Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Efficient Candidate Discovery for Chinese Domain Lexicon Construction via Recursive Lexical Combination

Yurun Lyu[1]

3007250027@email.cugb.edu.cn

China University of Geosciences (Beijing)

Zilin Wang[2]

University of International Business and Economics

Research Article

Keywords: Recursive Lexical Combination, domain lexicon construction, automatic term extraction, Chinese language resources, candidate discovery, terminology

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10783091/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Efficient Candidate Discovery for Chinese Domain Lexicon Construction via Recursive Lexical Combination

Yurun Lyu[1]*; Zilin Wang[2]

[1] China University of Geosciences (Beijing), No. 29 Xueyuan Road, Haidian District, Beijing 100083, China

[2] Business School, University of International Business and Economics, Beijing, China

* Corresponding author: Yurun Lyu; Email: 3007250027@email.cugb.edu.cn; Telephone: +86-18660816325

ORCID: Yurun Lyu, https://orcid.org/0009-0007-6164-4976; Zilin Wang, https://orcid.org/0009-0004-4166-1371

Acknowledgements

None.

Abstract

Domain lexicons are essential for identifying specialized concepts in text, but seed-based construction can miss emerging terminology. This study proposes Recursive Lexical Combination (RLC), a corpus-driven method that recursively expands contiguous Chinese lexical units while preserving word order and textual boundaries and terminates low-support paths using corpus support. Using national-level data asset policy texts, the method was configured on a validation set and evaluated on an independent held-out test set against seed-based lexicon, C-value, and direct n-gram baselines. RLC reduced candidate types from 35,436 to 15,341, achieving 56.71% candidate compression while retaining 530 gold-standard multi-unit document-term types with 80.55% candidate-pool recall. It also covered 474 of 602 true terms not found by the seed lexicon. Compression increased with term length and was accompanied by lower coverage of longer terms. After expert semantic validation, micro-averaged F1 increased from 0.2166 for automatic RLC to 0.4811. RLC therefore provides an efficient and traceable workflow for Chinese domain lexicon construction by separating search-space control from semantic validation and making the trade-off between candidate compression and term coverage explicit.

Keywords: Recursive Lexical Combination; domain lexicon construction; automatic term extraction; Chinese language resources; candidate discovery; terminology

1 Introduction

Textual data have become an important source of information in economics, finance, and management research. Corporate reports, policy documents, news articles, and announcements contain substantial information that is difficult to capture using conventional structured indicators. Text analysis has therefore been widely used to identify characteristics related to corporate strategy, risk, sentiment, and the operating environment (Gentzkow et al., 2019; Loughran & McDonald, 2020). In dictionary- or lexicon-based text measurement, domain lexicons determine which linguistic expressions are included in subsequent statistical analysis and indicator construction, and their coverage directly affects the conceptual boundaries represented by text-based variables. Previous studies have shown that the meaning of the same word may vary across languages and professional contexts, and that directly transferring general-purpose or cross-lingual dictionaries can introduce substantial domain bias (Du et al., 2022). As research increasingly extends from mature domains to emerging fields with rapidly evolving terminology, identifying newly appearing specialized expressions directly from target corpora has become a fundamental issue in domain lexicon construction.

Existing domain lexicons are often developed from established theories, professional knowledge, or manually selected core terms, which are then expanded through semantic similarity, word embeddings, or manual procedures. Such knowledge-driven approaches provide clear conceptual origins and facilitate control over the domain scope of the lexicon, but candidate discovery is also influenced by the initial vocabulary. Data assets provide a representative example. As institutional frameworks related to data factor markets, data circulation governance, data infrastructure, and data resource management continue to evolve, new specialized expressions have emerged in relevant policy texts. He et al. (2024) constructed a relatively systematic data asset lexicon by expanding predefined core terms using policy and regulatory texts, providing a lexical basis for text-based measurement of corporate data assets. However, its candidate search still begins with an established set of core terms. For expressions that emerge with institutional and practical developments, such as “可信数据空间” (trusted data space), “数据产品经营权” (data product operating rights), and “数据资源无形资产” (data resource intangible assets), domain lexicon construction also requires the ability to discover new terms directly from the target corpus rather than relying entirely on an existing vocabulary to define the candidate space.

The lexical structure of Chinese domain-specific texts further complicates this problem. Unlike English, Chinese does not contain explicit interword boundaries marked by spaces, and specialized concepts do not always correspond to single lexical items already identified by a word segmentation system. Instead, they may consist of multiple adjacent lexical units occurring in sequence. The original order, adjacency, and combination length of these lexical units therefore

affect the identification of specialized expressions. One direct strategy for discovering unknown terms is to enumerate all contiguous lexical-unit combinations within a predefined length range so that potential candidates are not prematurely excluded during generation. However, as combination length increases, direct enumeration produces a large number of low-frequency or semantically incomplete higher-order combinations, thereby expanding the scope of subsequent statistical screening and manual review. Domain-term discovery therefore concerns not only whether true terms can be identified, but also the size of the candidate search space required to achieve such coverage.

Automatic term extraction has progressed from statistical methods based on term frequency and C-value to machine learning, Transformer-based models, and generative language models, with continued advances in term boundary detection, cross-domain transfer, and semantic representation (Nugumanova et al., 2022; Tran et al., 2024; Xu et al., 2025). These approaches primarily focus on candidate generation, scoring, or final term classification, whereas practical domain lexicon construction also involves two related issues: the scope of candidate search and domain semantic validation. On the one hand, expanding the candidate space increases the opportunity to discover potential terms but also raises computational and review burdens. On the other hand, statistical features and semantic models can provide evidence for term identification, yet domain relevance, lexical completeness, and conceptual granularity remain dependent on specific professional contexts. For Chinese specialized texts with continuously expanding terminology, an important question is therefore whether the candidate search space can be reduced while maintaining high coverage of true terms, and whether corpus-driven candidate discovery can be effectively integrated with subsequent domain semantic validation.

To address this problem, this study proposes Recursive Lexical Combination (RLC). RLC uses lexical units obtained from Chinese word segmentation as its basic computational objects while preserving their original order, adjacency relationships, and structural boundaries in the source text. Starting from short combinations, lexical units are progressively extended to the right, and corpus support is used to determine whether each recursive path should continue. Low-support paths are terminated early, thereby avoiding further enumeration of the corresponding higher-order contiguous combinations. The resulting RLC candidate pool is subsequently processed using unified automatic output rules and expert semantic validation. National-level data asset policy texts are used as the experimental corpus. Parameters are selected on a Validation set, after which RLC is evaluated on an independent held-out Test set against the Seed-based Lexicon, C-value, and Direct n-gram methods. On the Test set, RLC reduces the number of candidate types from 35,436 to 15,341 relative to direct contiguous enumeration, corresponding to a Candidate Compression of 56.71%, while retaining 530 Gold_multi document-term types and achieving a Candidate Pool Recall of 80.55%. Among the 602 true terms not covered by the fixed Seed Lexicon, 474 are included in the RLC candidate pool, corresponding to a candidate coverage of 78.74%.

The contributions of this study lie primarily in the candidate discovery mechanism and evaluation framework for domain lexicon construction. First, RLC separates expansion eligibility along recursive paths from final term eligibility and uses corpus support for contiguous combinations to terminate low-support paths early. This preserves the original word order and generation provenance of candidates while reducing the enumeration of higher-order combinations that do not require further expansion. Second, domain lexicon construction is divided into two connected stages: candidate discovery and semantic validation. The automatic procedure is responsible for candidate search and search-space control, whereas expert validation determines candidate relevance, lexical completeness, and semantic boundaries. Third, the evaluation extends beyond final Precision, Recall, and F1 to include Candidate Pool Recall, Candidate Compression, the compression–coverage relationship across different term lengths, and seed-uncovered term discovery, thereby assessing domain lexicon construction from the perspective of the candidate generation process itself. This design provides a traceable candidate discovery approach for Chinese specialized texts characterized by continuously evolving terminology and frequent compound expressions.

2 Literature Review

2.1 Domain Lexicon Construction and Automatic Term Extraction

With the increasing use of textual data in economics, finance, and management research, domain lexicons have become an important foundation for transforming unstructured text into analyzable variables. The core of text analysis is not merely to count word frequencies, but to establish mappings between textual expressions and target concepts according to the research context (Gentzkow et al., 2019). In financial text analysis, Loughran and McDonald (2011) showed that many words classified as negative in general-purpose sentiment dictionaries do not carry the same sentiment in corporate financial disclosures, leading them to develop a domain-specific dictionary for financial contexts. Subsequent research further demonstrated that lexical meaning is jointly shaped by text type, language environment, and professional domain, and that the content of a domain lexicon directly affects the scope of information captured by text-based measures (Loughran & McDonald, 2020). This issue is particularly salient for non-English texts. Du et al. (2022) constructed a Chinese financial sentiment lexicon from large-scale Chinese financial news and showed that direct translation of English financial dictionaries does not adequately accommodate Chinese financial contexts. Ma et al. (2025) further developed a fine-grained sentiment lexicon based on the language used by Chinese investors, again demonstrating the need to identify effective lexical items within the target context. Meanwhile, machine learning has also been used to extract economic information directly from large-scale Chinese financial texts, further extending the use of text as research data (Zhou et al., 2024).

Earlier approaches to domain lexicon construction typically followed a knowledge-driven strategy. A set of core terms was first identified based on theory, existing dictionaries, or expert knowledge, and then expanded through synonym relations, word-embedding similarity, or manual procedures to form a more comprehensive domain vocabulary. These approaches provide a clear conceptual starting point, and the relationship between generated terms and the research construct is relatively easy to interpret. They therefore remain widely used in fields such as finance and management, where interpretability of text-based measures is important. The development of word-embedding techniques has further automated seed-based expansion by enabling candidate terms to be retrieved according to their distributional similarity to predefined core terms. Du et al. (2022), for example, used manually identified financial sentiment terms as seeds and expanded a Chinese financial lexicon with Word2Vec. In the field of data assets, He et al. (2024) similarly combined predefined core terms, policy and regulatory corpora, and Word2Vec-based semantic expansion to construct a data-asset-related lexicon, which was subsequently used to measure corporate data assetization in annual reports. Such approaches can effectively incorporate existing domain knowledge into lexicon construction, but the direction of candidate search remains directly linked to the initial seeds. When domain concepts evolve rapidly or newly emerging terms have more complex semantic relationships with existing core terms, lexicon coverage may therefore be constrained by the initial vocabulary.

In contrast to knowledge-driven lexicon expansion, Automatic Term Extraction (ATE) identifies domain-specific words or multiword expressions directly from target corpora. Traditional ATE generally consists of candidate generation and termhood assessment, using information such as term frequency, part-of-speech patterns, co-occurrence relations, nested structures, and contextual distributions to estimate the domain relevance of candidate terms. C-value/NC-value is a representative statistical approach that evaluates multiword terms based on candidate frequency, term length, and nesting within longer terms (Frantzi et al., 2000). Statistical information, linguistic rules, and machine learning have since been increasingly integrated into hybrid term extraction frameworks. Rigouts Terryn et al. (2020) developed manually annotated ATE evaluation datasets across multiple languages and domains and showed that the boundary between terms and general language is itself subject to annotator variation. Nugumanova et al. (2022) employed non-negative matrix factorization to estimate candidate-term importance from domain corpora, further extending the use of unsupervised statistical methods in term extraction. Recent systematic reviews indicate that term frequency, TF-IDF, part-of-speech

information, contextual relations, and statistical association measures remain important sources of information for ATE and are frequently combined with machine learning models (Di Nunzio et al., 2023; Xu et al., 2025).

The development of neural networks and pretrained language models has further transformed ATE. Some studies formulate term extraction as a sequence-labeling task, allowing models to identify term boundaries directly from text and thereby reducing reliance on manually designed statistical features. Tran et al. (2024) investigated cross-domain and cross-lingual term extraction using Transformer-based models and incorporated nested-term labeling into the model design, showing that domain shift, language transfer, and term nesting all affect automatic extraction performance. Panoutsopoulos et al. (2024) compared different BERT fine-tuning strategies in the agricultural domain and found that domain-specific pretraining and fine-tuning configurations influence the ability to identify specialized terms. With the adoption of generative large language models in terminology research, ATE has further expanded from supervised sequence labeling to in-context learning and prompt-based extraction. Tran et al. (2025) proposed LlamATE to examine how domain information and cross-domain examples affect term identification with generative language models. Chun et al. (2025) further improved the extraction of specialized terms and their boundaries by selecting few-shot demonstrations through syntactic retrieval. Recent reviews have accordingly incorporated Transformer-based sequence models, transfer learning, zero- and few-shot approaches, and generative large language models into the contemporary ATE methodological landscape (Tran et al., 2026).

Domain lexicon construction has therefore developed along two relatively distinct methodological paths. Knowledge-driven approaches begin with existing domain knowledge and determine the scope of the lexicon through seed terms, semantic relations, and expert judgment. They offer strong interpretability and conceptual control, but candidate discovery is constrained by the scope of the initial knowledge base. Corpus-driven ATE, by contrast, identifies specialized expressions directly from target texts. Its technical approaches have evolved from statistical scoring and hybrid features to Transformer-based models and large language models, offering broader possibilities for candidate coverage and automatic identification while remaining sensitive to training data, domain transfer, term boundaries, and candidate noise (Xu et al., 2025; Tran et al., 2026). In practical terminology work, automatic extraction commonly generates candidate terms that are subsequently reviewed and confirmed by domain specialists. Wissik (2025), in a survey of institutional terminology practices, likewise showed that the practical application of ATE continues to depend on factors such as language support, text sources, and workflow design. Taken together, these studies indicate that different methodological approaches exhibit distinct characteristics in candidate discovery scope, term identification mechanisms, and domain adaptation.

2.2 Chinese Domain Term Discovery and Compositional Characteristics

The identification of Chinese domain terms is first affected by lexical boundaries. Unlike languages with explicit whitespace delimiters, continuous Chinese text does not directly mark word boundaries, and the same character sequence may be segmented differently depending on context and analytical granularity. Research on Chinese word segmentation has long treated boundary identification, ambiguity resolution, and out-of-vocabulary word discovery as core problems. These issues become particularly pronounced in specialized domains, where general-purpose dictionaries and existing training data may fail to cover newly emerging expressions adequately (Gao et al., 2005; Li & Sun, 2009). Recent studies of domain-specific Chinese texts continue to encounter this problem. For example, Chinese financial texts contain numerous complex entities with specialized meanings, and segmentation results and lexical boundaries can further affect downstream entity recognition. Studies in the financial domain have therefore begun to incorporate domain-specific lexical information, contextual representations, and nested structures when identifying specialized expressions (Zhang et al., 2024). Research in other Chinese professional domains similarly shows that distributional differences between general lexical resources and target corpora can affect the recognition of specialized expressions, indicating that domain

vocabulary and boundary information remain components that require explicit treatment in Chinese text processing (Xiao et al., 2025).

The structure of specialized terms further complicates this boundary problem. Domain concepts do not always correspond to a single lexical item already identified by a segmentation tool, but may instead be expressed through combinations of two or more adjacent lexical units. Individual components may occur independently while also combining with additional elements to form longer terms, giving rise to multiword terms, nested terms, and boundary variants at different levels of granularity. C-value/NC-value was among the earlier approaches to incorporate term length, occurrence frequency, and nesting within longer candidates into multiword term evaluation, demonstrating that term length and nested structure themselves constitute important information for automatic term extraction (Frantzi et al., 2000). More recent ATE research has further addressed such structural complexity. Tran et al. (2024), for example, incorporated nested-term labeling into cross-domain term extraction and showed that term nesting, domain variation, and language transfer all affect extraction performance. Similar structural issues also arise in Chinese specialized entity recognition. Zhang et al. (2024) explicitly addressed nested entities in Chinese financial texts, while Li et al. (2020) used a lattice structure composed of characters and potential words to preserve lexical spans and positional relationships. For specialized Chinese texts, the internal order, adjacency, and start–end boundaries of candidate expressions therefore cannot be fully predetermined by general-purpose word segmentation alone.

Statistical distributions within the corpus provide another source of information for identifying such combinations and boundaries. Term frequency and document frequency indicate the extent to which a candidate expression occurs and is distributed across the target corpus, whereas contextual variation can help characterize the stability of candidate boundaries. Early research on Chinese word segmentation already used left and right contextual entropy to identify potential lexical boundaries in character sequences. Jin and Tanaka-Ishii (2006) further applied branching entropy to unsupervised Chinese word segmentation, using changes in the distributions of left and right contexts around contiguous sequences to infer potential boundaries. Related studies suggest that when a sequence can occur independently across a relatively diverse range of left and right contexts, its neighboring distributions contain information useful for boundary identification (Huang & Powers, 2003; Jin & Tanaka-Ishii, 2006). In automatic term extraction, frequency, part-of-speech information, statistical association, contextual distributions, and nesting relations have likewise continued to be used in candidate generation or termhood assessment (Frantzi et al., 2000; Xu et al., 2025). These statistical features cannot directly determine whether a combination carries domain-specific meaning, but they provide computable evidence regarding whether a candidate has sufficient corpus support and relatively stable structural boundaries.

Chinese specialized vocabulary is also characterized by strong domain dependence and dynamic expansion. Existing research on Chinese financial texts has shown that general-purpose or cross-lingual lexical resources cannot fully cover specialized expressions in target contexts (Du et al., 2022; Ma et al., 2025). In rapidly evolving institutional and technological domains such as data assets, new concepts continuously enter policy texts, causing the composition and boundaries of specialized vocabulary to expand together with the corpus environment. Chinese domain term identification therefore involves not only resolving the boundaries of existing lexical items, but also discovering newly formed specialized expressions from the target corpus.

Overall, Chinese domain term identification involves not only the boundary detection of individual lexical items, but also contiguous combinations of multiple lexical units, nested structures, and expressions of different lengths. Term frequency, contextual distributions, part-of-speech information, and nesting relations describe candidate characteristics from different perspectives, while the formation of specialized terms remains dependent on specific domains and contexts. For specialized texts whose terminology is still evolving, how to discover combinations with potential domain relevance directly from the corpus, while managing the relationship between candidate scope and term boundaries, remains an important issue in domain lexicon construction.

2.3 Research Gap and Study Positioning

Knowledge-driven domain lexicons can use established conceptual systems to control the scope of the vocabulary, but candidate discovery is consequently constrained by the boundaries of the initial knowledge base. Existing studies on Chinese domain lexicons have shown that specialized vocabulary exhibits clear linguistic and contextual specificity (Du et al., 2022; Ma et al., 2025). In the field of data assets, He et al. (2024) improved the corpus basis of lexicon construction by combining predefined core terms with Word2Vec-based expansion, but candidate search still begins from a predefined set of core terms. For emerging domains in which terminology continues to evolve, reducing the dependence of candidate discovery on initial seeds therefore remains an issue that warrants further investigation.

Corpus-driven automatic term extraction provides an alternative route for expanding candidate discovery. From statistical methods such as C-value to unsupervised semantic modeling, Transformer-based approaches, and large language models, existing studies have increasingly used frequency, context, nesting relations, and deep semantic representations to identify specialized terms directly from domain corpora (Frantzi et al., 2000; Nugumanova et al., 2022; Tran et al., 2024; Xu et al., 2025). As candidate length and structural complexity increase, however, term discovery involves both coverage and candidate-space size. Looser candidate generation can reduce the likelihood that potential terms are excluded at an early stage, but it also produces more candidates that require scoring, classification, or manual judgment. Stricter generation rules can reduce the processing scale, but may prematurely exclude true terms. This issue is particularly relevant for specialized expressions composed of multiple contiguous lexical units, where term length, nesting relations, and boundary variation all influence candidate formation (Rigouts Terryn et al., 2020; Tran et al., 2024). Compared with the extensive attention given to final term extraction performance, the compression of the candidate search space during candidate generation and its trade-off with true-term coverage have received relatively limited attention in existing ATE evaluation (Xu et al., 2025; Tran et al., 2026).

Further advances in automatic identification have not eliminated the role of domain semantic judgment. Statistical features can characterize candidate frequency, distributional scope, and contextual boundaries, while pretrained language models and large language models can exploit richer semantic representations for term identification. Nevertheless, domain relevance, lexical completeness, and conceptual granularity remain closely tied to specific professional contexts (Xu et al., 2025; Tran et al., 2025). Research on practical terminology workflows also shows that automatic term extraction is commonly used together with expert review, confirmation, and terminology management, indicating a clear functional division between automatic discovery and human judgment (Wissik, 2025). This issue is particularly relevant in evolving Chinese policy domains such as data assets: candidate terms require sufficient opportunity to be discovered from the target corpus, but their status as independent and complete domain terms must still be determined according to domain semantics.

Based on these observations, this study focuses on the discovery of contiguous lexical-unit combinations in Chinese specialized texts and treats candidate search and domain semantic validation as two connected stages of domain lexicon construction. The central question is whether the candidate search space can be reduced while maintaining high coverage of true terms, and whether a final lexicon can subsequently be formed through domain semantic validation. To address this question, this study develops the Recursive Lexical Combination method and evaluates its candidate coverage, candidate compression, and final term identification performance using separate Validation and held-out Test sets.

3 Methodology

3.1 Task Definition and Overall Framework

The core task of domain lexicon construction is to identify terms with stable semantic boundaries and clear domain relevance from a given domain corpus and to organize them into a structured lexical set for subsequent text analysis. In specialized domains where terminology continues to evolve, direct reliance on predefined seed terms may constrain

candidate discovery to the scope of existing knowledge, whereas exhaustive enumeration of all contiguous lexical-unit combinations can produce a rapidly expanding candidate space. Accordingly, this study divides domain lexicon construction into two connected stages: candidate discovery and semantic validation. The first stage exploits compositional structures within the corpus to generate a high-coverage yet size-controlled candidate set, while the second determines the final lexicon through domain semantic validation.

Let the domain corpus be

$$\mathcal{D} = \{d_1, d_2, \dots, d_N\}$$

where d_i denotes the i-th domain document. After structure-preserving preprocessing and word segmentation, each document is represented as an ordered sequence of lexical units. In this study, a lexical unit refers to the basic computational unit obtained after text preprocessing and segmentation and used for recursive combination; it is not equivalent to a morpheme in the strict linguistic sense. Based on these lexical units, all combinations satisfying the contiguity and boundary constraints within a specified length range can, in principle, be enumerated to form the Direct Candidate Universe, denoted by $(\mathcal{C}_D)$. As candidate length increases, direct enumeration produces a rapidly growing number of combinations, many of which have low corpus support or lack stable domain meaning, thereby increasing both computational burden and the scope of subsequent manual review.

To control this candidate space, this study proposes Recursive Lexical Combination (RLC). Rather than generating all higher-order contiguous combinations in advance, RLC expands lexical-unit sequences progressively from shorter to longer combinations. A current combination is extended one unit to the right only when it has sufficient statistical support in the corpus; otherwise, the corresponding recursive path is terminated immediately. The resulting RLC candidate pool is denoted by

$$\mathcal{C}_{RLC} \subseteq \mathcal{C}_D$$

This recursive search preserves the correspondence between candidate terms and their original text positions, combination order, and generation paths, while avoiding further enumeration of higher-order combinations that clearly lack sufficient corpus support. At the candidate generation stage, RLC is primarily designed to reduce the candidate search space requiring further evaluation while retaining as many true domain terms as possible. Final term identification is subsequently performed through automatic output rules and domain semantic validation.

After candidate discovery, statistical rules and domain semantic judgment serve different functions. Candidates in $\mathcal{C}_{RLC}$ can be further processed using features such as term frequency, document frequency, left and right branching entropy, part-of-speech information, and structural boundaries to generate automatic term outputs for evaluating the identification capability of the statistical rules themselves. In parallel, the RLC candidate pool undergoes expert semantic validation, in which reviewers determine whether each candidate should be included in the final lexicon according to domain relevance, lexical completeness, and semantic boundaries. The final domain lexicon can be expressed as

$$\mathcal{L}_{\text{final}} = \{c \in \mathcal{C}_{RLC} \mid J(c) = 1\}$$

where $J(c)$ denotes the final retention decision for candidate c after domain semantic validation. Under this design, the automatic procedure is primarily responsible for high-coverage candidate discovery and search-space control, whereas expert validation addresses semantic relevance, boundary completeness, and conceptual granularity that cannot be fully resolved by statistical rules.

The overall methodological framework is illustrated in Fig. 1. Raw domain texts first undergo structure-preserving preprocessing and are converted into ordered lexical units. RLC then generates candidate combinations layer by layer under support constraints and terminates recursion early along low-support paths, thereby producing a size-controlled candidate pool. On this basis, unified automatic output rules can be applied to obtain Automatic RLC results for comparable evaluation against other automatic term extraction methods. In parallel, the RLC candidate pool undergoes

expert semantic validation to form the final domain lexicon. By separating candidate discovery, search-space compression, and semantic validation, the framework retains the corpus-driven capability for automatic discovery while providing explicit generation paths and human auditability.

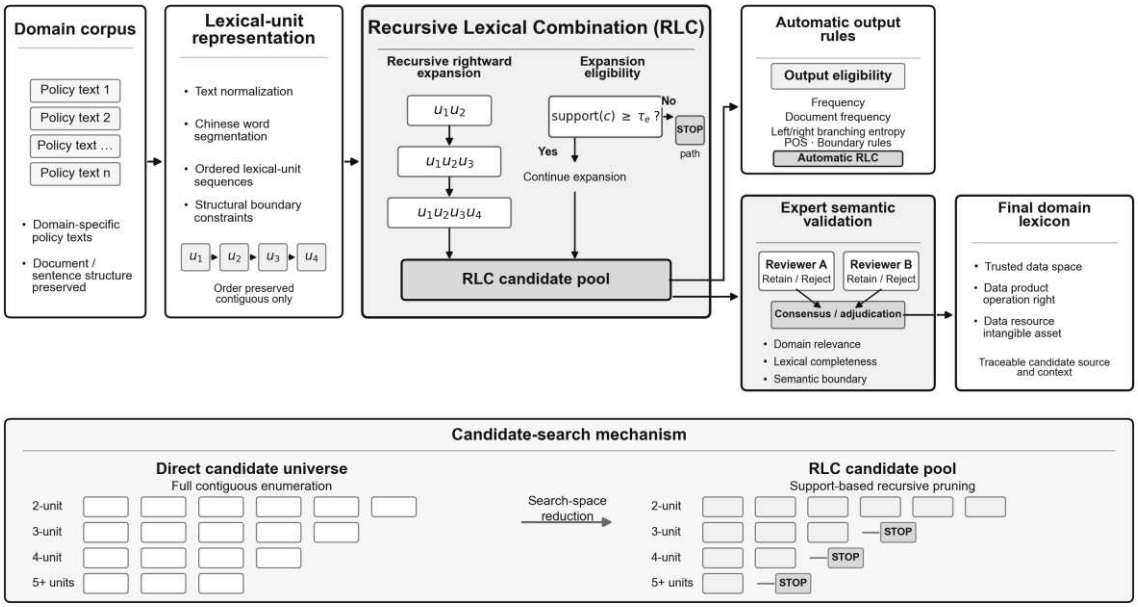


Fig. 1 RLC-based human-in-the-loop domain lexicon construction framework

3.2 Lexical Unit Representation and Boundary Constraints

The recursive combination process of RLC is built on ordered lexical-unit sequences derived from the original text. In this study, the basic computational objects obtained after Chinese word segmentation are defined as lexical units. A lexical unit is the basic computational unit used for recursive combination after text preprocessing and segmentation. It may consist of a single Chinese character or a multi-character word and is not equivalent to a morpheme in the strict linguistic sense. Using lexical units as the basic representation avoids imposing overly strong assumptions on the internal morphological structure of Chinese while preserving the structural property that specialized terms can be progressively formed through combinations of adjacent lexical units.

For any document in the domain corpus $(\mathcal{D})$, text normalization and Chinese word segmentation are first performed while preserving structural boundaries defined by documents, paragraphs, sentences, and punctuation. Let the m-th valid text segment after preprocessing be represented as:

$$r_m = (u_1, u_2, \dots, u_{L_m}),$$

where r_m denotes a text segment within which contiguous combinations are permitted, u_j denotes the lexical unit at position (j), and L_m is the number of lexical units contained in that segment. Valid text segments are determined by the original textual structure. Candidate combinations can be generated only within the same segment and are not allowed to cross sentence-ending markers, predefined punctuation boundaries, or other structural boundaries. This constraint ensures that every generated candidate corresponds to a local word sequence that actually occurs in the source text, preventing lexical units from different contexts from being recombined into sequences that do not occur in the original text.

On this basis, a candidate sequence beginning at position (i) and consisting of (k) contiguous lexical units is defined as:

$$c_{i,k} = (u_i, u_{i+1}, \dots, u_{i+k-1}), \qquad 1 \le i \le L_m - k + 1$$

Candidate formation must satisfy both contiguity and order preservation. Contiguity requires the lexical units constituting a candidate to be directly adjacent in the original text, without skipping any intervening component. Order preservation requires the combined sequence to retain exactly the same lexical-unit order as in the source text. For example, suppose a text segment is segmented as

[数据 | 基础设施 | 安全 | 标准]

Adjacent lexical units can then form contiguous combinations such as “数据基础设施” (data infrastructure), “基础设施安全” (infrastructure security), “安全标准” (security standard), and “数据基础设施安全” (data infrastructure security), which may be further extended into longer contiguous candidates. In contrast, “数据” (data) cannot be directly combined with the nonadjacent “安全” (security), nor can the original order of the lexical units be altered. Likewise, lexical units located in different sentences or separated by a prohibited structural boundary cannot form a valid candidate.

This representation provides each candidate with explicit textual provenance and structural information. For any candidate $c_{i,k}$, its constituent lexical units, starting position, combination length, and corresponding source-text segment can all be traced. Compared with an unordered set-of-words representation, ordered lexical-unit sequences preserve the local adjacency relations on which specialized term formation depends. Compared with direct character-level combination, segmentation-based lexical units also reduce a large number of character combinations that lack an independent semantic basis, thereby providing a more stable representation for subsequent candidate discovery. After this processing, each valid text segment is transformed into an ordered lexical-unit sequence with explicit structural boundaries and serves as input to RLC-based recursive candidate discovery.

3.3 Support-Based Recursive Candidate Discovery

After lexical-unit representation and structural boundary segmentation, candidate discovery can be formulated as a search over contiguous combinations in ordered lexical-unit sequences. If all contiguous combinations up to a specified maximum length are directly enumerated, the number of candidates increases rapidly with combination length, while many higher-order combinations lack sufficient corpus support. RLC adopts a rightward recursive expansion strategy from shorter to longer combinations, using the corpus support of the current combination to determine whether the corresponding path should continue. Low-support paths are therefore terminated before exhaustive enumeration. This mechanism draws on the anti-monotonic pruning principle of Apriori, while the search objects remain ordered contiguous sequences from the original text rather than unordered itemsets.

Let $\mathcal{R}$ denote the set of all valid text segments. For a candidate c, its corpus frequency is defined as:

$$f(c) = \sum_{r_m \in \mathcal{R}} \sum_i \mathbf{1}(c_{i,k} = c)$$

where $\mathbf{1}(\cdot)$ is an indicator function that equals 1 when the sequence at the corresponding position in a text segment forms candidate c, and 0 otherwise. Thus, $f(c)$ represents the total number of contiguous occurrences of candidate c in the corpus and serves as the support measure for determining whether a recursive path is eligible for further expansion.

RLC first generates all valid 2-unit candidates satisfying the contiguity and boundary constraints defined in Section 3.2, forming the initial candidate layer $\mathcal{C}_{RLC}^{(2)}$. Accordingly,

$$\mathcal{C}_{RLC}^{(2)} = \mathcal{C}_D^{(2)}$$

where $\mathcal{C}_D^{(2)}$ denotes the complete set of valid 2-unit candidates obtained through direct contiguous enumeration. This design does not compress the search space at the shortest compound-candidate layer; instead, support-based pruning is applied only to subsequent higher-order expansion.

For any candidate of length k,

$$c_{i,k} = (u_i, u_{i+1}, \dots, u_{i+k-1})$$

if a lexical unit u_{i+k} exists immediately to its right within the same text segment, its unique one-step rightward extension is defined as:

$$c_{i,k+1} = c_{i,k} \oplus u_{i+k} = (u_i, u_{i+1}, \dots, u_{i+k})$$

where $\oplus$ denotes contiguous concatenation that preserves the original lexical-unit order. Because every occurrence of a longer candidate necessarily contains an occurrence of its current prefix, the following relationship holds along the same rightward expansion path:

$$f(c_{i,k+1}) \leq f(c_{i,k})$$

This monotonic relationship provides the basis for support-based pruning. Let τ_e denote the minimum support threshold for recursive expansion and $K_{\max}$ the maximum permitted candidate length. The expansion eligibility of the current candidate is defined as:

$$E(c_{i,k}) = \mathbf{1}[f(c_{i,k}) \geq \tau_e \ \wedge \ k < K_{\max} \ \wedge \ u_{i+k} \text{ exists}]$$

When $E(c_{i,k}) = 1$, the current path continues to the right and generates $c_{i,k+1}$. When $E(c_{i,k}) = 0$, the path terminates. In particular, if

$$f(c_{i,k}) < \tau_e$$

then, according to the frequency monotonicity above, no further rightward extension of this candidate can recover the same support threshold. The corresponding branch therefore does not need to be enumerated further. RLC consequently concentrates the search on compositional paths that retain sufficient corpus support.

Candidate generation and continued candidate expansion represent two distinct states. A newly generated $k+1$-unit combination first enters the RLC candidate space, whereas its eligibility to generate a $k+2$-unit combination is determined only after its own support has been recalculated. Support at this stage therefore controls the continuation of the search path rather than whether the candidate qualifies as a complete domain term. Term frequency, document frequency, left and right branching entropy, part-of-speech information, and other conditions used to determine candidate output eligibility are defined separately in Section 3.4.

After recursive expansion across successive layers, the RLC candidate pool can be expressed as

$$\mathcal{C}_{RLC} = \cup_{k=2}^{K_{\max}} \mathcal{C}_{RLC}^{(k)}$$

where $\mathcal{C}_{RLC}^{(k)}$ denotes the set of k-unit candidates actually reached through valid recursive paths. Unlike Direct enumeration, which generates all valid contiguous combinations at each candidate length, RLC generates the next candidate layer only when the preceding path satisfies the expansion condition. As candidate length increases, terminated paths accumulate and the candidate space progressively contracts. Because the complete 2-unit layer is generated initially, support-based pruning affects only subsequent higher-order expansion; once a low-support path is terminated, its corresponding longer combinations are no longer enumerated.

The RLC candidate discovery procedure is summarized in Algorithm 1.

Algorithm 1 Recursive Lexical Combination candidate discovery

Input:	Valid lexical-unit sequences R Minimum expansion support τe Maximum candidate length Kmax
Output:	RLC candidate pool CRLC
1:	Generate all valid contiguous 2-unit candidates C(2)
2:	CRLC ← C(2)
3:	Current ← C(2)

4:	for k = 2 to Kmax − 1 do
5:	Next ← ∅
6:	for each candidate c in Current do
7:	compute corpus frequency f(c)
8:	if f(c) ≥ τe and a right-adjacent unit exists then
9:	c+ ← append the adjacent unit to c
10:	add c+ to CRLC
11:	add c+ to Next
12:	end if
13:	end for
14:	Current ← unique candidates in Next
15:	if Current is empty then break
16:	end for
17:	return CRLC

The recursive procedure determines whether a path should continue solely on the basis of corpus support; neither the Gold Standard nor domain labels are used during candidate generation. The original positions, lexical-unit order, and expansion relationships of candidates remain unchanged throughout recursion, allowing higher-order candidates to be traced back to their source texts and successive generation paths. The resulting RLC candidate set therefore satisfies the contiguity, boundary, and support conditions and is subsequently subjected to automatic output determination and expert semantic validation.

3.4 Candidate Eligibility and Automatic Output Rules

The RLC candidate pool C_{RLC} obtained through recursive expansion contains all candidate combinations actually generated along valid recursive paths. Inclusion in this set indicates only that a candidate satisfies the contiguity and recursive generation conditions; it does not directly determine whether the candidate is included in the automatic term output. RLC treats recursive expansion and term output as separate decisions. Expansion eligibility $E(c)$ is controlled by the corpus support and combination length defined in Section 3.3, whereas automatic output is determined by the statistical characteristics and lexical structure of the candidate itself.

The corpus frequency $f(c)$ of candidate c has already been defined in Section 3.3. On this basis, document frequency is further calculated as the number of documents in which the candidate occurs at least once. Let the domain corpus contain N documents. The document frequency of candidate c is defined as:

$$df(c) = \sum_{j=1}^{N} \mathbf{1}\left(c \in d_j\right),$$

where $\mathbf{1}(\cdot)$ is an indicator function that equals 1 when candidate c occurs in document d_j, and 0 otherwise. Corpus frequency records the cumulative number of occurrences of a candidate across the entire corpus, whereas document frequency records the number of documents over which it is distributed. The two measures characterize candidate properties in terms of occurrence intensity and cross-document distribution, respectively.

Candidate boundaries are characterized using the left and right adjacent contexts. Let $\mathcal{L}(c)$ and $\mathcal{R}(c)$ denote the sets of left- and right-adjacent lexical units observed when candidate c occurs in valid text segments. For a left-adjacent lexical unit x, its conditional probability is defined as:

$$p_L(x \mid c) = \frac{n_L(x,c)}{\sum_{x' \in \mathcal{L}(c)} n_L(x',c)},$$

where $n_L(x,c)$ denotes the number of times lexical unit x occurs immediately to the left of candidate c. The left branching entropy of candidate c is defined as:

$$H_L(c) = -\sum_{x \in \mathcal{L}(c)} p_L(x \mid c) \log p_L(x \mid c).$$

Similarly, let $n_R(y,c)$ denote the number of times lexical unit y occurs immediately to the right of candidate c. Its conditional probability is

$$p_R(y \mid c) = \frac{n_R(y,c)}{\sum_{y' \in \mathcal{R}(c)} n_R(y',c)},$$

and the right branching entropy is defined as

$$H_R(c) = -\sum_{y \in \mathcal{R}(c)} p_R(y \mid c) \log p_R(y \mid c).$$

Left and right branching entropy characterize the distribution of a candidate across its adjacent contexts. When a candidate repeatedly co-occurs with only a small number of fixed components, the corresponding contextual probability distribution is more concentrated and its branching entropy is lower. When a candidate occurs with a more diverse range of contextual components, its adjacent distribution is more dispersed and the corresponding branching entropy increases. The two sides are calculated separately so that the preceding and following boundaries of a candidate are characterized independently.

In addition to frequency and contextual distributions, automatic output also incorporates part-of-speech and structural-boundary constraints. Based on the part-of-speech labels obtained after word segmentation, a part-of-speech eligibility function $P(c)$ is defined. $P(c) = 1$ when the part-of-speech structure of candidate c satisfies the predefined term-output rules, and $P(c) = 0$ otherwise. A boundary eligibility function $B(c)$ is also defined to indicate whether the candidate satisfies the requirements of lexical-unit contiguity, original order, and predefined textual boundaries:

$$P(c), B(c) \in \{0,1\}.$$

Let τ_f, τ_d, τ_L, and τ_R denote the minimum thresholds for corpus frequency, document frequency, left branching entropy, and right branching entropy, respectively. The automatic output eligibility of candidate c is then defined as:

$$O(c) = \mathbf{1}\left[f(c) \geq \tau_f \wedge df(c) \geq \tau_d \wedge H_L(c) \geq \tau_L \wedge H_R(c) \geq \tau_R \wedge P(c) = 1 \wedge B(c) = 1\right].$$

When $O(c) = 1$, the candidate is included in the Automatic RLC output set. Otherwise, its generation record remains in the RLC candidate pool, but it is not included in the automatic term output. The Automatic RLC output can therefore be expressed as:

$$\mathcal{C}_{Auto} = \{c \in \mathcal{C}_{RLC} \mid O(c) = 1\}.$$

The output decision does not retroactively alter the recursive path. Whether a candidate continues to expand remains determined solely by the expansion eligibility defined in Section 3.3. Automatic RLC therefore represents only the automatic output obtained by applying statistical and structural rules to the RLC candidate pool. The specific thresholds are determined in Section 4.

3.5 Expert Semantic Validation and Lexicon Formation

After recursive generation, the RLC candidate pool enters the expert semantic validation stage. Expert validation is conducted on the complete RLC candidate set $\mathcal{C}_{RLC}$, rather than being restricted to the Automatic RLC output set $\mathcal{C}_{Auto}$ obtained in Section 3.4. The automatic output rules generate comparable term extraction results based on statistical characteristics, whereas expert semantic validation directly assesses whether each RLC candidate is suitable for inclusion in the final lexicon as an independent domain term.

Candidate validation is primarily based on domain relevance, lexical completeness, and semantic boundaries. Domain relevance requires a candidate to refer to a concept, object, activity, institutional arrangement, or technical element within the target domain. Lexical completeness requires the candidate to function as a relatively complete term rather than as a partial component of a longer expression. Semantic boundaries require the scope of the candidate combination to correspond approximately to the conceptual scope that it actually expresses. General policy expressions, organization names, overly broad generic expressions, and combinations with incomplete boundaries are not retained as final domain terms.

During validation, two reviewers independently assess each candidate. Let the decision functions of Reviewer A and Reviewer B for candidate c be:

$$J_A(c), J_B(c) \in \{0,1\},$$

where 1 denotes retention and 0 denotes rejection. Reviewers are provided with each candidate and the necessary context from the original text to assess its meaning and boundaries in actual usage, but they are not given access to the Gold Standard or any Gold-based evaluation results. Statistical characteristics of the candidate may be retained as provenance information, but the final semantic decision is not directly determined by automatic evaluation results.

When the two reviewers agree, their shared decision is adopted as the validation result for the candidate. When their judgments differ, the candidate enters a disagreement-resolution stage, in which a final decision is reached through consensus discussion or a predefined adjudication procedure. Let the final validation function after disagreement resolution be $J(c)$, defined as:

$$J(c) = \begin{cases} J_A(c), & J_A(c) = J_B(c), \\ J_C(c), & J_A(c) \neq J_B(c), \end{cases}$$

where $J_C(c)$ denotes the final decision reached for a disputed candidate through consensus or adjudication.

After all candidates have been reviewed, the final domain lexicon consists of all candidates with a retention decision:

$$\mathcal{L}_{final} = \{c \in \mathcal{C}_{RLC} \mid J(c) = 1\}.$$

For each retained candidate, its constituent lexical units, source text, and contextual information are also recorded to preserve traceability between the final domain lexicon and the RLC candidate generation process.

4 Experimental Design

4.1 Corpus, Data Split, and Gold Standard

This study uses national-level policy texts related to data assets as the experimental corpus. Policy documents provide a relatively standardized linguistic environment and contain a concentrated range of specialized expressions concerning data resources, data elements, data governance, data circulation, data security, and data infrastructure, thereby providing a stable textual setting for the compositional discovery of Chinese domain terms. The corpus is drawn from the policy documents included in the *Compilation of National-Level Data Asset Policies (January 2024–January 2025)* and contains 32 national-level policies related to data assets. Only the main policy texts are retained for the experiments; auxiliary materials such as the compilation preface, table of contents, and editorial notes are removed. Text normalization, structural boundary identification, and lexical-unit representation are then performed according to the procedures described in Section 3. The processed full corpus contains 186,751 Chinese characters and 3,639 sentences.

To separate method development from final evaluation, the corpus is partitioned in advance into a Validation set and a held-out Test set at the document level. The Validation set consists of two policy documents, P09 and P21, containing 9,441 characters and 156 sentences, and is used primarily for parameter selection and method configuration. The Test set consists of six policy documents, P04, P07, P12, P16, P27, and P29, containing 40,423 characters and 789 sentences, and is reserved for final evaluation after the configuration has been fixed. The remaining 24 documents are retained in the full corpus but are not included in the Validation or Test evaluation conducted in this study. During the experimental procedure, method configuration is first determined on the Validation set and is then kept fixed when applied to the Test set, thereby separating final testing from prior parameter selection.

To evaluate the ability of different methods to identify domain terms, Gold Standards are constructed for both the Validation and Test documents. Annotation is conducted on the original policy texts by identifying terms with clear domain-specific meaning and relatively complete semantic boundaries throughout each document. After the initial annotation, a second review is performed to correct omissions and boundary-related errors. According to the correspondence between the annotated terms and the lexical units defined in Section 3, the Gold Standard is further divided into Gold_all and Gold_multi. Gold_all retains all manually confirmed terms, including single-lexical-unit terms, multi-lexical-unit terms, and partially aligned terms. Gold_multi, by contrast, contains only terms composed of at least two fully and contiguously aligned lexical units. Because the central objective of RLC is to discover combinations of contiguous lexical units, subsequent comparisons among automatic methods, candidate coverage analysis, and the main Precision, Recall, and F1 evaluations all use Gold_multi as the primary reference set.

The same Gold Standard can be described at both the occurrence and type levels. A Gold occurrence denotes each actual annotated term span in the original text and therefore reflects the overall density of annotated terminology in the corpus. A corpus unique type denotes a normalized term counted only once within a given data split. Because the same term may occur in multiple policy documents and each document constitutes a separate evaluation environment, this study ultimately adopts the document-term type as the primary evaluation unit. Specifically, the same normalized term is counted only once within a document, but the same term occurring in different documents is counted separately. This evaluation scheme prevents a small number of frequently repeated terms from disproportionately influencing the results while preserving method performance across different documents.

Under this evaluation scheme, the Validation set contains 247 Gold_all occurrences, of which 209 belong to Gold_multi. Gold_multi contains 122 unique term types after deduplication across the Validation corpus, corresponding to 126 document-term types. The Test set contains 2,305 Gold_all occurrences, of which 1,886 belong to Gold_multi. Gold_multi contains 596 unique term types after deduplication across the Test corpus, corresponding to 658 document-term types. In all subsequent experiments, TP, FP, FN, and Candidate Pool Recall are evaluated against the Gold_multi document-term types in the corresponding data split. The principal statistics of the corpus partition and Gold Standard are reported in Table 1.

Table 1 Corpus, data split, and Gold Standard statistics

Scope	Documents	Characters	Sentences	Gold_all occurrences	Gold_multi occurrences	Gold_multi unique types	Gold_multi document-term types
Full corpus	32	186,751	3,639	—	—	—	—
Validation	2	9,441	156	247	209	122	126
Test	6	40,423	789	2,305	1,886	596	658

Note: Gold occurrence denotes each annotated term span in the original text; unique type denotes a normalized term counted once within the corresponding split; document-term type denotes a normalized term counted once within each document and separately across documents.

4.2 Comparison Methods

To evaluate the role of RLC in domain term discovery, this study establishes a comparison framework comprising predefined lexicons, statistical term extraction, and compositional candidate generation methods. The comparison methods cover two representative methodological paths: one determines the scope of terms directly from existing domain knowledge, whereas the other discovers candidate terms automatically from the target corpus. In addition, Direct n-gram is compared directly with Automatic RLC to distinguish the effect of the recursive candidate discovery mechanism from that of the subsequent automatic filtering rules. The complete evaluation includes five methods or processing frameworks: Seed-based Lexicon, C-value, Direct n-gram, Automatic RLC, and RLC + Expert Consensus.

The Seed-based Lexicon represents the conventional lexicon-based approach in which text is identified using a predefined domain vocabulary. A fixed Seed Lexicon containing 221 unique terms is used in this study and remains unchanged across the Validation and Test sets. The method directly matches the existing lexicon against the Gold Standard and does not generate additional compositional candidates from the target texts, thereby serving as a reference for evaluating the coverage provided by a predefined vocabulary. The 221-term lexicon is a secondary reproduction of a vocabulary reported in previous publicly available research rather than an official supplementary resource provided by the original authors. It is therefore used as a fixed seed-based baseline, and subsequent analyses further examine the ability of the other methods to discover true terms that fall outside this Seed Lexicon.

C-value serves as a corpus-driven statistical term extraction baseline, providing an automatic discovery reference distinct from the fixed-lexicon approach. Unlike the Seed-based Lexicon, C-value does not restrict term identification to a predefined domain vocabulary, but instead generates term outputs from candidate combinations in the target corpus according to their statistical characteristics. C-value therefore provides a corpus-driven ATE baseline and, together with the Seed-based Lexicon, represents the corpus-driven and knowledge-driven reference approaches used in the comparison.

Direct n-gram provides the most direct candidate-generation baseline for RLC. It uses the same lexical-unit representation, contiguity requirements, and textual boundary constraints as RLC and directly generates all valid contiguous combinations within the permitted length range, without using support-based recursive pruning to terminate higher-order combination paths. The candidate space produced by Direct n-gram can therefore be regarded as the Direct Candidate Universe under the given representation and boundary rules. Compared with RLC as defined in Section 3, the primary difference lies in the candidate search process: Direct n-gram exhaustively enumerates valid contiguous combinations, whereas RLC uses the support of preceding combinations to determine whether longer combinations should continue to be generated. Their comparison is therefore used primarily to evaluate how recursive search affects candidate-space size and true-term coverage.

Automatic RLC denotes the term set obtained by applying the unified automatic output rules to the RLC candidate pool. As defined in Section 3, inclusion in the RLC candidate pool indicates only that a candidate has passed the contiguous combination and recursive generation procedures; it does not imply that the candidate has already been identified as a final domain term. Automatic RLC further applies term frequency, document frequency, left and right branching entropy, part-of-speech information, and structural boundary rules to generate automatic term outputs. To ensure that the comparison between Direct n-gram and Automatic RLC primarily reflects differences in candidate generation mechanisms, the corresponding automatic output rules use the same evaluation criteria and frozen configuration, thereby avoiding confounding effects arising from different filtering standards.

Beyond the automatic methods, this study further evaluates the complete RLC + Expert Consensus framework. Expert validation is conducted on the complete candidate pool generated by RLC rather than only on the terms retained by Automatic RLC. Reviewers determine whether each candidate is suitable for inclusion as an independent domain term

according to domain relevance, lexical completeness, and semantic boundaries, and the final lexicon is formed through independent review and subsequent disagreement resolution.

4.3 Parameter Selection and Configuration

To prevent the final test results from influencing method configuration, parameter selection and final evaluation are conducted separately on the Validation set and the held-out Test set. The Validation set is used to determine the parameter configuration required for RLC recursive candidate generation and automatic term output, whereas the Test set is used only for final performance evaluation after the configuration has been frozen. During parameter selection, the Test Gold is not used for threshold determination, rule adjustment, or method selection, thereby maintaining a clear separation between method development and final testing.

The RLC parameter configuration corresponds to two stages: recursive candidate discovery and automatic term output. The recursive stage primarily involves the minimum expansion support and maximum candidate length, which determine the scope of candidate expansion. The automatic output stage further considers term frequency, document frequency, left and right branching entropy, part-of-speech information, and related conditions to filter combinations within the RLC candidate pool. Across different configurations, lexical-unit representation, contiguity constraints, original word order, and textual boundary rules are kept unchanged. The same Gold_multi and document-term type evaluation scheme used in the final test is also adopted, ensuring a consistent basis for comparison across parameter configurations.

The final configuration is selected based on both candidate discovery and automatic term identification performance on the Validation set. Parameter selection jointly considers true-term coverage and candidate-space size to avoid either increasing coverage through excessive expansion of the candidate space or losing large numbers of true terms through overly aggressive pruning. The automatic output configuration is evaluated using Precision, Recall, and F1 to assess term filtering performance. Additional candidate constraints are also compared during the Validation stage, but these settings do not yield stable performance improvements and are therefore excluded from the final method configuration. The frozen configuration consequently adopts support-driven recursive expansion together with term frequency, document frequency, left and right branching entropy, and part-of-speech conditions to generate the Automatic RLC output.

After the Validation stage is completed, the recursive parameters of RLC, the Automatic RLC output conditions, and the experimental settings of all comparison methods are kept fixed and applied directly to the held-out Test set using the same configuration.

4.4 Evaluation Metrics and Experimental Protocol

This study evaluates the different methods from three perspectives: final term identification quality, candidate discovery capability, and candidate-space compression. Final term identification quality measures the agreement between method outputs and the Gold Standard. Candidate discovery capability assesses how many true domain terms are covered by RLC before semantic validation, whereas candidate-space compression measures the extent to which recursive search reduces the number of candidates requiring further processing relative to direct enumeration. These three dimensions correspond respectively to final identification, candidate coverage, and search efficiency in the domain lexicon construction process.

For final term identification quality, Precision, Recall, and F1 are used as the basic evaluation metrics. Let TP denote the number of Gold terms correctly identified by a method, FP the number of output terms that do not appear in the Gold Standard, and FN the number of Gold terms not identified by the method. The metrics are defined as

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For multi-document evaluation, both Micro-F1 and Macro-F1 are reported. Micro-F1 first aggregates TP, FP, and FN across all Test documents and then calculates overall Precision, Recall, and F1, thereby reflecting overall identification performance on the entire Test set. Macro-F1, by contrast, calculates F1 separately for each document and then takes the arithmetic mean, giving documents of different sizes equal weight in the overall evaluation.

Because RLC primarily serves a candidate discovery function, evaluating only the final automatic output cannot fully characterize the effect of its search mechanism. Candidate Pool Recall is therefore used to evaluate the coverage of true terms in the RLC candidate pool and is defined as

$$\text{Candidate Pool Recall} = \frac{\text{N}(Gold_{multi} \cap C_{RLC})}{\text{N}(Gold_{multi})}$$

where C_{RLC} denotes the candidate pool generated recursively by RLC, and $\text{N}(Gold_{multi} \cap C_{RLC})$ denotes the number of true multi-lexical-unit terms included in the candidate pool. Candidate Pool Recall measures the coverage upper bound at the candidate discovery stage and is not equivalent to the final Recall of Automatic RLC or expert semantic validation. A Gold term that enters the candidate pool may still be excluded by the automatic output rules or subsequent semantic judgment. Candidate coverage and final term identification are therefore reported separately in the subsequent results.

RLC's control over the search space is evaluated using Candidate Compression. Taking all valid contiguous combinations generated by Direct n-gram as the Direct Candidate Universe, let N_{Direct} denote the number of candidate types generated by Direct n-gram and N_{RLC} the number of candidate types actually generated by RLC. Candidate Compression is defined as

$$\text{Candidate Compression} = 1 - \frac{N_{RLC}}{N_{Direct}}$$

This metric represents the proportion of candidate types eliminated through support-based recursive pruning. In addition to the number of candidate types, the experiments record the actual number of candidate occurrences in the corpus to provide a complementary measure of how recursive search affects the volume of candidate processing. Runtime is also reported as an auxiliary efficiency measure, primarily to describe the computational cost of different candidate generation procedures in the current experimental environment rather than as an independent basis for statistical inference. Reporting Candidate Compression together with Candidate Pool Recall makes it possible to characterize the trade-off between candidate-space reduction and Gold coverage.

To further examine the ability of different methods to discover terms beyond the predefined lexicon, this study defines a Seed-uncovered Gold set. Let the fixed Seed Lexicon be L_{seed}. Then

$$Gold_{seed-uncovered} = Gold_{multi} \setminus L_{seed}$$

On this basis, the coverage of Seed-uncovered Gold is calculated separately for C-value, Direct n-gram, Automatic RLC, and the RLC candidate pool. This evaluation distinguishes repeated identification of terms already contained in the existing lexicon from the discovery of new terms and is used primarily to assess whether corpus-driven candidate discovery can extend the terminological scope defined by the predefined seeds.

All automatic methods use the same normalized exact-matching rule. Candidate outputs are subjected to the same text normalization procedure before being compared with the Gold Standard; no fuzzy matching or post hoc correction based on semantic similarity is applied. When the same normalized term occurs repeatedly within a single document, it is counted as one document-term type; when it occurs in different documents, it is counted separately for each document.

The same evaluation scheme is maintained in analyses by candidate length, ensuring a consistent statistical basis for comparing candidate coverage and compression across terms of different lengths.

Expert semantic validation follows a blind review protocol separate from the Gold Standard. Two reviewers independently assess RLC candidates according to domain relevance, lexical completeness, and semantic boundaries. They may refer to the necessary source-text context in which a candidate occurs, but they are not given access to the Test Gold or any associated evaluation results. Candidates on which the two reviewers disagree are resolved through consensus discussion or a predefined adjudication procedure.

5 Results

5.1 Overall Performance on the Held-out Test Set

Table 2 Held-out Test performance of different methods

Method	TP	FP	FN	Precision	Recall	Micro-F1	Macro-F1
Seed-based Lexicon	56	67	602	0.4553	0.0851	0.1434	0.1370
C-value	301	3,851	357	0.0725	0.4574	0.1252	0.1157
Direct n-gram	218	1,137	440	0.1609	0.3313	0.2166	0.2001
Automatic RLC	218	1,137	440	0.1609	0.3313	0.2166	0.2001
RLC + Expert Consensus	286	245	372	0.5386	0.4347	0.4811	0.4627

Note: *All results are evaluated on the held-out Test set using Gold_multi document-term types.*

Table 2 reports the overall identification performance of the different methods on the held-out Test set. The automatic methods exhibit clear differences in the balance between Precision and Recall. The Seed-based Lexicon achieves a relatively high Precision of 0.4553 among the automatic methods, but its Recall is only 0.0851, with Micro-F1 and Macro-F1 values of 0.1434 and 0.1370, respectively. C-value increases Recall to 0.4574, while Precision decreases to 0.0725, resulting in a Micro-F1 of 0.1252 and a Macro-F1 of 0.1157. By comparison, Direct n-gram achieves a Precision of 0.1609 and a Recall of 0.3313, with a Micro-F1 of 0.2166 and a Macro-F1 of 0.2001, yielding a relatively balanced result between Precision and Recall.

Automatic RLC produces exactly the same automatic identification results as Direct n-gram. Both methods identify 218 TPs, while producing 1,137 FPs and 440 FNs, with identical Precision, Recall, Micro-F1, and Macro-F1 values of 0.1609, 0.3313, 0.2166, and 0.2001, respectively. Under the frozen automatic output rules, the two candidate-generation procedures ultimately produce the same set of automatically identified terms, and therefore show no difference in term identification metrics.

After expert semantic validation of the RLC candidate pool, the final Precision and Recall reach 0.5386 and 0.4347, respectively, while Micro-F1 increases to 0.4811 and Macro-F1 to 0.4627. Compared with Automatic RLC, Precision increases by 0.3777 and Recall by 0.1033, while Micro-F1 and Macro-F1 increase by 0.2645 and 0.2626, respectively. In terms of identification counts, TP increases from 218 to 286, FP decreases from 1,137 to 245, and FN decreases from 440 to 372. The results after expert semantic validation therefore retain more Gold terms while excluding a substantial number of non-Gold candidates, resulting in higher Precision, Micro-F1, and Macro-F1 than those of all automatic methods.

5.2 Candidate Discovery and Search-Space Compression

As shown in Fig. 2 and Table 3, under the same lexical-unit representation, candidate-length range, and textual boundary constraints, Direct n-gram generates 35,436 candidate types, whereas the RLC candidate pool contains 15,341 candidate types. This corresponds to a reduction of 20,095 candidates and a Candidate Compression of 56.71%. In terms of candidate occurrences, Direct n-gram generates 46,417 occurrences, compared with 26,187 for RLC, representing a reduction of 43.58%. The reductions in both candidate types and candidate occurrences indicate that recursive pruning not only compresses the deduplicated candidate set but also reduces the actual volume of candidates that must be generated and processed.

Fig. 2 also shows the Gold coverage retained after candidate-space compression. The Test set contains 658 Gold_multi document-term types, of which 530 are included in the RLC candidate pool, while 128 are not covered during the candidate discovery stage. This corresponds to a Candidate Pool Recall of 80.55%. Thus, while reducing the number of candidate types from 35,436 to 15,341 and achieving a Candidate Compression of 56.71%, RLC still covers approximately four-fifths of the Gold terms. The substantial reduction in the candidate space therefore does not result in a proportional loss of Gold coverage.

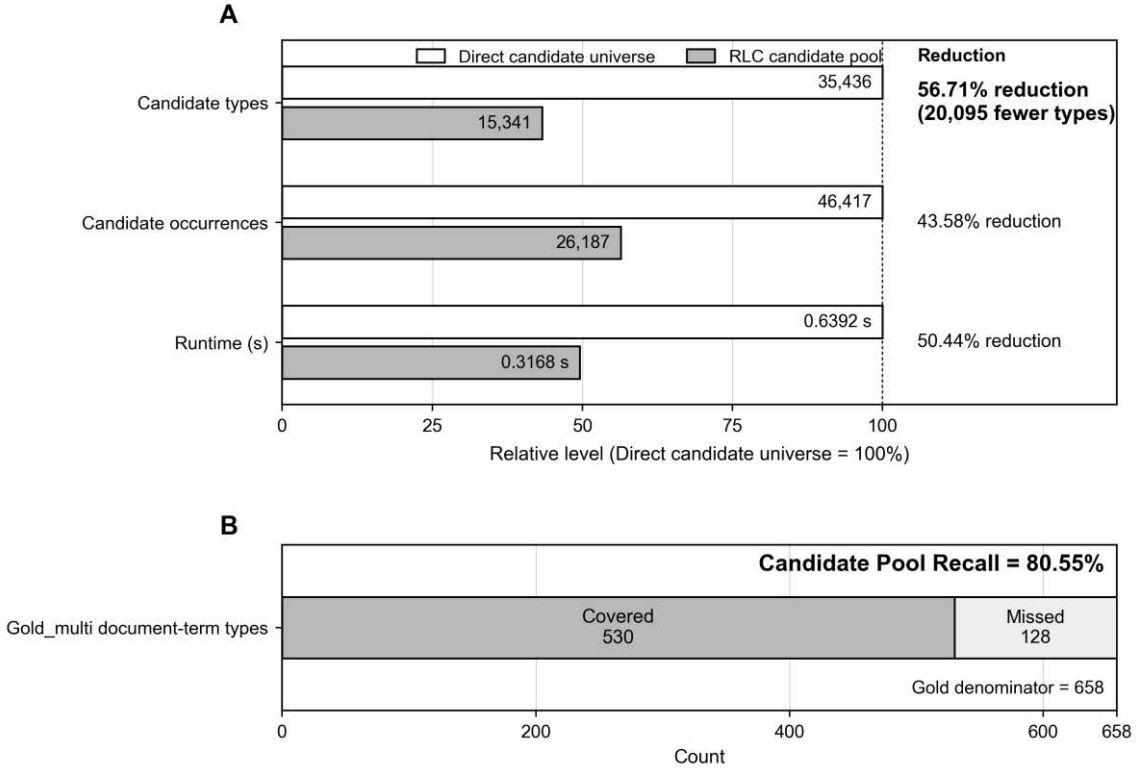


Fig. 2 Candidate space reduction and Gold coverage comparison between Direct n-gram and RLC on the held-out Test set

Table 3 further reports the computational cost of the candidate generation process. The single-run runtime is 0.6392 s for Direct n-gram and 0.3168 s for RLC, corresponding to a reduction of 50.44%, which is consistent with the reductions in candidate types and candidate occurrences. Because runtime is based on a single wall-clock record under the current experimental environment, it is used only as an auxiliary indicator of changes in candidate-generation scale rather than as an independent basis for performance comparison.

Table 3 Candidate discovery and computational efficiency on the held-out Test set

Metric	Direct n-gram	RLC	Change
Candidate types	35,436	15,341	−56.71%

Metric	Direct n-gram	RLC	Change
Candidate occurrences	46,417	26,187	−43.58%
Runtime (s)	0.6392	0.3168	−50.44%
Gold_multi document-term types covered	—	530 / 658	80.55%
Candidate Pool Recall	—	0.8055	—

Note: *Candidate Compression is calculated relative to the Direct candidate universe. Candidate Pool Recall denotes the proportion of Gold_multi document-term types contained in the RLC candidate pool. Runtime refers to a single wall-clock record under the current experimental environment.*

Overall, RLC exhibits a clear relationship between candidate compression and Gold coverage on the held-out Test set. Relative to Direct n-gram, RLC removes more than half of the candidate types while maintaining a Candidate Pool Recall of 80.55%. These results show that candidate-space compression does not produce a Gold coverage loss of comparable magnitude.

5.3 Length-Dependent Compression–Coverage Trade-off

Fig. 3 further compares RLC candidate coverage and candidate-space compression across different term lengths. Candidate Pool Recall decreases consistently as term length increases. For 2-unit terms, all 253 Gold terms in the Test set are included in the RLC candidate pool, yielding a Recall of 100.00%. When term length increases to 3 units, Recall decreases to 77.55% (190/245). Recall further declines to 60.87% (70/115) for 4-unit terms and 47.06% (16/34) for 5-unit terms. Among terms of 6 or more units, only 1 of the 11 Gold terms is included in the candidate pool, corresponding to a Recall of 9.09%.

In contrast to the decline in coverage, candidate-space compression increases consistently with term length. For 2-unit terms, both Direct n-gram and RLC generate 8,248 candidates, resulting in no candidate compression. For 3-unit terms, the number of candidates decreases from 8,756 under Direct n-gram to 3,885 under RLC, corresponding to a Compression of 55.63%. Compression further increases to 76.91%, 84.49%, and 88.59% for 4-unit, 5-unit, and 6+ unit terms, respectively, with RLC retaining 1,749, 956, and 557 candidates in the corresponding groups. Thus, as term length increases, RLC retains progressively fewer candidates relative to the Direct candidate universe.

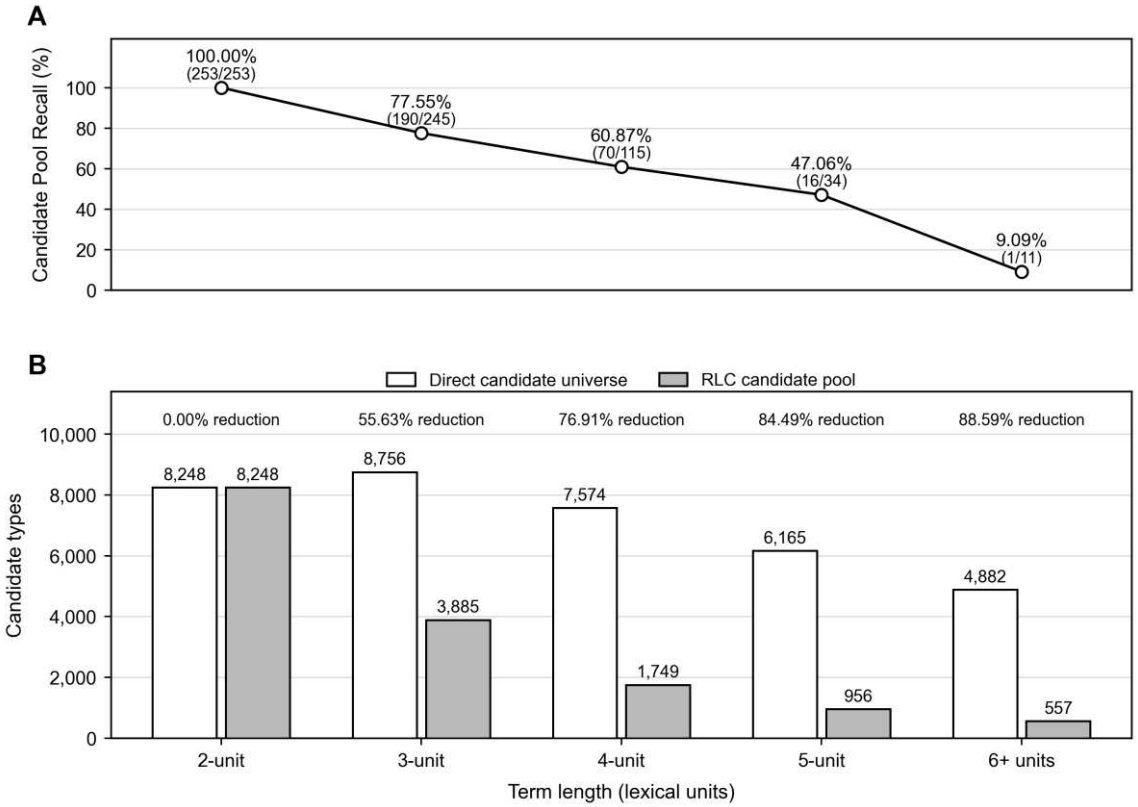


Fig. 3 Length-dependent trade-off between Candidate Pool Recall and candidate-space compression on the held-out Test set

Taken together, these results show that candidate discovery performance varies systematically with term length. For 2-unit terms, RLC does not compress the candidate space and retains complete Gold coverage. For terms of 3 to 5 units, candidate compression becomes increasingly pronounced, while Candidate Pool Recall remains between 47.06% and 77.55%. By contrast, 6+ unit terms exhibit the highest level of candidate compression but substantially lower Gold coverage. Fig. 3 therefore reveals a clear length-dependent relationship: as term length increases, RLC achieves stronger candidate-space compression, accompanied by lower Gold coverage.

The number of Gold terms also differs across length groups, with the 6+ unit group containing only 11 Gold_multi document-term types. Overall, RLC maintains relatively high candidate coverage for short- and medium-length terms. As term length increases, candidate compression becomes stronger, while Gold coverage progressively declines.

5.4 Cross-Document Stability and Held-out Generalization

Table 4 further reports the candidate discovery and term identification results of RLC across the six held-out Test documents. Candidate Pool Recall remains above 70% for all documents, ranging from 71.71% to 91.23%. Specifically, P12 and P27 achieve Candidate Pool Recall values of 91.23% and 90.91%, respectively; P07 and P29 reach 78.33% and 78.92%; and P04 and P16 reach 71.71% and 74.07%. All six documents therefore maintain relatively high levels of Gold coverage, indicating that the candidate discovery capability of RLC remains comparatively stable across different policy texts.

Automatic RLC yields document-level F1 values ranging from 0.0774 to 0.2944, whereas F1 after expert semantic validation ranges from 0.3502 to 0.6412. Specifically, the Expert F1 values for P04, P07, P12, P16, P27, and P29 are 0.4033, 0.3502, 0.6412, 0.3600, 0.5333, and 0.4882, respectively. For all six Test documents, Expert F1 is higher than the corresponding Automatic RLC result, with the direction of improvement remaining consistent across documents.

Table 4 Per-document performance on the held-out Test set

Document	RLC candidate pool	Candidate Pool Recall	Automatic RLC F1	Expert Precision	Expert Recall	Expert F1
P04	3,024	0.7171	0.2161	0.5385	0.3224	0.4033
P07	3,167	0.7833	0.2283	0.3285	0.3750	0.3502
P12	3,257	0.9123	0.2944	0.6450	0.6374	0.6412
P16	1,695	0.7407	0.0774	0.3913	0.3333	0.3600
P27	893	0.9091	0.2000	0.5217	0.5455	0.5333
P29	4,574	0.7892	0.1846	0.7045	0.3735	0.4882

Note: *Candidate Pool Recall is calculated using Gold_multi document-term types within each Test document. Expert results correspond to the final semantic-review stage.*

The comparison between Validation and the held-out Test further shows that the main characteristics of RLC candidate discovery persist under the frozen configuration. Candidate Compression changes from 59.90% on the Validation set to 56.71% on the Test set, indicating that the degree of candidate-space compression remains broadly stable. Candidate Pool Recall increases from 68.25% to 80.55%. For Gold terms outside the Seed Lexicon, the coverage of the RLC candidate pool likewise increases from 65.52% to 78.74%. Thus, without further adjustment of parameters or rules, RLC maintains a similar level of candidate compression on the held-out Test set while achieving higher coverage of both overall true terms and seed-uncovered terms.

Taken together, the per-document Test results and the Validation–Test comparison show that RLC maintains stable candidate discovery performance on policy documents not involved in parameter selection. Candidate-space compression does not materially weaken on the held-out Test set, while Gold coverage increases; meanwhile, F1 after expert semantic validation exceeds Automatic RLC for all six Test documents. These results indicate that, within the current data asset policy corpus, the candidate compression and term coverage characteristics of RLC extend consistently to held-out documents.

5.5 Discovery of Seed-Uncovered Terms

Table 5 further compares the ability of different methods to discover true domain terms outside the Seed Lexicon. The held-out Test set contains 602 seed-uncovered Gold_multi document-term types. C-value correctly identifies 252 of these terms, corresponding to a Recall of 41.86%. Direct n-gram and Automatic RLC each correctly identify 174 terms, with a Recall of 28.90%. By contrast, the RLC candidate pool covers 474 of the 602 terms, corresponding to a coverage rate of 78.74%. Thus, most true terms not covered by the fixed Seed Lexicon are successfully included in the candidate set available for subsequent semantic validation.

The coverage of seed-uncovered terms in the RLC candidate pool is close to its overall Candidate Pool Recall. Across the full Test set, RLC covers 530 of 658 Gold_multi document-term types, yielding a Candidate Pool Recall of 80.55%, while its coverage of seed-uncovered Gold remains 78.74%. This result shows that RLC maintains a relatively high level of coverage even for true terms outside the fixed Seed Lexicon.

After expert semantic validation, 239 of the 602 seed-uncovered Gold terms are retained in the final results, corresponding to a Recall of 39.70%. This value is lower than the candidate-pool coverage of 78.74%, reflecting the additional filtering that occurs between candidate discovery and final semantic validation. Compared with Automatic

RLC, expert validation increases seed-uncovered Recall from 28.90% to 39.70%, allowing more true terms already present in the candidate pool to be retained in the final results.

Table 5 Discovery of Seed-uncovered Gold terms on the held-out Test set

Method / stage	Correctly covered	Recall
Seed-based Lexicon	0	0.00%
C-value	252	41.86%
Direct n-gram	174	28.90%
Automatic RLC	174	28.90%
RLC candidate pool	474	78.74%
RLC + Expert Consensus	239	39.70%

Note: *Seed-uncovered Gold consists of Gold_multi document-term types that are not contained in the fixed 221-term Seed Lexicon. Candidate-pool results represent candidate discovery coverage, whereas the Expert Consensus result represents terms retained after semantic review.*

This complementary coverage is also reflected in specific terms. RLC candidate discovery and subsequent semantic validation identify true terms outside the Seed Lexicon such as “数据产品经营权” (data product operating rights), “可信数据空间” (trusted data space), “数据基础设施安全标准” (data infrastructure security standards), “数据资源无形资产” (data resource intangible assets), “个人信息出境标准合同” (standard contract for outbound transfer of personal information), and “数据安全事件分级” (data security incident classification). These terms span different themes, including data rights, data infrastructure, data security, data asset accounting, and cross-border data governance, indicating that seed-uncovered discovery is not concentrated in a single type of domain expression.

Overall, the RLC candidate pool maintains high coverage of true terms outside the Seed Lexicon, successfully discovering 474 of the 602 seed-uncovered Gold terms. Together with the overall candidate-compression results reported above, this finding further shows that RLC can reduce the candidate search space while still discovering a substantial number of domain terms from within the corpus that are not covered by the predefined lexicon.

6 Discussion

6.1 Candidate Discovery and Semantic Validation

The held-out Test results show that Direct n-gram and Automatic RLC produce identical term outputs under the unified automatic output rules, despite substantial differences in the scale of their candidate-generation processes. Relative to Direct n-gram, RLC reduces the number of candidate types from 35,436 to 15,341, corresponding to a Candidate Compression of 56.71%, while maintaining a Candidate Pool Recall of 80.55%. The effect of RLC recursive pruning therefore occurs primarily in the candidate search process preceding automatic output: by terminating low-support paths, it reduces the number of combinations that enter subsequent filtering and semantic validation.

This candidate compression exhibits a clear dependence on term length. RLC retains 100% coverage of 2-unit terms, while Candidate Pool Recall progressively declines and the proportion of candidate compression increases as combination length grows. Candidate Compression reaches 55.63%, 76.91%, and 84.49% for 3-unit, 4-unit, and 5-unit terms, respectively, while the corresponding Candidate Pool Recall declines from 77.55% to 47.06%. For terms of 6 or more units, Candidate Compression reaches 88.59%, whereas coverage falls to 9.09%. This pattern is consistent with the recursive expansion rule of RLC. Shorter combinations with sufficient support can continue to form longer candidates;

as combination length increases, some low-support paths are terminated early, thereby reducing the number of higher-order candidates. The resulting control of the candidate space is therefore accompanied by reduced coverage of low-frequency long terms, producing a consistent length-dependent relationship between candidate compression and term coverage.

The coverage of terms outside the predefined lexicon further illustrates the role of corpus-driven compositional search in RLC candidate discovery. The Test set contains 602 Gold terms not covered by the fixed Seed Lexicon, of which 474 enter the RLC candidate pool, corresponding to a Candidate Pool Recall of 78.74%. These include expressions such as “可信数据空间” (trusted data space), “数据产品经营权” (data product operating rights), “数据基础设施安全标准” (data infrastructure security standards), and “数据安全事件分级” (data security incident classification), whose discovery is not restricted to the existing term scope of the fixed Seed Lexicon. Recursive expansion based on contiguous lexical units and their support relationships in the source text allows candidate discovery to proceed along compositional paths that actually occur in the corpus, thereby extending coverage to specialized expressions not included in the predefined lexicon. At the same time, some seed-uncovered terms remain absent from the candidate pool, particularly longer combinations or those with lower support, reflecting the constraint that recursive support requirements impose on the scope of new-term discovery.

Once candidates enter the RLC candidate pool, statistical rules alone remain insufficient to fully resolve domain relevance, lexical completeness, and semantic boundaries. After expert semantic validation, Micro-F1 increases from 0.2166 for Automatic RLC to 0.4811. RLC uses lexical combinations and statistical support to control the candidate space, while expert semantic validation further determines the domain meaning, lexical boundaries, and conceptual completeness of the candidates. Candidate discovery and semantic validation thus form a continuous process from corpus-based compositional search to final term determination.

6.2 Scope and Limitations of the Method

RLC candidate discovery relies on contiguous lexical units and their statistical support in the corpus, and its discovery capability is therefore jointly influenced by term length and occurrence frequency. The held-out Test results show that 2-unit terms are fully covered, whereas Candidate Pool Recall progressively declines as term length increases, with substantially lower coverage for 6+ unit terms than for shorter terms. Error analysis also shows that some candidate-discovery failures arise from low-support recursive paths and failures to expand longer terms. This indicates that, although a fixed support constraint effectively compresses the higher-order candidate space, it is also more likely to terminate the generation paths of low-frequency long terms prematurely. For domains containing a large number of long compound terms or low-frequency specialized expressions, future extensions could adjust the expansion threshold according to candidate length or introduce supplementary discovery mechanisms for low-frequency long terms to improve the balance between candidate compression and coverage.

The experimental corpus in this study consists primarily of national-level data asset policy texts. These documents have relatively standardized linguistic structures and stable specialized expressions, providing a suitable environment for candidate discovery based on contiguous combinations and statistical support. Candidate Pool Recall remains above 70% across all six held-out Test documents, while the proportions of candidate compression are also relatively similar between the Validation and Test sets, indicating a degree of stability of the frozen configuration across different policy documents within the current domain. However, corpora such as news articles, social media texts, and internal corporate documents may differ substantially in expression patterns, lexical boundaries, and compositional stability. The current findings therefore cannot be directly generalized to these text types. The applicability of RLC to other specialized domains and informal corpora requires further evaluation using independent datasets.

The Gold Standard and expert semantic validation also involve human judgment in lexicon evaluation. The current Gold Standard was constructed through full-text annotation followed by a second-round review, without independent

double annotation. Consequently, the semantic boundaries and conceptual granularity of some specialized terms may still be affected by annotation judgments. Although expert semantic validation can further exclude incomplete combinations, general expressions, and candidates whose domain scope is inconsistent with the target domain and that are difficult to identify using statistical rules alone, the final retention decisions still depend on the criteria used to assess domain relevance and lexical completeness. Future research could introduce independent double annotation, inter-annotator agreement statistics, and more detailed term-boundary guidelines to improve the consistency of both the Gold Standard and final lexicon evaluation.

Overall, RLC is better suited to domain corpora characterized by continuously evolving specialized terminology, frequent compound expressions, and relatively standardized textual structures. In such settings, recursive candidate discovery can exploit local compositional structures in the corpus to control candidate scale while providing a traceable candidate basis for subsequent domain semantic judgment. Further research could focus on the discovery of longer terms, adaptive expansion conditions, and cross-domain and cross-text-type validation to examine the stability of the relationship between candidate compression and true-term coverage across different corpus environments.

7 Conclusion

This study addresses two challenges in domain lexicon construction for emerging Chinese domains: the limited coverage of predefined vocabularies and the rapid expansion of candidate spaces generated by contiguous lexical-unit combinations. To address these issues, we propose Recursive Lexical Combination (RLC), which progressively generates candidates from contiguous lexical units and their statistical support relationships in the source text, while terminating further expansion along low-support paths. Automatic filtering and expert semantic validation are subsequently applied to complete term identification. Using national-level data asset policy texts, the study adopts an experimental design that separates the Validation set from the held-out Test set and systematically evaluates candidate discovery, search-space compression, seed-uncovered term discovery, and final term identification.

The results show that, on the held-out Test set, RLC reduces the number of candidate types from 35,436 to 15,341, corresponding to a Candidate Compression of 56.71%, while retaining 530 Gold_multi document-term types and achieving a Candidate Pool Recall of 80.55%. Candidate compression increases with term length, accompanied by corresponding changes in coverage. Among the 602 Gold terms not covered by the fixed Seed Lexicon, 474 are included in the RLC candidate pool, indicating that recursive discovery based on compositional relationships within the corpus can identify a substantial number of specialized expressions beyond the existing lexicon. Expert semantic validation further improves final term identification quality, increasing Micro-F1 from 0.2166 for Automatic RLC to 0.4811. All six held-out Test documents maintain relatively high levels of candidate coverage, and the characteristics of candidate compression and term discovery remain broadly consistent across different policy texts.

These findings frame domain lexicon construction as two connected processes: candidate discovery and semantic validation. Lexicon expansion based on a fixed Seed Lexicon relies on existing domain knowledge, whereas direct contiguous enumeration retains a much larger candidate search space. RLC provides a traceable candidate discovery path between these two approaches by using compositional structures within the corpus to progressively generate new terms and concentrating subsequent domain judgment on a compressed candidate set. The scope of lexicon candidates can therefore be driven to a greater extent by the actual compositional relationships present in the target corpus rather than being entirely constrained by a predefined vocabulary. For Chinese specialized domains with continuously evolving terminology, this combination of corpus-driven candidate discovery and domain semantic validation provides an interpretable and reproducible approach to identifying emerging specialized terms and constructing domain lexicons.

Statements and Declarations

Funding

No funds, grants, or other support was received.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Author contributions

Both authors contributed to the conception and design of the study, analysis and interpretation of results, manuscript drafting and revision, and approved the version submitted for publication.

Ethics approval

Ethical approval was not required for this study. The study used publicly available policy texts and did not involve the collection of personal or sensitive data.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Data availability

The policy documents analysed in this study are publicly available national-level data-asset policy texts. Derived annotation data, the Gold Standard, and candidate lists supporting the reported findings are available from the corresponding author on reasonable request, subject to applicable legal, licensing, and institutional restrictions.

Code availability

The code supporting the findings of this study is available from the corresponding author on reasonable request.

References

Chun, Y., Kim, M., Kim, D., Park, C., & Lim, H. (2025). Enhancing automatic term extraction with large language models via syntactic retrieval. *Findings of the Association for Computational Linguistics: ACL 2025*, 9916–9926. https://doi.org/10.18653/v1/2025.findings-acl.516

Di Nunzio, G. M., Marchesin, S., & Silvello, G. (2023). A systematic review of automatic term extraction: What happened in 2022? *Digital Scholarship in the Humanities, 38*(Supplement_1), i41–i47. https://doi.org/10.1093/llc/fqad030

Du, Z., Huang, A. G., Wermers, R., & Wu, W. (2022). Language and domain specificity: A Chinese financial sentiment dictionary. *Review of Finance, 26*(3), 673–719. https://doi.org/10.1093/rof/rfab036

Frantzi, K. T., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: The C-value/NC-value method. *International Journal on Digital Libraries, 3*(2), 115–130. https://doi.org/10.1007/s007999900023

Gao, J., Li, M., Wu, A., & Huang, C.-N. (2005). Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics, 31*(4), 531–574. https://doi.org/10.1162/089120105775299177

Gentzkow, M., Kelly, B., & Taddy, M. (2019). Text as data. *Journal of Economic Literature, 57*(3), 535–574. https://doi.org/10.1257/jel.20181020

He, Y., Chen, L., & Du, Y. (2024). Can data assetization alleviate financing constraints of "specialized, refined, distinctive, and innovative" SMEs? *China Industrial Economics*, (8), 154–173. https://doi.org/10.19581/j.cnki.ciejournal.2024.08.008 [In Chinese]

Huang, J. H., & Powers, D. M. W. (2003). Chinese word segmentation based on contextual entropy. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation* (pp. 152–158). COLIPS Publications. https://aclanthology.org/Y03-1017/

Jin, Z., & Tanaka-Ishii, K. (2006). Unsupervised segmentation of Chinese text by use of branching entropy. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions* (pp. 428–435). Association for Computational Linguistics. https://aclanthology.org/P06-2056/

Li, X., Yan, H., Qiu, X., & Huang, X. (2020). FLAT: Chinese NER using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 6836–6842). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.611

Li, Z., & Sun, M. (2009). Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics, 35*(4), 505–512. https://doi.org/10.1162/coli.2009.35.4.35403

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance, 66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

Loughran, T., & McDonald, B. (2020). Textual analysis in finance. *Annual Review of Financial Economics, 12*, 357–375. https://doi.org/10.1146/annurev-financial-012820-032249

Ma, Y., Zhang, T., & Zhang, P. (2025). The construction of a Chinese fine-grained sentiment dictionary for Chinese domestic investors (CN-FSD) and its application. *Expert Systems with Applications, 264*, 125739. https://doi.org/10.1016/j.eswa.2024.125739

Nugumanova, A., Akhmed-Zaki, D., Mansurova, M., Baiburin, Y., & Maulit, A. (2022). NMF-based approach to automatic term extraction. *Expert Systems with Applications, 199*, 117179. https://doi.org/10.1016/j.eswa.2022.117179

Panoutsopoulos, H., Espejo-Garcia, B., Raaijmakers, S., Wang, X., Fountas, S., & Brewster, C. (2024). Investigating the effect of different fine-tuning configuration scenarios on agricultural term extraction using BERT. *Computers and Electronics in Agriculture, 225*, 109268. https://doi.org/10.1016/j.compag.2024.109268

Rigouts Terryn, A., Hoste, V., & Lefever, E. (2020). In no uncertain terms: A dataset for monolingual and multilingual automatic term extraction from comparable corpora. *Language Resources and Evaluation, 54*(2), 385–418. https://doi.org/10.1007/s10579-019-09453-9

Tran, H. T. H., González-Gallardo, C. E., Doucet, A., & Pollak, S. (2025). LlamATE: Automated terminology extraction using large-scale generative language models. *Terminology, 31*(1), 5–36. https://doi.org/10.1075/term.00082.tra

Tran, H. T. H., Martinc, M., Caporusso, J., Delaunay, J., Doucet, A., & Pollak, S. (2026). Recent advances in automatic term extraction: A comprehensive survey. *ACM Computing Surveys, 58*(9), Article 226, 1–35. https://doi.org/10.1145/3787584

Tran, H. T. H., Martinc, M., Repar, A., Ljubešić, N., Doucet, A., & Pollak, S. (2024). Can cross-domain term extraction benefit from cross-lingual transfer and nested term labeling? *Machine Learning, 113*, 4285–4314. https://doi.org/10.1007/s10994-023-06506-7

Wissik, T. (2025). Impact of automatic term extraction on terminology work: A qualitative interview study in institutional settings. *Terminology, 31*(1), 110–135. https://doi.org/10.1075/term.00085.wis

Xiao, Y., Ji, Z., Li, J., & Zhu, Q. (2025). DualFLAT: Dual Flat-Lattice Transformer for domain-specific Chinese named entity recognition. *Information Processing & Management, 62*(1), 103902. https://doi.org/10.1016/j.ipm.2024.103902

Xu, K., Feng, Y., Li, Q., Dong, Z., & Wei, J. (2025). Survey on terminology extraction from texts. *Journal of Big Data, 12*, 29. https://doi.org/10.1186/s40537-025-01077-x

Zhang, H., Dang, Y., Zhang, Y., Liang, S., Liu, J., & Ji, L. (2024). Chinese nested entity recognition method for the finance domain based on heterogeneous graph network. *Information Processing & Management, 61*(5), 103812. https://doi.org/10.1016/j.ipm.2024.103812

Zhou, Y., Fan, J., & Xue, L. (2024). How much can machines learn finance from Chinese text data? *Management Science, 70*(12), 8962–8987. https://doi.org/10.1287/mnsc.2022.01468