of words with the underlined candidate are shown in bold.

Candidates	***F_raw***	***F_rec***	**Word position index**
austrian-led, network	1	0	19534,19535
european, partners	3	0	**19536,19537**
austrian-led	1	0	19534
network, european, partners	1	0	19535,**19536,19537**
partners, against, corruption	3	0	**19537,19538,19539**
austrian-led, network, european, partners, against, corruption	1	1	19534,19535,**19536,19537,19538,19539**
network, european, partners, against, corruption	1	0	19535,**19536,19537,19538,19539**
<u>european, partners, against, corruption</u>	<u>3</u>	<u>2</u>	<u>19536,19537,19538,19539</u>

partners" in terms of word positions. As a result, phrases that do not directly share word positions can still belong to the same group.

This step is necessary to remove incomplete or partial phrases. In each group, the phrases with the highest rectified frequency (or raw frequency if the rectified frequency is 1 or 0) are selected. If the rectified or raw frequency of a phrase is strictly greater than 1, such a phrase is accepted. The phrase, as well as all candidates in the group that share common word positions with the accepted phrase, are then removed from the group. The process is repeated until no phrases remain in the group.

In the example shown in Table 5, all candidates that have common word positions with the candidate *"European partners against corruption"* are removed from the group. After that, two candidates, *"Austrian-led network"* and *"Austrian-led"*, remain in the group, but their frequencies do not exceed 1, so they are also removed. It is important to note that if a candidate is removed from one group, it will not be removed from other groups.

This approach helps minimize the number of candidates and highlights the most coherent and meaningful phrases.

In general, the phrase extraction method from text includes several interrelated steps, each of which contributes to improving the accuracy and quality of the extraction of relevant phrases. In the first stage, part-of-speech templates are applied for initial filtering and extraction of a set of potential candidate phrases to cover a wide range of possible terms. In the prefiltering stage, less informative phrases are eliminated, which helps to reduce the number of candidates and improve precision. Using a frequency-based filter, incomplete phrases are removed, and their boundaries are accurately identified, which helps eliminate noisy data and improve the quality of the remaining candidates.

4.4 TopicScore filter

The TopicScore metric presented in [21] is used for the semantic filtering of extracted unigrams and multi-word expressions. Unlike the original approach, where it was applied exclusively to phrases, in this paper, TopicScore is used for both multi-word expressions and unigrams.

The metric is defined as the cosine similarity between the candidate embedding (*w_c*) and the sentence embedding (*w_sent*) in which it occurs. The higher the similarity value, the greater the probability that the candidate is a term relevant to the given context. In this study, a candidate is classified as a term if the cosine similarity exceeds a threshold value of 0.4.

$$TopicScore = \frac{w\_c * w\_sent}{\|w\_c\| * \|w\_sent\|} \quad (2)$$

Embeddings are computed using the BERT model, which generates context vectors for sentences. The use of context embeddings avoids the out-of-vocabulary (OOV) problem characteristic of static vector-based methods.

The TopicScore metric facilitates the selection of terms that are most relevant to the subject domain and also allows the identification of the most informative candidate terms.

4.5 Abbreviation extraction

For abbreviation extraction from text, it is important to preserve the original case. As a basic strategy for abbreviation extraction, the rule that unigrams containing two or more uppercase letters are treated as abbreviations has been used. However, this method has some limitations, as it is vulnerable to cases where a word is entirely in uppercase, such as in headings or sections of text. An example of such a case is the word *"ABSTRACT"*, which would be misidentified as an abbreviation under this rule.

The keyword extraction method presented by the authors of [24] considers character case as one of the features used to identify significant keywords in the text. In addition, some names may contain multiple capital letters within a single word (e.g., *"YouTube"*), allowing the approach to detect not only acronyms but also named entities.

4.6 Extracting named entities

To extract named entities, POS tagging is applied to the text while preserving the original case. Named entity extraction is performed using the *PROPN** and *[PROPN, ADP]** patterns. This approach accounts for the possibility that prepositions may occur in the names of many entities, but only if they appear in the middle of the sequence. This allows the selection of sequences that correspond to typical named entity structures, preserving their integrity.

Unlike phrase extraction methods, which cover all possible word sequences, this approach focuses on extracting complete and continuous word sequences labeled with the *PROPN* tag. This allows for more accurate identification of text fragments such as names of organizations, geographical entities, and other named categories. Thus, the method focuses on identifying structures that represent complete named entities, helping to improve extraction accuracy.

Abbreviations and named entities are neither semantically nor statistically filtered. This is because the frequency of such elements may be too low, and their semantic meaning may not be sufficiently unambiguous. For example, abbreviations are often sequences of letters that may represent long names and be perceived as random character strings. In the case of named entities, such as people's names, they may be associated with a variety of activities, making their contextual meaning less specific and more universal. As a result, such entities may appear in different domains and may not always be clearly associated with a specific topic or field.

4.7 Fine-tuning approach

There are several key factors to consider when extracting multiword expressions. First, the choice of part-of-speech combination patterns is important because it determines

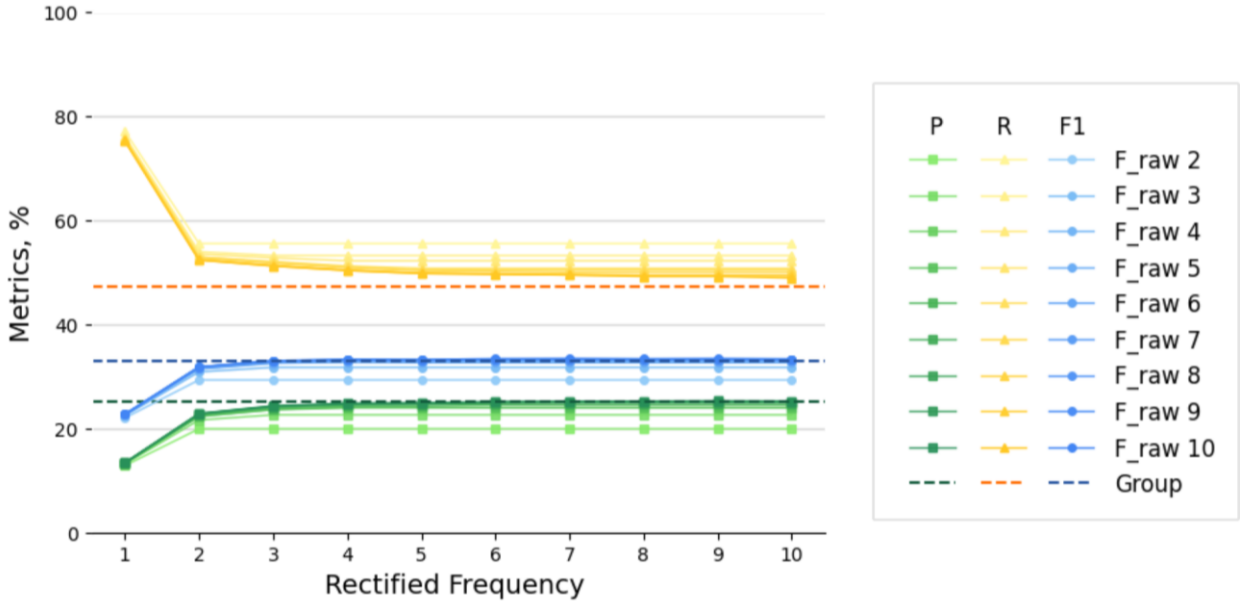

Figure 3: Effect of raw and rectified frequency thresholds on Precision(P), Recall (R) and F1 score on corp (en) domain from ACTER. The dotted line Group indicates phrase retrieval rates only at the filtering stage with grouping by common word positions, excluding phrases.

the recall and the number of extracted terms. Too many candidates may increase the computational burden of subsequent processing steps, including filtering and analysis.

Second, setting thresholds for raw and rectified frequencies is an important aspect. Lowering these thresholds may reduce the accuracy of term extraction, while increasing them may reduce recall. The definition of thresholds also depends on the size of the corpus and the texts. For the ACTER corpus, thresholds have been set raw frequency above 9 and rectified frequency above 3. If the corpus or texts are too small, it is recommended to lower the frequency thresholds. As in the case of the ACL RD-TE 2.0 enclosure, where the thresholds were set: raw frequency above 2 and rectified frequency above 1.

Figure 3 shows the variation of the indicators depending on the frequency thresholds. The evaluation of the indicators was performed considering the phrases extracted in the grouping phase based on common word positions. Thus, the phrases obtained by grouping and the phrases extracted based on frequency thresholds were compared with the list of true multiword terms.

Table 6: Recall (R, %) of extracted phrase terms when using grouping and the effect of frequency thresholds. The Group column contains the recall values obtained at the filtering stage by grouping phrases by common word positions. The Frequency thresholds column presents the gain in recall due to additional phrase extraction using frequency thresholds.

Data Set	lang	domain	Group		Frequency thresholds		Total recall for Phrases
			N	R	N	R	
ACTER	en	corp	1300	47,13	+169	+4,02	51,15
		equi	995	44,22	+51	+1,88	46,1
		wind	1741	43,25	+309	+7,74	50,99
		htfl	1615	31,6	+138	+2,76	34,36
	fr	corp	1367	41,39	+142	+1,34	42,73
		equi	729	35,98	+18	+0,2	36,18
		wind	1290	42,33	+188	+4,81	47,14
		htfl	876	25,13	+68	+2,18	27,31
	nl	corp	1134	42,58	+53	+0,39	42,97
		equi	648	32,91	+16	+1,26	34,17
		wind	1045	33,01	+45	+2,89	35,9
		htfl	948	31,09	+31	+0,53	31,62
ACL-RD- TEC 2.0	en	anntator1	752	22,58	+2534	+40,65	63,23
		anntator2	1041	23,36	+3686	+43,55	66,91

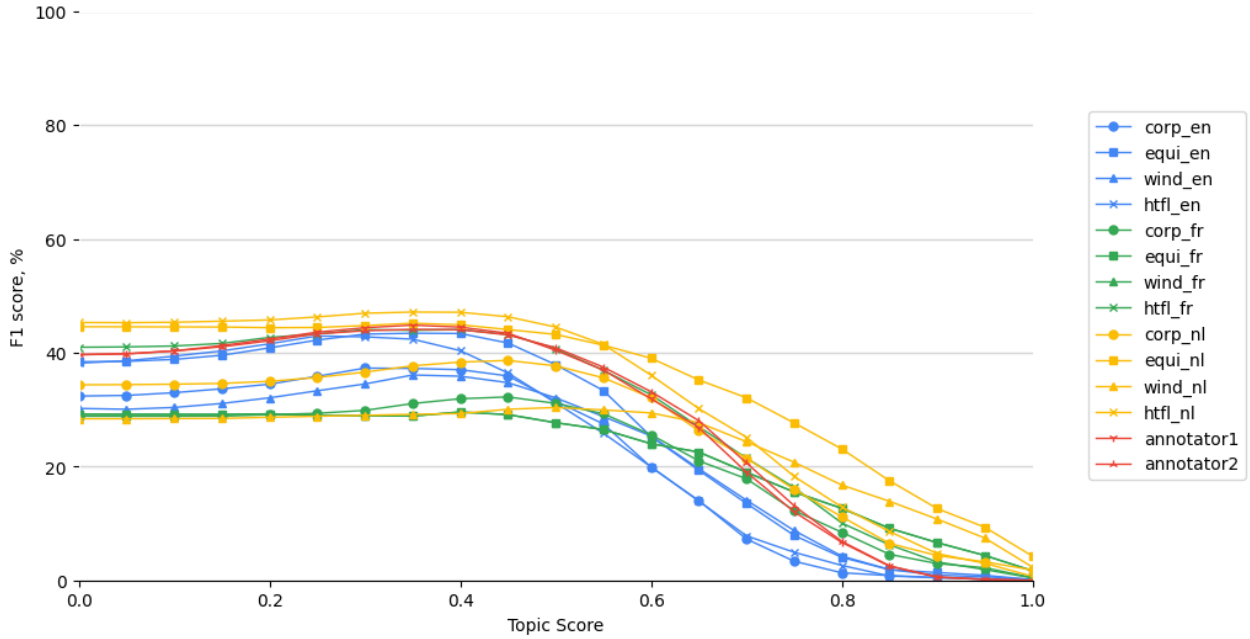

Figure 4: Variation of F1 score at different TopicScore thresholds on the ACTER and ACL datasets

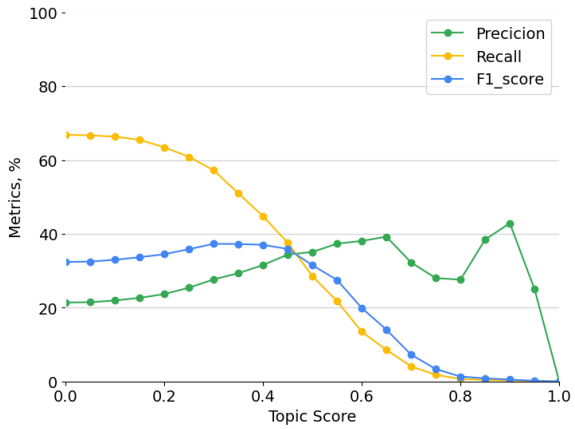

Figure 5: Variation of scores at different TopicScore thresholds on corp(en) domain from ACTER

The analysis showed that there is little change in recall at a rectified frequency (*F_rec*) above 4. Precision and the F1-score remain stable at rectified frequency values above 3. Therefore, a threshold of 3 for *F_rec* was chosen for the ACTER dataset.

For raw frequency, the indices stop changing at values above 5, meaning that setting the threshold in the range of 5 to 10 has a negligible impact. However, a threshold of 9 was chosen to improve precision, as the raw frequency does not reflect the significance of the terms as accurately as the rectified frequency. Lowering the thresholds leads to a decrease in precision, which may negatively affect the quality of the extracted phrases, as it may introduce errors in boundary detection or the selection of unrelated words.

Table 6 presents the effect of frequency thresholds on recall. The data shows how much recall increases when additional frequency filtering is used. If only thresholds are considered, phrases that may have already been retrieved in the grouping step may be extracted.

Frequency threshold filtering has a more significant impact on small corpora such as ACL. In the case of large texts, threshold filtering gives only a minor addition to the core set of extracted phrases. However, this approach is effective for small texts where grouping-based filtering did not provide a significant gain in the number of extracted terms.

For generating contextual vectors of unigrams, phrases and sentences, the model *"sentence-transformers/all-MiniLM-L6-v2"* for English and *"sentence-transformers/paraphrase-multilingual-mpnet-base-v2"* for French and Dutch is used. The TopicScore method uses a similarity threshold value (0.4) for both unigrams and phrases. If the cosine similarity between a unigram or phrase vector and a sentence vector exceeds this threshold, such a unit is classified as a term.

The selection of the optimal threshold for TopicScore is based on an analysis of the F1-score, precision, and recall metrics at different threshold values. Figure 4 shows that F1-score reaches its maximum value at TopicScore in the range of 0.3-0.4, and its variation in this interval is insignificant. The mean value of the threshold at which F1-score was maximized is 0.375.

However, TopicScore has a significant impact on the balance between recall and precision of candidate term selection, which is illustrated in Figure 5. Here is an example of performance variation as a function of the threshold. Since precision was prioritized over recall at this stage of the study, a threshold of 0.4 was chosen as the optimal value.

The list of stop words was taken from the GitHub repository *"term-extraction-project/stop_words"* for English, while for French and Dutch, data from the *"stopwords-iso"* repository was used.

5 Evaluation method

Precision, Recall, and F1-score metrics are used to evaluate the effectiveness of the developed T-Extractor approach. Recall (R) characterizes the proportion of correctly extracted terms out of all relevant terms. Precision (P) indicates the percentage of extracted terms that are correct. The F1-score represents the harmonic mean of Precision and Recall, providing a balanced assessment of the model's quality.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

For a more in-depth analysis of the proposed approach's performance, the model was evaluated at four key stages of term and named entity extraction: (1) after candidate extraction, (2) after semantic filtering with TopicScore, (3) after adding abbreviations, and (4) at the final stage, after extracting PROPN sequences.

The first step involves extracting and pre-filtering candidates, covering all steps before applying TopicScore filtering, and is labeled Extract Candidates. The second step reflects the result after TopicScore filtering and is referred to as TopicScore Filter. The third step shows score changes after adding extracted words based on the abbreviation rule and is labeled Abb Extract. The final step involves the extraction of sequences with PROPN tag, more related to named entities, and is called NE Extract. This step shows the results after adding extracted candidates using the algorithm for extracting named entities.

To better understand the quality of term extraction in the corp domain from the ACTER (en) dataset, the results computed at each stage are presented, focusing on evaluating the extraction performance of unigrams (uni), phrases (mwe), and generic terms (All). Unigrams and multi-word expressions were evaluated separately, as their extraction methods differ significantly. Extracted unigrams (uni) were compared to true single-word terms, while extracted phrases (mwe) were compared to true multi-word terms. This approach allows a detailed analysis of the performance of each extraction step and evaluates its impact on the overall quality of the extracted terms.

To evaluate the effectiveness of the two rules for extracting abbreviations and proper noun sequences, the precision of term and named entity identification was measured. The proportion of extracted abbreviations was determined by comparison with a set of true terms. Similarly, the identified named entity sequences were compared to a reference list to calculate their proportion among the true terms.

Finally, the final F1-score for the T-Extractor term extraction method was compared with the results of other term extraction methods, both supervised and unsupervised, such as HAMLET, GPT-3.5-Turbo, PromptATE, TALN-LS2N, BERT3, BERT6, NMF, UA, and UA1.

6 Results

The results of term and named entity extraction using T-Extractor are presented in Table 7. The first stage (Candidate extraction), which includes candidate extraction and pre-filtering, identifies, on average, about 70% of the true terms, with a precision of 24% and an F1-score of 35%. At this stage, the F1-score for T-Extractor already outperforms the results of many unsupervised approaches presented in Table 10.

After applying filtering using Topic Score, the F1-score increases by 3.1% on average. Precision increases

Table 7: Results (%) of extracting terms and named entities using the T-extractor annotator.

Data set	Lang	Domain	Candidate extract			Topic Score filter			Abb extract			NE extract		
			P	R	F1	P	R	F1	P	R	F1	P	R	F1
ACL-RD-TEC 2.0	en	annotator 1	27,3	72,1	39,6	36,0	58,1	44,5	36,0	59,5	44,8	35,0	61,2	44,5
		annotator 2	27,2	73,0	39,6	35,1	59,1	44,0	35,0	60,3	44,2	33,8	61,8	43,7
ACTER	en	Corp	21,4	67,0	32,4	31,6	44,8	37,0	31,4	46,7	37,6	31,5	55,3	40,1
		Equi	26,6	69,5	38,4	40,2	47,1	43,4	40,3	47,6	43,7	41,6	58,2	48,5
		Wind	19,6	67,3	30,4	29,8	45,1	35,9	29,7	47,7	36,6	28,6	58,3	38,4
		HTFL	28,2	58,4	38,0	42,8	38,2	40,3	44,8	43,0	43,9	43,7	48,7	46,0
	fr	Corp	18,7	64,2	28,9	24,2	47,7	32,1	24,7	50,4	33,2	25,4	53,4	34,5
		Equi	18,9	63,6	29,1	21,7	46,6	29,6	21,7	47,0	29,7	22,7	51,6	31,5
		Wind	14,0	67,3	23,2	16,6	55,9	25,6	17,0	58,9	26,4	17,4	63,0	27,3
		HTFL	30,1	64,0	40,9	41,8	46,5	44,1	42,9	49,4	45,9	42,5	50,8	46,3
	nl	Corp	22,4	73,4	34,4	28,4	59,0	38,4	28,8	60,5	39,0	29,0	63,8	39,9
		Equi	32,0	72,9	44,4	35,1	62,2	44,9	35,2	62,6	45,1	35,6	65,9	46,2
		Wind	17,5	75,4	28,3	19,4	60,3	29,3	19,7	61,6	29,8	20,8	68,5	32,0
		HTFL	32,2	76,3	45,3	39,9	57,5	47,1	40,9	60,3	48,7	40,4	62,2	49,0
Average			24,0	68,9	35,2	31,6	52,0	38,3	32,0	54,0	39,2	32,0	58,8	40,6

Table 8: Results (%) of term and named entity extraction by processing stage for the domain "Corruption" (Corp) from the ACTER (en) dataset divided into unigrams (Uni), phrases (MWE) and all terms (All).

Dataset	Corp	Candidate extract			Topic Score filter			Abb extract			NE extract		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
ACTER (en)	Uni	19,6	88,1	32,1	35,6	48,2	41,0	34,8	53,0	42,0	35,0	60,8	44,4
	MWE	23,8	52,6	32,8	29,0	42,4	34,4	29,0	42,4	34,4	29,1	51,6	37,2
	All	21,4	67	32,4	31,6	44,8	37,0	31,4	46,7	37,6	31,5	55,3	40,1

significantly by about 7.6%, while recall drops by 16.9%. After Topic Score filtering, the F1-score averages 38.3%.

In the next stage (Abb extract), after adding the candidates extracted using abbreviation rules, both precision and recall increase slightly by about 1–2%. The average F1-score in this step is 39.2%.

The final stage involves extracting PROPN sequences (NE extraction), which increases the F1-score by 1.4%. In summary, after all the steps of term and named entity extraction, the average precision is 32%, the average recall is 58.8%, and the average F1-score is 40.6%.

Table 8 presents the results of term and named entity extraction for the *"corruption"* (Corp) domain from the ACTER (en) dataset, categorized into unigrams (uni), phrases (mwe), and all terms across different processing steps. These data show the dynamics of improvement in the results of term and named entity extraction at each processing stage for different types of terms. The analysis shows that at almost all stages, phrase extraction performed worse than unigram extraction.

In the candidate extraction stage, the F1-scores for unigrams and phrases were almost similar, but the differences in precision and recall were significant. Higher recall and lower precision were observed for unigrams than for phrases. This indicates that with further filtering, the recall for phrases will decrease significantly, which in turn may negatively affect the performance of the model.

After applying the Topic Score filter, the F1-score for unigrams increased significantly and became higher than that for phrases. Although unigram recall dropped significantly, precision increased by 16%, indicating more

Table 9: Precision (%) evaluation of extracting relevant terms using rules for finding abbreviations and proper noun sequences.

Dataset	Lang	Domain	Abbreviation	Named Entities
ACL-RD-TEC 2.0	en	annotator 1	45,89	36,49
		annotator 2	46,56	35,01
ACTER	en	Corp	39,55	40,66
		Equi	64,00	59,53
		Wind	34,81	32,05
		HTFL	73,56	55,63
	fr	Corp	37,75	54,64
		Equi	57,41	46,84
		Wind	28,31	30,82
		HTFL	72,14	57,35
	nl	Corp	47,11	44,71
		Equi	62,77	61,44
		Wind	57,43	49,61
		HTFL	80,00	55,38
Average			53,4	47,2

effective filtering. The F1-score for phrases increased by only 1.6%, which confirms that Topic Score filter works more effectively for unigrams.

After the abbreviation extraction step (Abb extract), the F1-score for unigrams increased slightly, which is because this step is mainly focused on extracting specific unigrams. In the case of unigrams, the precision slightly decreased but the recall increased, which indicates that this stage extracted meaningful words that can be both terms and named entities.

At the PROPN sequence extraction (NE extraction) stage, the overall F1-score for this domain reached 40.1%. The F1-scores for unigrams and phrases also increased, indicating the importance of this stage for the extraction of meaningful lexical units. Recall for unigrams and phrases is at around 50%, while precision is around 30%. Despite this, the performance for phrases remains lower than for unigrams, indicating the need to optimize the approach for phrase extraction.

Thus, although phrase extraction significantly lags unigram extraction at almost all stages, the term and named entity extraction process demonstrate performance improvements at each step. Filtering techniques such as Topic Score filter show better results for unigrams, while phrases require further optimization.

Table 9 shows the precision evaluation of extracting terms and named entities using two simple rules: for extracting abbreviations and sequences of proper nouns.

The average extraction precision using the abbreviation rule is 53.4%, indicating the ability of the approach to extract relevant words with over 50% precision. However, the approach performed poorly in the wind domain in English and French.

The average precision of extracting relevant words using the rule for noun sequences is 47.2%. Despite this, the approach showed efficiency and high retrieval precision in several cases. However, the rule was less successful for the ACL RD-TEC 2.0 dataset, as well as for the wind domain in all languages of the ACTER corpus.

Table 10 shows the F1-score results for the T-Extractor annotator compared to other supervised and unsupervised term extraction methods. Figure 6 presents the F1 score results for the HAMLET [27], T-Extractor, and NMF [20] annotators applied to the ACTER corpus in three languages. It is evident that T-Extractor has significantly reduced the gap between supervised and unsupervised approaches, closely approaching HAMLET in several cases. Its performance in the Corp domain was comparable to that of the supervised method. However, in the equi domain for French and Dutch, a notable performance gap remains, likely due to language-specific characteristics that were not accounted for during adaptation. Nevertheless, T-Extractor outperformed the unsupervised NMF approach, achieving a higher F1 score. A detailed comparison of its performance with supervised and unsupervised methods is presented below.

The T-Extractor method shows superiority over most unsupervised approaches when applied to the ACTER dataset. However, on the ACL RD-TEC 2.0 dataset, its F1-score (44.5%) is inferior to that of the UA method (50%). At the same time, the re-evaluated version of the UA approach, denoted as UA1, showed an average F1 score 8.3% lower on the ACTER dataset compared to the T-Extractor method.

Compared to the NMF method [20], the T-Extractor annotator performs better across all domains and languages. For example, on the corp_en domain, T-Extractor achieves 40.12% while NMF shows 25.72%. A similar difference is observed across all other languages and domains, confirming the effectiveness of T-Extractor in unsupervised term extraction tasks. However, on the Equi(fr) domain, the difference between T-Extractor (31.5%) and NMF (27.2%) is only 4.3%, suggesting low language- and domain-specific dependency. This highlights the need for further improvements to the T-Extractor method, as its performance may vary depending on the context.

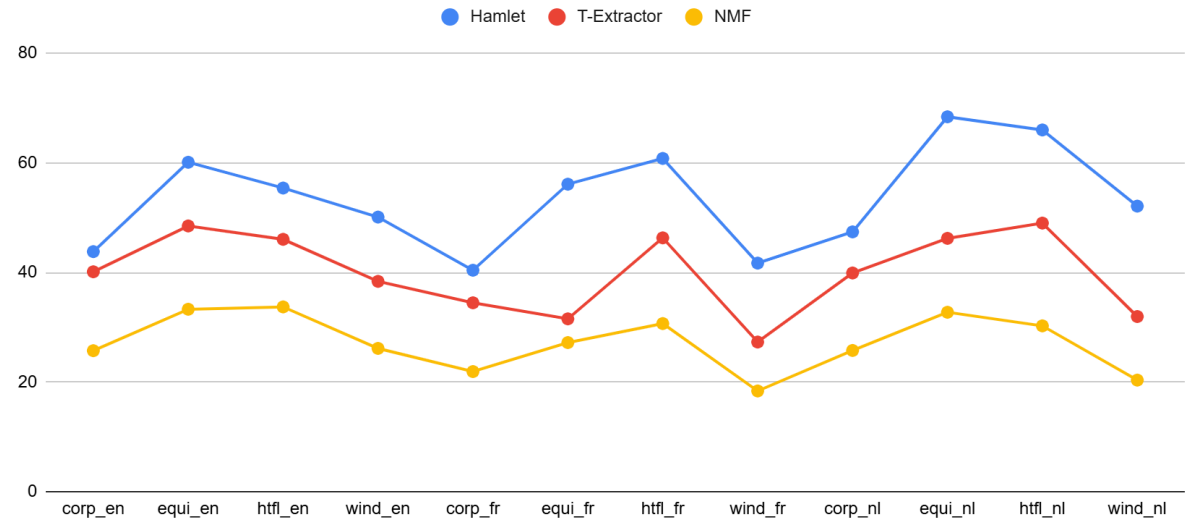

Figure 6: Comparative analysis of F1 score (%) for HAMLET, T-Extractor, and NMF in term and named entity extraction on the ACTER corpus

Table 10: Comparison of F1 scores (%) of the T-Extractor annotator with other supervised and unsupervised methods on ACTER data set.

Annotators	**en**				**fr**				**nl**			
	Corp	**Equi**	**Wind**	**HTFL**	**Corp**	**Equi**	**Wind**	**HTFL**	**Corp**	**Equi**	**Wind**	**HTFL**
Supervised												
HAMLET [27]	43,8	60,1	50,1	55,4	40,4	56,1	41,7	60,8	47,4	68,4	52,1	66,0
GPT-3.5-Turbo [29]	31,4	49,7	32,5	55,6	-	-	-	-	-	-	-	-
promptATE (Llama 2-Chat) OF#3 [30]	-	-	-	51,4	-	-	-	47,8	-	-	-	55,4
TALN-LS2N [28]	-	-	-	46,66	-	-	-	48,15	-	-	-	-
BERT3 [28]	32,8	42,2	34,8	45,7	29,6	36,4	27,8	48,4	-	-	-	-
BERT6 [28]	35,5	41,6	28,5	44,1	41,7	16,2	16,4	36,9	-	-	-	-
Unsupervised												
NMF [20]	25,7	33,3	26,1	33,7	21,9	27,2	18,4	30,7	25,8	32,7	20,4	30,3
UA1 [22]	24,3	28,9	29,5	32,7	-	-	-	-	-	-	-	-
T-Extractor	40,1	48,5	38,4	46,0	34,5	31,5	27,3	46,3	39,9	46,2	32,0	49,0

As for the supervised methods, the HAMLET annotator [27] significantly outperforms T-Extractor. However, compared to other supervised methods, T-Extractor achieves results in some domains that are either superior or not significantly lower. This indicates that T-Extractor, despite its unsupervised nature, is an effective tool for term extraction and its results may be comparable or even superior to supervised approaches in some cases.

The HAMLET annotator, a supervised method, outperforms T-Extractor across all metrics. This is particularly evident in Figure 6, where HAMLET performs well, especially on domains related to French and Dutch languages, where its performance is significantly higher compared to the other methods. However, the difference between HAMLET and T-Extractor is not always so large. For example, in the corp_en domain, T-Extractor performs competitively, achieving 40.1% compared to HAMLET's 43.8%. This indicates that T-Extractor can show results comparable to supervised methods in certain contexts.

The GPT-3.5-Turbo model [29] performs better than T-Extractor, especially in the HTFL domain (ACTER en), where its F1-score is 9.6% higher than T-Extractor. Nevertheless, overall, T-Extractor's performance is not significantly lower than that of the supervised GPT-3.5-Turbo method. For example, in the corp and wind domains, the T-Extractor method outperforms the GPT-3.5-Turbo method by 8.7% and 5.9%, respectively. In the equi domain, F1 performance is almost identical, with a difference of only 1.2% in favor of GPT-3.5-Turbo.

The promptATE (Llama 2-Chat, OF#3) [30] method outperforms T-Extractor in all three languages, but its performance in other domains remains uncertain. It should be noted that on the HTFL(fr) dataset, the differences in results are insignificant. For a more accurate and objective evaluation of promptATE's performance, it is necessary to analyze its performance on additional domains and datasets.

The TALN-LS2N method [28] also outperforms T-Extractor, but the difference in results is not significant. However, TALN-LS2N requires a significant amount of labeled data, which limits its applicability when there is a lack of high-quality annotation.

As for BERT3 and BERT6 [28], their performance is on average slightly inferior to T-Extractor, especially in English. However, they perform better on some other languages, e.g., BERT3 shows a slight superiority over T-Extractor on French. BERT6 significantly outperforms T-Extractor on the Corruption (French) domain, but is inferior on the other domains, indicating that its performance is heterogeneous across languages and domains.

To evaluate the statistical significance of the T-Extractor results, a paired t-test was conducted with the NMF approach. This annotator was chosen for comparison with T-Extractor because both are unsupervised methods and were tested on the largest number of texts compared to other approaches.

The results of the paired t-test showed that the t-statistic was 12.31 and the p-value was 8.96×10^{-8}. Since the p-value is significantly lower than the standard threshold of 0.05, the difference between the methods is statistically significant, thus rejecting the null hypothesis that there is no difference in their quality. This indicates that T-Extractor significantly outperforms NMF in terms of the F-measure, demonstrating superior performance in term and named entity extraction.

In general, the T-Extractor annotator shows competitive results in term extraction, outperforming many unsupervised methods. Despite lagging significantly behind the supervised method HAMLET, T-Extractor achieves comparable results in some contexts, such as in the corp_en domain. Overall, the performance

of T-Extractor can be close to that of other supervised methods such as GPT-3.5-Turbo, and in some domains and languages even outperforms them.

7 Discussion

The advantage of T-Extractor over other annotators is the integration of statistical and semantic approaches for term extraction, as well as its independence from labeled data. The T-Extractor exhibits high recall in the candidate extraction phase, ensuring that more potential terms are retained in subsequent filtering steps compared to alternative methods. The use of part-of-speech patterns instead of n-grams, as in approaches such as NMF, TALN-LS2N, or BERT-based models, contributes to extracting more meaningful word combinations and improves the accuracy of the method. In addition, the T-Extractor can identify longer terminological candidates rather than being limited to unigrams and pentagrams.

The proposed methodology for customizing part-of-speech patterns provides greater flexibility in forming terminological expressions and reduces the cost of manually enumerating possible part-of-speech combinations. This approach demonstrates advantages over UA, which is limited exclusively to noun phrases. In addition, the noun chunks mechanism does not always efficiently identify phrase boundaries, which was noted by the authors when implementing the UA1 method, leading to incorrect identification of terminological candidates. In the T-extractor, term boundaries are determined based on rectified and raw frequency measures, which increases its efficiency when working with large corpora.

An additional factor affecting the efficiency of the T-Extractor is its improved text preprocessing and filtering system. In particular, the use of multi-level cleaning mechanisms in the candidate extraction stage, the setting of spaCy to avoid splitting multiword terms with hyphenation, and the preservation of the original case during POS-tagging have contributed to minimizing noise. For example, in TALN-LS2N, the candidate filtering step is described in less detail: the authors only exclude a limited set of undesirable classes, such as words starting with conjunctions and pronouns. In the GPT-3.5-Turbo and promptATE methods, where candidates are generated automatically, a check for their presence in the source text is applied, but these approaches show a tendency to select common words. In this context, additional cleaning of stop words or applying semantic filtering could improve the relevance of the extracted terms.

The use of semantic filtering in the T-Extractor allowed for the extraction of more topic-relevant candidates. In UA, this mechanism was applied only to multi-word expressions, but not to unigrams, which probably negatively affected the quality of term extraction in its UA1 version tested on the ACTER dataset. NMF also lacks semantic filtering, which may have reduced the performance of the method.

Compared to HAMLET, the key advantage of T-Extractor is that it does not require annotated data, but it is inferior in candidate extraction performance. Like HAMLET, T-extractor uses various features to identify terms including statistical, linguistic and semantic characteristics. It is possible that the use of a hybrid approach combining different features, heuristics and filtering methods allowed the T-extractor running in unsupervised mode to achieve closer performance to supervised methods than other unsupervised approaches.

One of the key features of T-Extractor is its ability to extract unigrams more efficiently than multi-word terms. This is because the quality of phrase extraction largely depends on the correct definition of phrase boundaries. The proposed approach is based on frequency characteristics, which may reduce its efficiency when processing small texts. In addition, T-Extractor excludes phrases that occur only once in the text, which potentially affects the recall of term extraction. Unlike multi-word expressions, single-word terms do not require additional boundary refinement, ensuring their higher recall.

Unigrams are filtered more efficiently than phrase expressions in the Topic Score filter. This may be because it is easier to form a meaningful vector representation for unigrams compared to phrases, or due to their higher recall, which results in fewer relevant candidates being retained among multi-word terms. This stage is also sensitive to the choice of model for generating vector representations of the context, which has a direct impact on the quality of term extraction.

In the step of adding abbreviations, the efficiency of term extraction depends on the correct text case. Additionally, adding abbreviations primarily enhances unigram extraction results, as confirmed by the test results.

Named entity extraction significantly improves both unigram and phrase extraction, particularly in texts where such entities appear infrequently. This approach helps to increase both the recall and precision of phrase extraction, making the process more accurate and comprehensive. An interesting observation is that some named entities and abbreviations, extracted along with unigrams and phrases in the Candidate Extract step, may be filtered out during the Topic Score step. However, additional extraction rules enable the recovery of filtered candidates, ultimately enhancing overall annotation results.

Analysis of the ACTER corpus data presented in Figure 6 reveals patterns related to domain and language dependency. In some domains, such as Equi and HTFL, annotators perform well, whereas in others, such as corp and wind, their performance declines significantly. Additionally, in French, the HAMLET, NMF, and T-Extractor methods yielded lower results than in English and Dutch, confirming the language dependency of these approaches.

The average Pearson correlation between the annotators' results was 0.797, indicating a strong and positive correlation. This may also indicate the specificity of text structure for different domains and languages. Considering these factors may play a key role in understanding term features and improving term extraction in various contexts.

Thus, the study provided answers to the research questions posed.

Firstly, the impact of combined features on the term extraction process remains significant, despite the continuous advancement and improvement of deep learning models. This suggests that despite the availability of powerful neural network-based methods, traditional linguistic and statistical approaches remain crucial in terminology processing. Moreover, this observation supports the hypothesis that applied linguistics is unlikely to become solely the domain of deep learning research. Rather, it is expected to remain an interdisciplinary field at the intersection of linguistics, statistics, and computational methods.

Secondly, the analysis demonstrated that in the absence of annotated training data, the significance of utilizing the T-Extractor tool increases. This is because its methodology compensates for the lack of labeled corpora by leveraging heuristics, statistical patterns, and pre-existing knowledge about terms. As a result, automatic term extraction methods can operate effectively even with limited training data, making them valuable tools for low-resource languages and specialized domains.

8 Conclusions

This paper analyzed the performance of the T-Extractor annotator in unsupervised term extraction tasks and compared it with other methods including supervised and unsupervised approaches. The results showed that T-Extractor is a competitive tool that shows stable performance on different languages and domains.

The main advantages of T-Extractor lie in its ability to work without annotated data, which makes it suitable for text processing in resource-constrained environments. Using a combination of rules, statistical and semantic analysis, it achieved high retrieval recall. However, the annotator showed lower precision, indicating that candidate filtering mechanisms need to be improved, especially in the phrase boundary detection phase.

T-Extractor is particularly better at extracting unigrams, while phrase extraction is more difficult due to its dependence on frequency characteristics. In the Candidate Extract step, the limitation of the method manifests itself in the inability to extract rare phrases, which reduces recall. The addition of named entity and abbreviation processing steps has a positive impact on recall and precision, especially in texts with rare entities. Additional rules for recovering filtered candidates also contributed to the improvement of the metrics.

Comparison with other unsupervised methods showed that T-Extractor outperforms them in almost all domains and languages. For example, on the corp(en) domain, T-Extractor achieved 40.12% on the F1 metric, while NMF demonstrated 25.72%. However, on individual domains such as equi(fr), the difference between the methods is minimal (31.5% for T-Extractor vs. 27.2% for NMF), indicating that the method can be further optimized.

The supervised method HAMLET shows significantly better results. For example, the average difference in F1 metric between HAMLET and T-Extractor is 9.1% (in English), 14.9% (in French), and 16.7% (in Dutch). However, in some domains, such as corp (en), T-Extractor achieves performance close to supervised approaches (40.1% compared to 43.8% for HAMLET). Compared to other supervised approaches, T-Extractor showed similar results and even outperformed some in certain domains.

Currently, one of the main limitations of the T-Extractor approach is the difficulty of accurately identifying phrase boundaries and extracting low-frequency phrases. A promising direction for future research is to enhance phrase boundary detection algorithms by employing syntactic analysis techniques, such as constructing syntax trees or analyzing word dependencies. Additionally, incorporating artificial intelligence techniques could further improve the precision of phrase boundary identification.

To refine semantic filtering, another potential improvement is integrating static vector representations of words. This approach would allow the model to account not only for contextual dependencies but also for the invariant lexical meaning of terms, leading to more accurate filtering and selection.

Furthermore, the development of a classification module for extracted terms presents another promising avenue. This module could categorize terms based on multiple criteria, distinguishing domain-specific, general, and out-of-domain terms, as well as classifying them thematically according to the text's content.

In addition, classifying named entities according to the MUC-7 scheme could be incorporated, providing a more detailed and structured representation of extracted entities. It is expected that integrating such a classifier would not only enhance the quality of term extraction but also increase the significance of T-Extractor as a tool for processing specialized texts.

Acknowledgement

This research has been funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number AP19677756 «Unsupervised term extraction: a set of models and datasets for high-tech domains and low-resource languages».

Data availability statement

The code is available in the GitHub repository https://github.com/term-extraction-project/T-Extractor (accessed on 21 January 2025)

Abbreviations

The following abbreviations are used in this manuscript:

T-Extractor	Term Extractor
ACTER	Annotated Corpora for Term Extraction Research
ACL RD-TEC 2.0	Association for Computational Linguistics Reference Dataset for Terminology Extraction and Classification, version 2.0
ACL	Association for Computational Linguistics

HAMLET	Hybrid Adaptable Machine Learning approach to Extract Terminology
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
NE	Named Entity
NMF	Non-Negative Matrix Factorization
UA	Unsupervised Annotator
UA1	Unsupervised Annotator 1
YAKE	Yet Another Keyword Extractor
ChatGPT	Chat Generative Pre-Trained Transforme
Llama	Large Language Model Meta AI
OOV	Out-Of-Vocabulary
POS	Part-of-Speech
MUC-7	Message Understanding Conference 7

References

[1] Hubauer, T.; Lamparter, S.; Haase, P.; Herzig, D. M. Use Cases of the Industrial Knowledge Graph at Siemens. *International Workshop on the Semantic Web*, 2018.

[2] Zhou, D.; Zhou, B.; Zheng, Z.; Soylu, A.; Cheng, G.; Jimenez-Ruiz, E.; Kostylev, E. V.; Kharlamov, E. Ontology Reshaping for Knowledge Graph Construction: Applied on Bosch Welding Case. In *The Semantic Web – ISWC 2022; Springer-Verlag*: Berlin, Heidelberg, 2022; pp. 770–790. https://doi.org/10.1007/978-3-031-19433-7_44.

[3] Dirksen, N.; Takahashi, S. Artificial Intelligence in Japan 2020; *Netherlands Enterprise Agency*, 2020.

[4] Shiroishi, Y.; Uchiyama, K.; Suzuki, N. Better Actions for Society 5.0: Using AI for Evidence-Based Policy Making That Keeps Humans in the Loop. *Computer*, 2019, 52 (11), 73–78. https://doi.org/10.1109/mc.2019.2934592.

[5] Rožanec, J. M.; Novalija, I.; Zajec, P.; Kenda, K.; Tavakoli Ghinani, H.; Suh, S.; Veliou, E.; Papamartzivanos, D.; Giannetsos, T.; Menesidou, S. A.; Alonso, R.; Cauli, N.; Meloni, A.; Recupero, D. R.; Kyriazis, D.; Sofianidis, G.; Theodoropoulos, S.; Fortuna, B.; Mladenić, D.; Soldatos, J. Human-Centric Artificial Intelligence Architecture for Industry 5.0 Applications. *International Journal of Production Research*, 2022, 61 (5), 1–26. https://doi.org/10.1080/00207543.2022.2138611.

[6] Eiden, M. Connecting the Dots with Knowledge Graphs — Opening Statement | *Cutter Consortium*. Cutter.com. https://www.cutter.com/article/connecting-dots-knowledge-graphs.

[7] Drouin, P.; Grabar, N.; Hamon, T.; Kageura, K.; Takeuchi, K. Computational Terminology and Filtering of Terminological Information. *Terminology International Journal of Theoretical and Applied Issues in Specialized Communication*, 2018, 24 (1). https://doi.org/10.1075/term.24.1.

[8] Curcic, D. Number of Academic Papers Published Per Year – *WordsRated*. *Wordsrated*. https://wordsrated.com/number-of-academic-papers-published-per-year/.

[9] Du, R.; An, H.; Wang, K.; Liu, W. A Short Review for Ontology Learning: Stride to Large Language Models Trend. *arXiv (Cornell University)*, 2024. https://doi.org/10.48550/arxiv.2404.14991.

[10] Tran, H.; Martinc, M.; Caporusso, J.; Doucet, A.; Pollak, S. The Recent Advances in Automatic Term Extraction: A Survey. *arXiv (Cornell University)*, 2023. https://doi.org/10.48550/arxiv.2301.06767.

[11] Wang, K.; Gu, S.; Chen, B.; Zhao, Y.; Luo, W.; Zhang, Y. TermMind: Alibaba's WMT21 Machine Translation Using Terminologies Task Submission. In *Proceedings of the Sixth Conference on Machine Translation; Association for Computational Linguistics*, 2021; pp. 851–856.

[12] Huy, H. N. L.; Minh, H. H.; Van, T. N.; Van, H. N. Keyphrase Extraction Model: A New Design and Application on Tourism Information. *Informatica*, 2021, 45 (4), 563-569. https://doi.org/10.31449/inf.v45i4.3493.

[13] Kimura, Y.; Komamizu, T.; Hatano, K. An Automatic Labeling Method for Subword-Phrase Recognition in Effective Text Classification. *Informatica*, 2023, 47 (3), 315–326. https://doi.org/10.31449/inf.v47i3.4742.

[14] Michon, E.; Crego, J.; Senellart, J. Integrating Domain Terminology into Neural Machine Translation. In *Proceedings of the 28th International Conference on Computational Linguistics*; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 3925–3937. https://doi.org/10.18653/v1/2020.coling-main.348.

[15] Condamines, A.; Picton, A. Textual Terminology : Origins, Principles and New Challenges. In *Theoretical Approaches to Terminology; John Benjamins*, 2022; pp. 219–236. https://doi.org/10.1075/tlrp.23.10con.

[16] Jaleniauskienė, E.; Čičelytė, V. Insight into the Latest Computer and Internet Terminology. *Studies About Languages*, 2011, 0 (19). https://doi.org/10.5755/j01.sal.0.19.955.

[17] Zhang, J.; Chen, S.; Hua, J.; Niu, N.; Liu, C. Automatic Terminology Extraction and Ranking for Feature Modeling. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*; Melbourne, Australia, 2022; pp. 51–63. https://doi.org/10.1109/re54965.2022.00012.

[18] Kafando, R.; Decoupes, R.; Valentin, S.; Sautot, L.; Teisseire, M.; Roche, M. ITEXT-BIO: Intelligent Term EXTraction for BIOmedical Analysis. *Health Information Science and Systems*, 2021, 9. https://doi.org/10.1007/s13755-021-00156-6.

[19] Terryn, A. R.; Hoste, V.; Drouin, P.; Lefever, E. TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. In *Proceedings of the 6th International Workshop on Computational Terminology*; European Language

Resources Association: Marseille, France, 2020; pp. 85–94.

[20] Nugumanova, A.; Akhmed-Zaki, D.; Mansurova, M.; Baiburin, Y.; Maulit, A. NMF-Based Approach to Automatic Term Extraction. *Expert Systems with Applications*, 2022, 199, 117179. https://doi.org/10.1016/j.eswa.2022.117179.

[21] Fusco, F.; Staar, P.; Antognini, D. Unsupervised Term Extraction for Highly Technical Domains. *arXiv (Cornell University)*, 2022. https://doi.org/10.48550/arxiv.2210.13118.

[22] Kalykulova, A.; Kairatuly, B.; Rakhymbek, K.; Kyzyrkanov, A.; Nugumanova, A. Evaluation of IBM's Proposed Term Extraction Approach on the ACTER Corpus. In *IX — International Scientific Conference "Computer Science and Applied Mathematics"*; Almaty, Kazakhstan, 2024; pp. 597–604.

[23] Firoozeh, N.; Nazarenko, A.; Alizon, F.; Daille, B. Keyword Extraction: Issues and Methods. *Natural Language Engineering*, 2019, 26 (3), 259–291. https://doi.org/10.1017/s1351324919000457.

[24] Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; Jatowt, A. YAKE! Keyword Extraction from Single Documents Using Multiple Local Features. *Information Sciences*, 2020, 509, 257–289. https://doi.org/10.1016/j.ins.2019.09.013.

[25] Anjum, O.; Almasri, M.; Xiong, J.; Hwu, W. PhraseScope: An Effective and Unsupervised Framework for Mining High Quality Phrases. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*; 2021; pp. 639–647. https://doi.org/10.1137/1.9781611976700.72.

[26] Di Nunzio, G. M.; Marchesin, S.; Silvello, G. A Systematic Review of Automatic Term Extraction: What Happened in 2022? *Digital Scholarship in the Humanities*, 2023, 38 (Supplement_1), i41–i47. https://doi.org/10.1093/llc/fqad030.

[27] Terryn, A. R.; Hoste, V.; Lefever, E. HAMLET: Hybrid Adaptable Machine Learning Approach to Extract Terminology. *Terminology International Journal of Theoretical and Applied Issues in Specialized Communication*, 2021, 27 (2), 254–293. https://doi.org/10.1075/term.20017.rig.

[28] Hazem, A.; Bouhandi, M.; Boudin, F.; Daille, B. TermEval 2020: TALN-LS2N System for Automatic Term Extraction. In *Proceedings of the 6th International Workshop on Computational Terminology*; European Language Resources Association: Marseille, France, 2020; pp. 95–100.

[29] Banerjee, S.; Chakravarthi, B. R.; McCrae, J. P. Large Language Models for Few-Shot Automatic Term Extraction. In *Natural Language Processing and Information Systems*; Springer, Cham, 2024; Vol. 14762, pp. 137–150. https://doi.org/10.1007/978-3-031-70239-6_10.

[30] Tran, H. T. H.; González-Gallardo, C.-E.; Delaunay, J.; Doucet, A.; Pollak, S. Is Prompting What Term Extraction Needs? In *27th International Conference, TSD 2024*; Springer-Verlag: Berlin, Heidelberg, 2024; Vol. 15048, pp. 17–29. https://doi.org/10.1007/978-3-031-70563-2_2.

[31] QasemiZadeh, B.; Schumann, A.-K. The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*; European Language Resources Association (ELRA): Portorož, Slovenia, 2016; pp. 1862–1868.

[32] Terryn, A. R.; Hoste, V.; Lefever, E. A Gold Standard for Multilingual Automatic Term Extraction from Comparable Corpora: Term Structure and Translation Equivalents. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.

[33] Oliver, A.; Mercè Vàzquez. TermEval 2020: Using TSR Filtering Method to Improve Automatic Term Extraction. In *Proceedings of the 6th International Workshop on Computational Terminology*; European Language Resources Association: Marseille, France, 2020; pp. 106–113.