Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.073212

ARTICLE

A Chinese Abbreviation Prediction Framework Based on Chain-of-Thought Prompting and Semantic Preservation Dynamic Adjustment

Jingru Lv[1], Jianpeng Hu[1,*], Jin Zhao[2] and Yonghao Luo[1]

[1]School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China
[2]School of Computer Science, Fudan University, Handan Road, Shanghai, 200433, China

*Corresponding Author: Jianpeng Hu. Email: mr@sues.edu.cn

Received: 12 September 2025; Accepted: 26 November 2025; Published: 10 February 2026

ABSTRACT: Chinese abbreviations improve communicative efficiency by extracting key components from longer expressions. They are widely used in both daily communication and professional domains. However, existing abbreviation generation methods still face two major challenges. First, sequence-labeling-based approaches often neglect contextual meaning by making binary decisions at the character level, leading to abbreviations that fail to capture semantic completeness. Second, generation-based methods rely heavily on a single decoding process, which frequently produces correct abbreviations but ranks them lower due to inadequate semantic evaluation. To address these limitations, we propose a novel two-stage framework with Generation–Iterative Optimization for Abbreviation (GIOA). In the first stage, we design a Chain-of-Thought prompting strategy and incorporate definitional and situational contexts to generate multiple abbreviation candidates. In the second stage, we introduce a Semantic Preservation Dynamic Adjustment mechanism that alternates between character-level importance estimation and semantic restoration to optimize candidate ranking. Experiments on two public benchmark datasets show that our method outperforms existing state-of-the-art approaches, achieving Hit@1 improvements of 15.15% and 13.01%, respectively, while maintaining consistent results in Hit@3.

KEYWORDS: Abbreviation; chain-of-thought prompting; semantic preservation dynamic adjustment; candidate ranking

1 Introduction

Abbreviations condense full expressions into shorter forms while preserving core semantics. Unlike English abbreviations that typically follow the initial letter rule (e.g., “NBA” derived from “National Basketball Association”), Chinese abbreviation generation is more complex and irregular. As shown in Table 1, “复旦大学” (Fudan University) is abbreviated as “复旦” by keeping the first two characters, “上海迪士尼” (Shanghai Disney) is shortened to “迪士尼” by selecting the last three characters, while “北京大学” (Peking University) is abbreviated as “北大” by selecting two discontinuous characters. This irregular character selection, together with the requirement to preserve semantic integrity, poses great challenges for language models and downstream applications such as entity linking, information extraction, and text understanding in accurately interpreting Chinese abbreviations. Therefore, Chinese abbreviation prediction is not only a linguistic problem but also a key task for enhancing semantic representation and understanding in NLP systems. These complexities highlight the necessity for a more structured and semantically guided approach to Chinese abbreviation prediction.

Table 1: Examples of full forms and abbreviations

Full form	Abbreviation
复旦大学 (Fudan University)	复旦
上海迪士尼 (Shanghai Disney)	迪士尼
北京大学 (Peking University)	北大

In addressing the irregularity of Chinese abbreviation prediction, previous studies have mainly explored two lines of research: sequence labeling and sequence generation. The sequence labeling method models Chinese abbreviation prediction as a character-level label assignment task: each character in the full form is assigned a binary label (e.g., "1" indicates retention and "0" indicates exclusion). For such tasks, traditional models like Conditional Random Fields (CRF) [1–3] and Support Vector Machines (SVM) [4] are typical solutions. These models leverage either manually designed features or automatically extracted features via neural networks to learn the correlation between features and labels, thereby identifying the most probable label sequence. However, these methods often fail to capture the global semantics of the complete expression.

Motivated by these limitations, sequence generation methods were introduced. They treat abbreviation prediction as a sequence-to-sequence task, generating abbreviations directly from full forms using encoder–decoder architectures. Some models further incorporate auxiliary components, such as quality evaluators [5] or abbreviation-type classifiers [6], to improve output relevance. Although generation-based methods alleviate the limitations of feature-based models through contextual representations, they still struggle with candidate ranking, as the correct abbreviation may not be prioritized. These limitations reveal two fundamental challenges in Chinese abbreviation prediction: (1) semantic incompleteness, as existing models fail to capture global contextual meaning; and (2) suboptimal candidate ranking, since correct abbreviations may not appear among the top positions during decoding.

To address these limitations, we introduce Generation–Iterative Optimization for Abbreviation (GIOA), a novel two-stage framework that integrates structured reasoning with semantic refinement. The motivation behind this design is to enable both structured reasoning and dynamic adjustment of semantic representations, ensuring that generated abbreviations are semantically faithful and properly ranked. In the first stage, we design a Chain-of-Thought (CoT) prompting template that integrates both definitional and situational contexts to guide structured abbreviation generation. This stage aims to capture the holistic semantics of the full form and produce semantically consistent candidate abbreviations. In the second stage, we introduce a Semantic Preservation Dynamic Adjustment (SPDA) mechanism, which iteratively alternates between character importance estimation and semantic integrity restoration to refine and re-rank candidate abbreviations, thereby improving prediction accuracy and reducing ranking errors.

In summary, the main contributions of this paper are as follows.

- We propose a novel two-stage framework named Generation–Iterative Optimization for Abbreviation (GIOA) for Chinese abbreviation prediction, which addresses the limitations of previous methods in semantic understanding and candidate ranking, thereby improving overall prediction accuracy.
- We introduce a Chain-of-Thought (CoT) prompting mechanism that integrates definitional and situational contexts to guide structured generation and produce semantically consistent and well-structured abbreviations.
- We design a Semantic Preservation Dynamic Adjustment (SPDA) mechanism that iteratively alternates between character importance estimation and semantic integrity restoration to refine and reorder candidates, improving both semantic fidelity and ranking accuracy.

- Extensive experiments conducted on two public benchmark datasets demonstrate the effectiveness of GIOA. The proposed method achieves improvements of 15.15% and 13.01% in the Hit@1 metric, respectively, while maintaining competitive performance on Hit@3 compared to state-of-the-art approaches.

2 Related Work

2.1 Abbreviation Generation Methods

Early studies on abbreviation generation mainly relied on rule-based and statistical approaches using linguistic features. Despite showing effectiveness, these approaches depended heavily on handcrafted features, lacking flexibility and cross-domain generalization capability.

With the advent of deep learning, neural methods became dominant. Early neural approaches typically modeled abbreviation prediction as a sequence labeling problem, using CRF [3] or HMM [5] models with handcrafted features such as part-of-speech tags and word boundaries [6]. BiLSTM-CRF models [7] were later introduced to capture richer contextual and semantic dependencies. While sequence labeling provided structural control, it still struggled to ensure global semantic consistency and context-dependent abbreviation accuracy [8].

To overcome these limitations, researchers reformulated abbreviation prediction as a sequence generation task. Wang et al. [8] pioneered the use of Seq2Seq models with label transformation and multi-layer embeddings, followed by attention-based and Transformer-based architectures [9] adversarial mechanism, offering greater flexibility. However, sequence generation models often rely on one-shot decisions and lack iterative refinement to improve candidate quality.

Recently, large language models (LLMs) [10] have demonstrated strong semantic understanding and reasoning abilities. Chain-of-Thought (CoT) prompting [11] enables structured reasoning by eliciting intermediate steps, further enhancing semantic comprehension. Nevertheless, existing methods still lack dynamic mechanisms for semantic refinement and candidate ranking, which motivates our proposed GIOA framework to address these challenges.

2.2 Semantic Similarity Computation and Candidate Ranking

Semantic similarity computation quantifies the relatedness between text segments. Early methods based on lexical or statistical features struggled to capture contextual meaning [12]. With deep learning, pre-trained language models provided distributed representations for more accurate similarity estimation [13]. Recent studies further adopted attention and context-fusion mechanisms to enhance fine-grained alignment [14], supporting better evaluation of semantic consistency between abbreviations and full forms.

Candidate ranking is a key optimization technique in information retrieval and natural language processing, aiming to refine the order of initial results through semantic modeling. Pairwise models such as MonoT5 [15] enhance ranking accuracy by jointly modeling the semantic relevance between queries and candidates, while late-interaction architectures like ColBERTv2 [16] achieve finer-grained semantic matching with high efficiency. Recent approaches, including listwise methods and large language model–based or graph-enhanced [17] strategies, further improve global optimization and semantic understanding in complex ranking scenarios.

3 Methodology

In this section, we provide a detailed introduction to the GIOA framework, beginning with an overall framework overview, followed by descriptions of the generation stage and iterative optimization stage.

3.1 Problem Definition

Given a Chinese full form $F = [x_1, x_2, ..., x_n]$ consisting of n characters, the goal is to generate a valid abbreviation $A = [a_1, a_2, ..., a_m]$, where $m < n$ and all characters in A are selected from F. To enhance semantic modeling, the input also incorporates two types of contextual information: a definitional context C_{def} that describes the basic meaning of the term, and a situational context C_{env} that reflects its practical usage in real scenarios.

3.2 GIOA Framework

As shown in Fig. 1, we propose the GIOA framework composed of two stages, namely the generation stage and the iterative optimization stage. The upper-left part of the framework consists of the input section, which includes full form, definitional context, and situational context. These inputs are processed in the generation stage, which employs a pre-trained Transformer model and a CoT prompting mechanism to sequentially perform semantic unit extraction, abbreviation rule analysis, and abbreviation form construction, thereby generating structurally sound and semantically appropriate initial abbreviation candidates. The candidate set then proceeds to the iterative optimization stage, where low-quality candidates with syntactic errors or semantic inconsistencies are first filtered using linguistic constraints. The remaining candidates are further processed by the SPDA mechanism, which alternates between character importance evaluation and semantic integrity restoration. Through iterative refinement, this mechanism dynamically balances information compression with semantic fidelity, ultimately producing the optimal abbreviation.

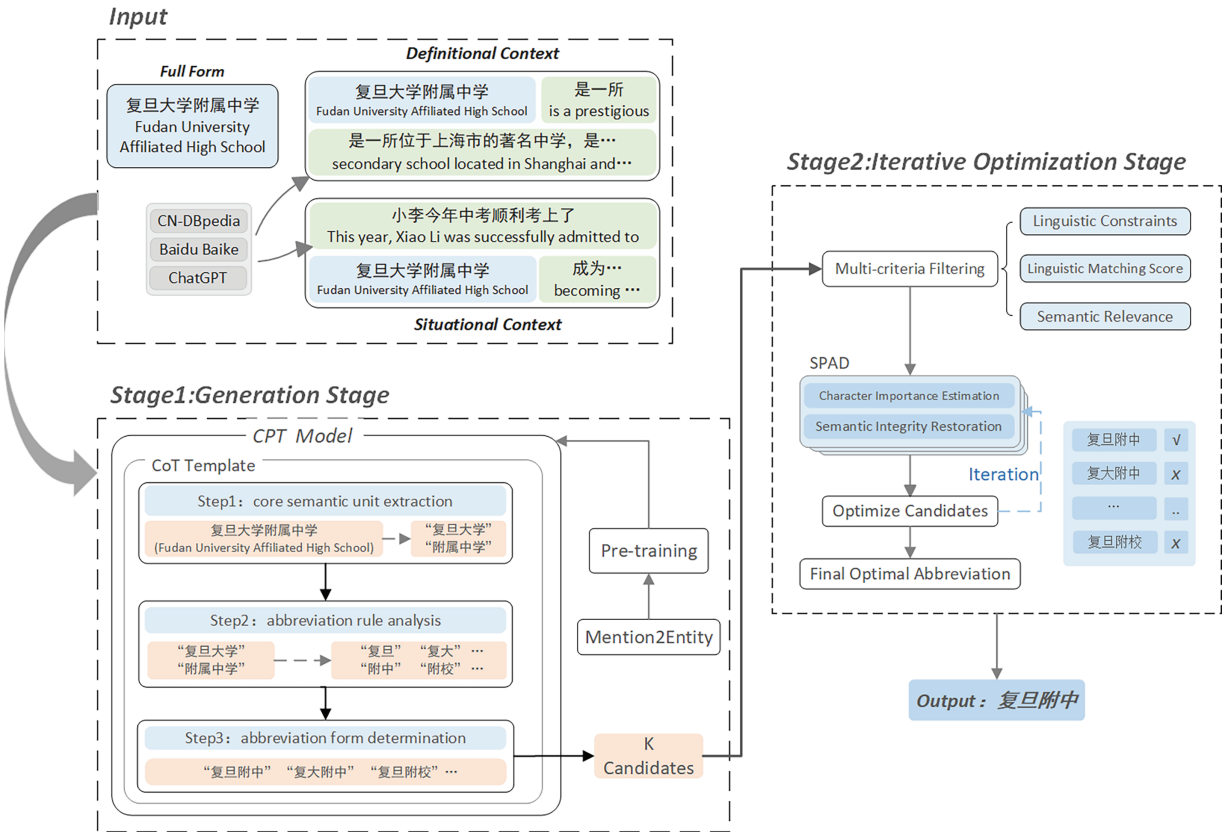

Figure 1: The overall architecture of the generation–iterative optimization framework

3.3 Generation Stage

In the generation stage, we construct a candidate generation process based on a CoT prompting mechanism. This process centers on the Chinese pre-trained model (CPT) [8], which performs deep semantic representation learning on the input full form and its dual contexts, including definitional and situational contexts, thereby enhancing the model's understanding of abbreviation usage scenarios. Building on this foundation, the CoT prompting mechanism guides the model to sequentially execute three key steps: core semantic extraction, abbreviation convention analysis, and abbreviation form determination, progressively generating structurally sound and semantically appropriate initial candidates. Structured semantic modeling and reasoning enable the model to accurately capture key information essential for abbreviation formation, which in turn improves candidate generation quality.

3.3.1 Pre-Trained CPT Model for Candidate Generation

To support semantic modeling for abbreviation generation, we employ the Chinese Pretrained Transformer (CPT) model, pre-trained on large-scale Chinese corpora. CPT integrates both NLU and NLG capabilities through a shared encoder and two specialized decoders optimized via masked language modeling and denoising auto-encoding. This architecture effectively captures deep semantic information for our abbreviation task. Given limited abbreviation training data, we fine-tuned CPT on the Mention2Entity (M2E) [18] dataset from CN-DBpedia [19], containing over one million entity-mention pairs with high semantic overlap. These pairs, with longer text as input and shorter text as output, enable sequence-to-sequence training that closely aligns with abbreviation generation requirements.

During the fine-tuning phase, given an input full form sequence [8] $F = [x_1, x_2, ..., x_n]$, the model learns to generate its abbreviated form $A = [a_1, a_2, ..., a_m]$ using an encoder-decoder architecture. The generation probability can be expressed as:

$$P(A|F) = \prod_{t=1}^{m} P(a_t|a_{<t}, F) \tag{1}$$

We optimize the model using the standard cross-entropy loss function, formulated as:

$$L_{gen} = -\frac{1}{m}\sum_{t=1}^{m} \log P_\theta(a_t|a_{<t}, F) \tag{2}$$

where θ denotes the model parameters, a_t represents the t-th character in the abbreviated form, and $a_{<t}$ denotes all previous characters before time step t.

3.3.2 Chain-of-Thought Prompting with Context Integration

To address the lack of interpretability in traditional generation models, we design a Chain-of-Thought prompting template that integrates contextual understanding [11]. This template guides the model to naturally incorporate both definitional context (standard word definitions) and situational context (specific usage scenarios) during the reasoning process [20].

The template guides the model through a three-step structured reasoning process:

1. Core Semantic Extraction: Identify the core semantic units and compositional structure within the full form.

2. Abbreviation Convention Analysis: Analyze appropriate abbreviation conventions based on definitional understanding and usage contexts.

3. Abbreviation Form Determination: Based on the analysis from the previous two steps, determine the most reasonable abbreviated form.

This approach allows the model to jointly consider the compositional structure of the full form, the semantic importance of each component, and contextual constraints during reasoning, forming a coherent decision chain rather than applying fixed rules [21]. To provide a clear illustration of the overall Chain-of-Thought reasoning process, we present a visual overview of the three-step reasoning procedure in Fig. 2. This example shows how definitional and situational contexts jointly guide structured abbreviation generation.

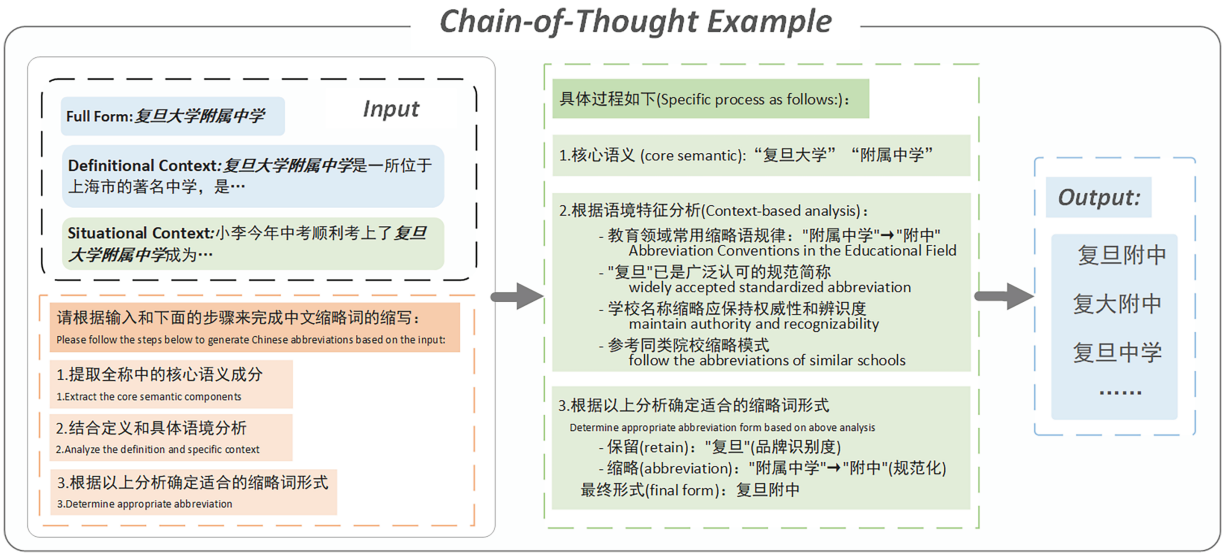

Figure 2: Overall chain-of-thought reasoning process for abbreviation generation

3.3.3 Dual-Layer Context Enhancement Mechanism

To enhance understanding of the holistic semantics of full forms, we propose a dual-layer context enhancement mechanism that aims to improve semantic comprehension through contextual augmentation. This mechanism integrates both definitional and situational contexts, as illustrated in Table 2, enabling the model to generate high-quality abbreviations that conform to general linguistic norms.

Table 2: Examples of definitional and situational contexts

Full form	Definitional context	Situational context
复旦大学附属中学	复旦大学附属中学是一所位于上海市的著名中学, 是复旦大学的附属学校.	小李今年中考顺利考上复旦大学附属中学, 是他们班第一个考入重点高中的.
Fudan University Affiliated High School	Fudan University Affiliated High School is a well-known secondary school in Shanghai and is affiliated with Fudan University.	Xiao Li passed the high school entrance examination this year and was admitted to Fudan University Affiliated High School, becoming the first student in his class to enter a key high school.

Definitional Context Acquisition. We first retrieve definitional context from CN-DBpedia; if it is unavailable, we search for relevant content from Baidu entries or search results. In a few cases, additional semantic information is supplemented through manual annotation. This definitional context offers standard definitions and background knowledge of full forms, aiding the model in understanding their formal semantics.

Situational Context Generation. To simulate usage scenarios in specific contexts, we mainly obtain real usage examples from news websites and academic literature databases such as CNKI, and further generate supplementary contexts using ChatGPT to expand the coverage of the corpus.

Table 3 presents the distribution of definitional and situational contexts collected from multiple sources. Specifically, the D_1 dataset contains approximately 94,655 definitional contexts and 94,655 situational contexts, while the D_2 dataset includes about 10,786 definitional contexts and 10,786 situational contexts. This large-scale and diverse contextual resource significantly enhances the quality and semantic plausibility of the generated candidates.

Table 3: Distribution of definitional and situational contexts across datasets

Context type	Context collection	D_1	D_2
Definitional context	CN-DBpedia	87,306 (92.2%)	9707 (90.0%)
	Baidu Baike items	4204 (4.5%)	863 (8.0%)
	Baidu Baike pages	2974 (3.1%)	205 (1.9%)
	Manual collection	171 (0.2%)	11 (0.1%)
Situational context	News websites	70,897 (74.9%)	8899 (82.5%)
	CNKI query	18,931 (20.0%)	1639 (15.2%)
	ChatGPT	4827 (5.1%)	248 (2.3%)

3.3.4 Candidate Generation Strategy

Based on the CoT analysis and enhanced semantic representation, we employ a beam-search strategy to generate diverse candidate abbreviations [22]. To ensure both quality and diversity, we adopt three practical constraints:

Diversity penalty. During beam search, we down-weight candidates that are overly similar to previously expanded ones to reduce redundancy in the candidate set.

Length control. The target length m is constrained proportionally to the full-form length n, avoiding extreme truncation or overlong outputs.

Quality threshold. Candidates with generation probability below a predefined threshold δ are discarded.

With the above CoT guidance and dual-layer context enhancement, the generator produces semantically accurate, usage-appropriate, and diverse candidate sets $\{A_1, A_2, \ldots, A_p\}$, providing high-quality pools for the subsequent optimization stage [23].

3.4 Iterative Optimization Stage

In this stage, to address the issue that correct abbreviations may be generated but fail to rank at the top, we design an optimization process centered on the SPDA mechanism. This process first applies rule-based

filtering to remove candidates with grammatical or structural inconsistencies, and then iteratively refines the remaining candidates by alternating between character-level evaluation and semantic integrity restoration.

3.4.1 Preliminary Screening Mechanism

To identify high-quality candidates from the initial abbreviation candidate set, we introduce a multi-dimensional screening framework that evaluates candidates based on linguistic feature constraints, linguistic matching degree, and semantic similarity.

1. Linguistic Feature Constraints: The system first checks basic properties of candidate abbreviations, such as whether their length falls within 2–6 characters and whether they form valid alphanumeric combinations for readability and display. It also filters out candidates containing common function words like "的", "和", and "与", which are usually omitted in abbreviations.

2. Linguistic Matching Degree Assessment: Evaluates the linguistic correlation between candidates and their full forms by checking whether candidates follow the pinyin initial-letter sequence, which is common in Chinese abbreviations. It also measures their edit distance (Levenshtein) to ensure textual correlation. For English or mixed Chinese-English abbreviations, it verifies that letter-case structure follows conventional usage patterns, including appropriate proportions of uppercase letters.

3. Semantic Relevance Evaluation: This critical component computes TF-IDF similarity between candidates and full forms to assess lexical association, then uses bert-base-chinese to measure deep semantic similarity, ensuring candidates retain the essential meaning of full forms. This deep learning-based evaluation significantly improves screening accuracy.

Our screening framework dynamically adjusts evaluation criteria based on full form characteristics. For longer full forms (over six characters), it lowers the semantic similarity threshold (from 0.5 to 0.3) to account for information loss during abbreviation. This adaptive mechanism handles diverse abbreviation patterns while maintaining quality standards [23]. Through multi-dimensional screening, the framework removes low-quality candidates while retaining high-quality ones, filtering about 60% of suboptimal candidates and preserving over 95% of high-quality ones, greatly improving downstream processing efficiency and quality.

3.4.2 Semantic Preservation Dynamic Adjustment Mechanism

To address the issues of incorrect candidate ranking and inconsistent abbreviation quality, we introduce the SPDA mechanism, which is built upon a pre-trained Chinese language model optimized for natural language understanding and generation tasks. This foundation provides robust support for abbreviation construction, semantic evaluation, and structural refinement [8]. The SPDA mechanism integrates character importance estimation with semantic integrity restoration through an iterative optimization process. In each iteration, the system evaluates the contextual significance of individual characters and performs semantic restoration based on the retained character combinations. This process progressively enhances both the representational quality and ranking accuracy of candidates until convergence is achieved or a predefined maximum number of iterations is reached, as illustrated in Fig. 3.

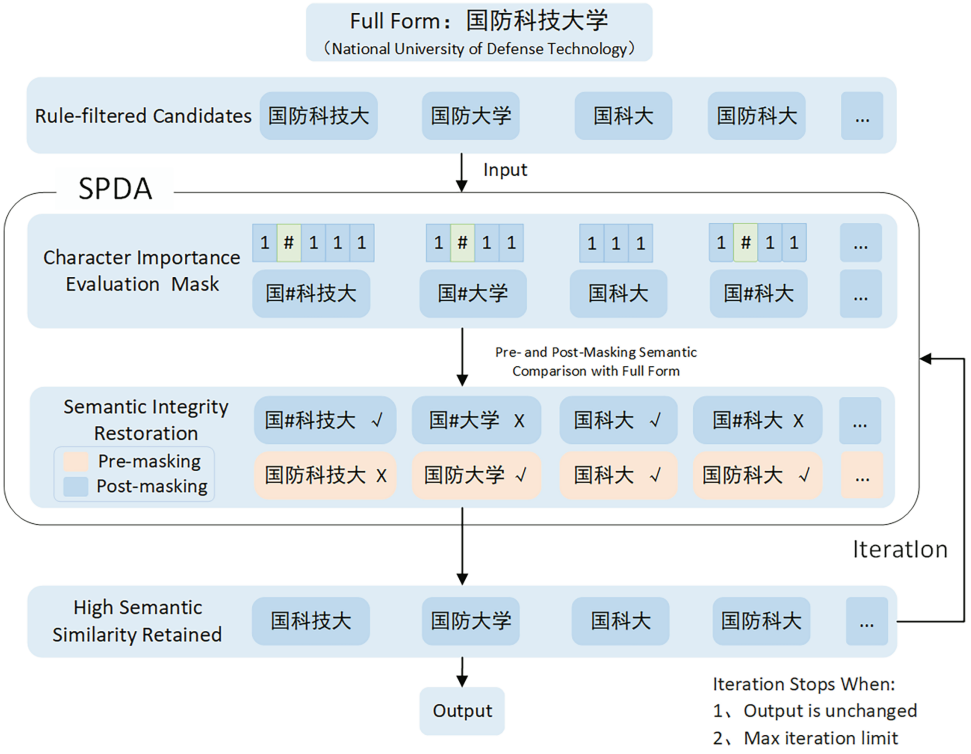

Figure 3: SPDA flowchart

In the character importance estimation stage, the system dynamically evaluates each character $a_{k,j}^{(t)}$ in the candidate abbreviation $A_k^{(t)} = [a_{k,1}, a_{k,2}, \ldots, a_{k,j}]$ to determine its contribution to semantic expression. A comprehensive scoring function is designed that considers character position information (initial or final characters tend to be more important), lexical features (content words are more informative), information entropy (rare characters usually convey more information), and centrality in the knowledge graph of the domain. The score is computed as:

$$\alpha_{k,j}^{(t)} = \sum_i \omega_i f_i(a_{k,j}) \tag{3}$$

where $f_i(\cdot)$ represents linguistic statistical features and ω_i denotes the corresponding weights.

The system then generates a masking code:

$$m_{k,j}^{(t)} = \begin{cases} 1, & \text{if } \alpha_{k,j}^{(t)} \geq \tau^{(t)} \quad \text{(retain)} \\ 0, & \text{if } \alpha_{k,j}^{(t)} < \tau^{(t)} \quad \text{(mask)} \end{cases} \tag{4}$$

That is, if the importance score of a character is above or equal to the current threshold, it is retained (mask code = 1); otherwise, it is replaced with a placeholder symbol "#", and excluded from the next round of abbreviation construction. The masked candidate is defined as:

$$A_k^{(t)\text{masked}} = \text{Mask}(A_k^{(t)}, m_k^{(t)}) \tag{5}$$

In the semantic integrity restoration stage, the system determines whether semantic meaning has changed after masking by comparing two semantic similarity scores with the original full name F.

The first type measures the semantic similarity between the original unmasked version $A_k^{(t)}$ and the full form, denoted as s_k^{raw}, which evaluates whether the original candidate provides a better semantic expression.

The second type measures the semantic similarity between the masked version $A_k^{(t)\text{masked}}$ and the full form, denoted as s_k^{mask}, which is used to assess whether the compression operation affects semantic integrity.

The similarity is computed as:

$$s_k^{\text{raw}} = \text{sim}(X, A_k^{(t)}), \quad s_k^{\text{mask}} = \text{sim}(X, A_k^{(t)\text{masked}}) \tag{6}$$

If the masked version retains better semantic integrity, we keep it; otherwise, we revert to the original:

$$A_k^{(t+1)} = \begin{cases} A_k^{(t)}, & \text{if } s_k^{\text{raw}} \geq s_k^{\text{mask}} \\ A_k^{(t)\text{masked}}, & \text{otherwise} \end{cases} \tag{7}$$

The process runs for 3–5 rounds with gradually relaxed thresholds and stops when outputs stabilize ($A_k^{(t+1)} = A_k^{(t)}$) or when the maximum iteration is reached.

Through this iterative process, the system refines character selection to balance information compression and semantic retention. By introducing dynamic, evolutionary strategies, SPDA overcomes the one-shot limitation of traditional methods and better reflects the cognitive process of human abbreviation formation.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We conducted experiments on two publicly available datasets. The first dataset, denoted as D_1, was constructed by Wang et al. [8] based on the large-scale Chinese knowledge graph CN-DBpedia and Baidu Baike, the largest online Chinese encyclopedia. The second dataset, denoted as D_2, was proposed by Zhang and Sun [7] and is sourced from the People's Daily corpus and the SIGHAN Chinese word segmentation corpus. It should be noted that some entries in D_2 do not strictly adhere to the standard full form–abbreviation format. Detailed statistics for both datasets are presented in Table 4.

Table 4: Statistics of datasets D_1 and D_2 used in our experiments

Category	D_1			D_2		
	Train	Valid	Test	Train	Valid	Test
Total entries	75,724	9466	9465	7551	1078	2157
Negative full form	–	–	–	1828	255	578
Avg. length of full forms	9.6	9.7	9.6	5.6	5.6	5.7
Avg. length of abbreviations	4.8	4.8	4.8	2.5	2.5	2.5
Avg. length of contexts	162	161.7	162.6	131.9	132.5	132.3
Max. length of contexts	1125	699	677	373	490	419

4.1.2 Baseline Methods

To evaluate the performance of our method, we adopt mainstream models in Chinese abbreviation generation and related tasks as baselines, covering two categories: sequence labeling and sequence generation. The former frames abbreviation prediction as a character-level tagging task to extract key information from full forms, while the latter treats it as a text-to-text generation task, typically leveraging encoder–decoder architectures or prompt-based language models to directly produce candidates. Detailed results are reported in Table 5.

Table 5: Performance comparison of different models on D_1 and D_2

Models	D_1		D_2	
	Hit@1	**Hit@3**	**Hit@1**	**Hit@3**
Sequence labeling models				
CRF	39.4	50.6	56.7	69.4
RNN-RADD	42.5	65.4	52.6	68.6
Bi-LSTM+CRF	43.4	66.6	57.3	64.9
BERT+Bi-LSTM+CRF	44.9	68.8	58.3	68.8
HJCL	51.3	71.8	58.4	82.6
Sequence generation models				
Seq2Seq-CADC	47.6	71.6	57.6	73.6
CPT	53.2	75.1	61.4	83.3
ChatGLM+CoT	36.9	61.3	40.7	64.2
SimpleBart	47.2	73.0	61.6	80.2
ChatGLM-6b+Dis	56.5	77.9	66.6	85.5
Our method	71.65	76.27	79.35	83.83

CRF [2]: Formulates abbreviation generation as a sequence labeling task using a classical CRF model that leverages handcrafted linguistic features.

RNN-RADD [24]: Incorporates a recursive neural framework with a dynamic dictionary to better model contextual dependencies and enhance abbreviation generation performance.

Bi-LSTM+CRF [7]: Combines Bi-LSTM with CRF to capture long-range dependencies and structured output constraints for sequence labeling in abbreviation tasks.

BERT+Bi-LSTM+CRF [25]: Integrates pre-trained BERT representations with Bi-LSTM and CRF layers to enrich semantic encoding and improve sequence labeling accuracy.

HJCL [26]: Utilizes a hierarchical joint contrastive learning framework with instance-level and label-level losses to boost accuracy and discriminability in abbreviation labeling.

Seq2Seq-CADC [8]: Implements a multi-level pre-training strategy within a sequence-to-sequence architecture to enhance generalization and representation quality.

CPT [18]: Incorporates context during inference only, aligning generation with original input to enhance evaluation accuracy without contextual training.

ChatGLM+CoT [27]: Introduces intermediate reasoning steps guided by abbreviation rules during generation to improve accuracy and interpretability.

SimpleBart [28]: Leverages the BART framework with continuous pre-training to refine the fluency and quality of generated abbreviations.

ChatGLM-6b+Dis [22]: Combines a type discriminator with a fine-tuned LLM for candidate generation, followed by joint scoring to assess rationality.

4.2 Evaluation Metrics and Implementation Details

Evaluation metrics: We adopt Hit@1 and Hit@3 as evaluation metrics. Hit@1 measures whether the top-ranked abbreviation exactly matches the ground truth, while Hit@3 evaluates if the correct abbreviation appears within the top three candidates, comprehensively assessing both accuracy and diversity. To ensure fair comparison, all models are trained and evaluated under identical settings, and the same random seeds are used across experiments for reproducibility.

Implementation details: Our approach is implemented using PyTorch and runs on an NVIDIA RTX 4090 GPU. The candidate generation stage employs CPT-base (12 encoder layers, 6 decoder layers, 768-dimensional hidden states, 12 attention heads), pre-trained on the M2E dataset for 10 epochs (learning rate 2e−5, batch size 64) and fine-tuned on the abbreviation dataset for 10 epochs. The iterative optimization stage uses a pre-trained Chinese language model with 12 Transformer layers, trained for 5 epochs (learning rate 3e−5, AdamW optimizer). During inference, beam search with width 12 balances diversity and computational efficiency.

4.3 Overall Performance

The overall performance of all comparative methods is shown in Table 5. Our conclusions are: (1) our method significantly outperforms previous approaches with 15.15% and 13.01% improvements on Hit@1 for both datasets, while maintaining comparable Hit@3 performance, demonstrating its effectiveness; (2) the dual-layer context enhancement mechanism significantly improves performance by combining definitional and contextual information, leading to more accurate and reasonable candidate generation; (3) the SPDA iterative optimization mechanism enhances candidate ranking quality by better balancing information compression and semantic preservation compared to traditional probability-based methods.

4.4 Impact of Candidate Number

We further analyze the impact of candidate generation strategies by varying the candidate number k from 1 to 30. As shown in Fig. 4, when $k < 10$, Hit@k on D_1 and D_2 increases by 45% and 82.5%, respectively, showing significant improvements. However, when $k > 10$, the performance gain gradually flattens. At $k = 12$, Hit@k reaches 83.1% on D_1 and 87.8% on D_2, achieving a balance between candidate diversity and quality. Compared to $k = 20$, the decoding time is reduced by 0.2 s on D_1 and 1.0 s on D_2, resulting in a 67% improvement in decoding efficiency with negligible performance loss. Further increasing k leads to marginal accuracy gains but sharply rising computational cost, which is impractical in real-world applications.

Considering the trade-off between performance and efficiency, we finally set $k = 12$ in the GIOA framework. This configuration ensures high accuracy, efficient runtime, and lower memory consumption, providing a stable foundation for subsequent optimization and deployment.

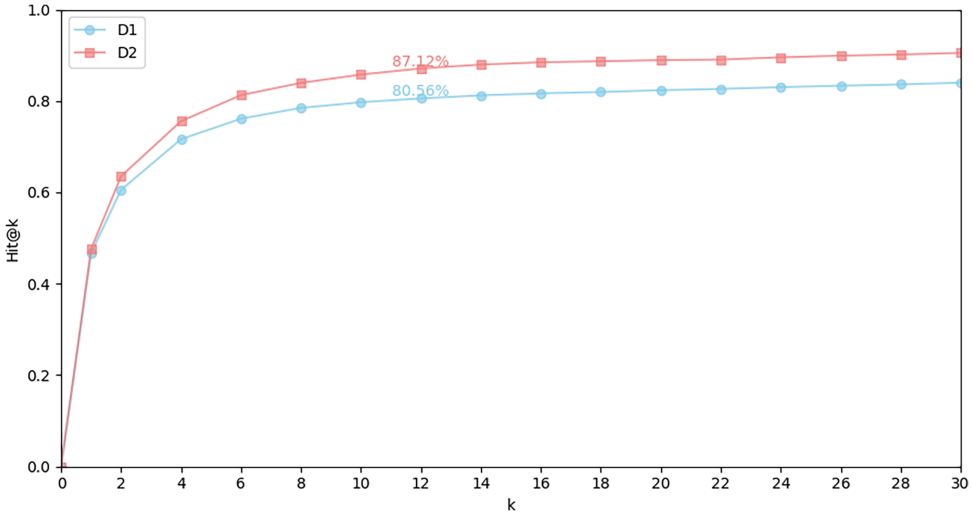

Figure 4: The impact of the number of candidates on the two datasets

4.5 Impact of Context Selection

We conduct a detailed analysis of how different context lengths affect the performance of our abbreviation generation model. Specifically, we test four configurations with maximum context lengths set to 0 (no context), 50, 100, and 150 tokens. When the input exceeds the specified limit, we apply truncation to preserve consistency; otherwise, the full context is retained.

As shown in Table 6, incorporating context information generally leads to consistent performance improvements across both datasets. Compared to the model without context, even a short context window (e.g., $L = 50$) yields a noticeable gain in Hit@1. As context length increases to $L = 100$ and $L = 150$, the improvements continue but gradually stabilize. On D_1, our method achieves a 1.61% increase in Hit@1 compared to $L = 50$, while on D_2, the gain reaches 1.33%. This confirms the importance of contextual information in enabling the model to understand semantic cues and produce more accurate abbreviations. Notably, our final configuration achieves the highest Hit@1 and Hit@3 scores on both datasets, balancing effectiveness and computational efficiency.

Table 6: Results of our models with various context lengths on D_1 and D_2

The length of context	D_1		D_2	
	Hit@1	**Hit@3**	**Hit@1**	**Hit@3**
W/o context	69.50	73.26	76.92	80.75
$L = 50$	70.04	74.27	78.25	81.79
$L = 100$	71.32	74.69	78.95	82.84
$L = 150$	71.37	74.92	79.21	83.46
Our method	71.65	76.27	79.35	83.83

4.6 Impact of Different Models in Generation

In the generation stage of our framework, we adopt a structured generation pipeline based on CPT. To investigate the impact of different generation models on this stage, we replace the generation component

with Qwen2.5, Mengzi-T5-base, and ERNIE-GEN for comparison. The candidate results generated by each model are shown in Table 7. Our method consistently outperforms the alternative models on both datasets during the generation stage, demonstrating stronger generation ability and expression accuracy. This further validates the effectiveness of task-specific modeling in improving abbreviation generation. It is worth emphasizing that our method achieves the highest Hit@1 and Hit@3 scores on both D_1 and D_2, indicating its superior performance in abbreviation generation.

Table 7: Results generated in the generation stage by different models on D_1 and D_2

Models	D_1		D_2	
	Hit@1	**Hit@3**	**Hit@1**	**Hit@3**
Qwen2.5	48.29	60.26	50.92	67.95
Mengzi-T5-base	61.92	68.14	62.31	71.05
ERNIE-GEN	56.29	59.17	58.52	65.39
Our method	63.65	70.69	68.15	76.23

4.7 Comparison of Prompting Strategies

In order to systematically investigate the influence of prompt design on the performance of abbreviation generation, we devise and evaluate a set of representative prompt strategies across two benchmark datasets. Specifically, we examine four configurations, including W/o Prompt, Prompt-only, Template Prompt, and our proposed strategy that integrates contextual information, structural cues, and task-specific semantic signals. Based on these prompt settings, we conducted comparative experiments on D_1 and D_2 (Table 8). The results indicate that well-designed prompts, especially those that combine structural and semantic guidance, can greatly enhance the understanding of task intent and improve the accuracy and reliability of abbreviation generation.

Table 8: Results generated by our method under different prompt strategies across D_1 and D_2

Prompt strategies	D_1		D_2	
	Hit@1	**Hit@3**	**Hit@1**	**Hit@3**
W/o prompt	70.05	74.17	77.94	81.93
Prompt-only	71.25	74.32	78.27	82.59
Template prompt	71.31	75.19	78.94	83.21
Our method	71.65	76.27	79.35	83.83

4.8 Ablation Study

To better understand the contribution of each mechanism in the proposed GIOA framework, we conducted ablation experiments on two public datasets. Specifically, we sequentially removed the context integration, rule-based screening, semantic preservation, and iterative optimization mechanisms to assess their respective impacts on model performance. The results are summarized in Table 9.

Table 9: Ablation results of different modules on D_1 and D_2

Models	D_1		D_2	
	Hit@1	**Hit@3**	**Hit@1**	**Hit@3**
All	71.65	76.27	79.35	83.83
W/o definitional context	69.95	73.82	77.25	81.20
W/o situational context	70.36	74.78	78.27	81.98
W/o two-stage context	69.50	73.26	76.92	80.75
W/o prompt	71.05	74.17	77.94	81.93
w/o rule	71.21	75.32	78.91	83.02
w/o SPDA	70.14	74.12	76.44	80.21
W/o optimize	68.93	73.91	76.04	80.12

The results show that the complete model achieves the best overall performance. Removing the context integration module causes the largest drop, confirming the importance of contextual semantics. Similarly, removing the rule-based screening and SPDA mechanisms degrades performance, indicating their roles in refining candidate ranking and preserving semantic consistency. Overall, these results validate the effectiveness and necessity of each component in the GIOA framework.

5 Discussion and Analysis

(1) Unique Advantages and Innovations: The proposed GIOA method incorporates a dual-layer context enhancement mechanism and SPDA iterative optimization mechanism, effectively addressing semantic preservation and candidate ranking challenges. The significant improvements in Hit@1 metrics (15.15% and 13.01% increases) demonstrate its superior balance between semantic understanding and information compression. The SPDA mechanism successfully integrates semantic integrity with efficient information compression through dynamic adjustment.

(2) Advantages over Baseline Models: GIOA surpasses traditional Seq2Seq and annotation-based methods by effectively combining definitional and contextual information through its dual-layer mechanism. The SPDA optimization process enhances ranking accuracy, producing more practical abbreviations. The model's design aligns with Chinese abbreviation formation patterns, overcoming the limitations of probability-based generation methods.

(3) Limitations and Future Directions: While GIOA excels in Hit@1 performance, improvements are needed for Hit@3 scores. The model faces challenges in computational complexity and domain adaptability. Future work will focus on incorporating external knowledge bases, exploring domain adaptation strategies, and optimizing model structure to enhance practical application efficiency.

6 Conclusion and Future Work

In this paper, we focus on Chinese abbreviation prediction and introduce a novel generation–iterative optimization framework called GIOA, which integrates Chain-of-Thought (CoT) reasoning and Semantic Preservation Dynamic Adjustment (SPDA). Based on this perspective, firstly, we reformulate abbreviation prediction as a reasoning-driven generation task, where CoT prompting guides structured abbreviation generation by incorporating definitional and situational contexts to enhance semantic understanding. Secondly, we implement an iterative optimization mechanism, SPDA, which alternates between character importance evaluation and semantic integrity restoration to refine candidate quality and ranking. Furthermore, we

construct a context-enriched dataset and conduct extensive experiments to validate our framework. The results demonstrate that GIOA achieves significant improvements over state-of-the-art methods on two public datasets, improving Hit@1 by 15.15% and 13.01%, respectively, while maintaining competitive Hit@3 performance. Overall, our findings confirm that combining structured reasoning with dynamic semantic refinement effectively improves both accuracy and interpretability, offering practical insights for future abbreviation generation research.

For future work, we plan to extend GIOA to industrial fault analysis scenarios, where abbreviation generation can help unify terminology and improve the interpretability of fault reports. We are currently constructing a domain-specific dataset that integrates industrial texts and fault records to further explore the applicability of our framework in real-world environments.

Acknowledgement: I sincerely thank my professor, Jianpeng Hu.

Funding Statement: This research is supported by the National Key Research and Development Program of China (2020AAA0109300) and the Shanghai Collaborative Innovation Center of data intelligence technology (No. 0232-A1-8900-24-13).

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Jingru Lv; Supervision: Jianpeng Hu; Data collection and validation: Jingru Lv, Yonghao Luo; Writing—original draft preparation: Jingru Lv; Writing—review and editing: Jianpeng Hu, Jin Zhao. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study are publicly available. The D1 dataset was constructed based on CN-DBpedia (http://kw.fudan.edu.cn/cndbpedia/) (accessed on 25 November 2025) and Baidu Baike. The D2 dataset can be accessed at https://github.com/zhangyics/Chinese-abbreviation-dataset (accessed on 25 November 2025).

Ethics Approval: This study did not involve human participants, animal subjects, or sensitive data collection requiring ethical approval.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Sun X, Li W, Meng F, Wang H. Generalized abbreviation prediction with negative full forms and its application on improving Chinese web search. In: Mitkov R, Park JC, editors. Proceedings of the Sixth International Joint Conference on Natural Language Processing; 2013 Oct 14–18; Nagoya, Japan. p. 641–7.
2. Yang D, Yc Pan, Furui S. Automatic Chinese abbreviation generation using conditional random field. In: Ostendorf M, Collins M, Narayanan S, Oard DW, Vanderwende L, editors. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers; 2009 May 31–Jun 5; Boulder, Colorado. Stroudsburg, PA, USA: Association for Computational Linguistics; 2009. p. 273–6.
3. Chen H, Zhang Q, Qian J, Huang X. Chinese named entity abbreviation generation using first-order logic. In: Mitkov R, Park JC, editors. Proceedings of the Sixth International Joint Conference on Natural Language Processing; 2013 Oct 14–18; Nagoya, Japan. p. 320–8.
4. Zhang L, Li S, Wang H, Sun N, Meng X. Constructing Chinese abbreviation dictionary: a stacked approach. In: Kay M, Boitet C, editors. Proceedings of COLING 2012; 2012 Dec 8–15; Mumbai, India. p. 3055–70.
5. Chang JS, Lai YT. A preliminary study on probabilistic models for Chinese abbreviations. In: Proceedings of the Third SIGHAN Workshop on Chinese Language Processing; 2004 Jul 25; Barcelona, Spain. Stroudsburg, PA, USA: Association for Computational Linguistics; 2004. p. 9–16.

6. Zhang L, Li L, Wang H, Sun X. Predicting Chinese abbreviations with minimum semantic unit and global constraints. In: Moschitti A, Pang B, Daelemans W, editors. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. Stroudsburg, PA, USA: Association for Computational Linguistics; 2014. p. 1405–14.
7. Zhang Y, Sun X. A Chinese dataset with negative full forms for general abbreviation prediction. In: Calzolari N, Choukri K, Cieri C, Declerck T, Goggi S, Hasida K, et al., editors. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); 2018 May 7–12; Miyazaki, Japan.
8. Wang C, Liu J, Zhuang T, Li J, Liu J, Xiao Y, et al. A sequence-to-sequence model for large-scale Chinese abbreviation database construction. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22; 2022 Feb 21–25; Virtual. New York, NY, USA: Association for Computing Machinery; 2022. p. 1063–71. doi:10.1145/3488560.3498430.
9. Torres J, Vaca C, Terán L, Abad CL. Seq2Seq models for recommending short text conversations. Expert Syst Appl. 2020;150:113270. doi:10.1016/j.eswa.2020.113270.
10. Zeng A, Liu X, Du Z, Wang Z, Lai H, Ding M, et al. GLM-130B: an open bilingual pre-trained model. arXiv:2210.02414. 2023.
11. Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models. arXiv:2201.11903. 2023.
12. Camacho-Collados J, Pilehvar MT. From word to sense embeddings: a survey on vector representations of meaning. arXiv:1805.04032. 2018.
13. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using siamese BERT-Networks. In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 3982–92.
14. Zhang B, Li C. Advancing semantic textual similarity modeling: a regression framework with translated ReLU and smooth K2 loss. In: Al-Onaizan Y, Bansal M, Chen YN, editors. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2024. p. 11882–93.
15. Nogueira R, Jiang Z, Pradeep R, Lin J. Document ranking with a pretrained sequence-to-sequence model. In: Cohn T, He Y, Liu Y, editors. Findings of the Association for Computational Linguistics: EMNLP 2020; 2020 Nov 16–20; Online. Stroudsburg, PA, USA: Association for Computational Linguistics; 2020. p. 708–18.
16. Santhanam K, Khattab O, Saad-Falcon J, Potts C, Zaharia M. ColBERTv2: effective and efficient retrieval via lightweight late interaction. In: Carpuat M, de Marneffe MC, Ruiz Meza IV, editors. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2022 Jul 10–15; Seattle, WA, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2022. p. 3715–34.
17. Ren R, Wang Y, Zhou K, Zhao WX, Wang W, Liu J, et al. Self-calibrated listwise reranking with large language models. In: Proceedings of the ACM on Web Conference 2025, WWW '25; 2025 Apr 28–May 2; Sydney, NSW, Australia. New York, NY, USA: Association for Computing Machinery; 2025. p. 3692–701.
18. Tong H, Xie C, Liang J, He Q, Yue Z, Liu J, et al. A context-enhanced generate-then-evaluate framework for Chinese abbreviation prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22; 2022 Oct 17–21; Atlanta, GA, USA. New York, NY, USA: Association for Computing Machinery; 2022. p. 1945–54. doi:10.1145/3511808.3557219.
19. Wu Y, Liu X, Feng Y, Wang Z, Zhao D. Jointly learning entity and relation representations for entity alignment. arXiv:1909.09317. 2019.
20. Gu X, Chen X, Lu P, Li Z, Du Y, Li X. AGCVT-prompt for sentiment classification: automatically generating chain of thought and verbalizer in prompt learning. Eng Appl Artif Intell. 2024;132:107907. doi:10.1016/j.engappai.2024.107907.
21. Wang C, Liu X, Chen Z, Hong H, Tang J, Song D. DeepStruct: pretraining of language models for structure prediction. arXiv:2205.10475. 2023.

22. Liu J, Tian X, Tong H, Xie C, Ruan T, Cong L, et al. Enhancing Chinese abbreviation prediction with LLM generation and contrastive evaluation. Inf Process Manag. 2024;61(4):103768. doi:10.1016/j.ipm.2024.103768.
23. Cao K, Yang D, Liu J, Liang J, Xiao Y, Wei F, et al. A context-enhanced transformer with abbr-recover policy for chinese abbreviation prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22; 2022 Oct 17–21; Atlanta, GA, USA. New York, NY, USA: Association for Computing Machinery; 2022. p. 2944–53. doi:10.1145/3511808.3557074.
24. Zhang Q, Qian J, Guo Y, Zhou Y, Huang X. Generating abbreviations for Chinese named entities using recurrent neural network with dynamic dictionary. In: Su J, Duh K, Carreras X, editors. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; 2016 Nov 1–5; Austin, TX, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2016. p. 721–30.
25. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T, editors. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); 2019 Jun 2–7; Minneapolis, MN, USA. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 4171–86.
26. Yu S, He J, Gutiérrez-Basulto V, Pan JZ. Instances and labels: hierarchy-aware joint supervised contrastive learning for hierarchical multi-label text classification. In: Bouamor H, Pino J, Bali K, editors. Findings of the Association for Computational Linguistics: EMNLP 2023; 2023 Dec 6–10; Singapore. Stroudsburg, PA, USA: Association for Computational Linguistics; 2023. p. 8858–75.
27. Yang L, Wang Z, Li ZX, Na JC, Yu J. An empirical study of multimodal entity-based sentiment analysis with ChatGPT: improving in-context learning via entity-aware contrastive learning. Inf Process Manag. 2024;61:103724. doi:10.1016/j.ipm.2024.103724.
28. Sun R, Xu W, Wan X. Teaching the pre-trained model to generate simple texts for text simplification. In: Rogers A, Boyd-Graber J, Okazaki N, editors. Findings of the Association for Computational Linguistics: ACL 2023; 2023 Jul 9–14; Toronto, ON, Canada. Stroudsburg, PA, USA: Association for Computational Linguistics; 2023. p. 9345–55.