A Hybrid Extraction Framework for Job Postings Based on XPath and NER

Yongyi Lin[1,*], Youjiang Zhou[1], Xiaoxu Wei[2], Hao Sun[1]
[1] *School of Intelligent Media Engineering, Communication University of China Nanjing, Nanjing, China*
[2] *Department of Information and Intelligent Engineering, Shanghai Publishing and Printing College, Shanghai, China*
**Corresponding Author*

Abstract: With the rapid growth of online recruitment platforms, numerous unstructured job postings are dispersed across websites. Diverse page layouts and unstructured job entity descriptions greatly hinder automated information extraction. Rule-based approaches feature high operational efficiency yet poor generalization capability, whereas NER models excel at semantic comprehension but incur substantial computational overhead. This paper proposes a two-stage hybrid extraction framework. In the first stage, XPath rules are adopted to achieve fast regional positioning, while the second stage employs a BERT-BiLSTM-CRF model to conduct fine-grained entity recognition covering corporate names, job positions and salary intervals. Experimental results demonstrate that the proposed framework achieves superior performance compared with baseline methods in terms of precision, recall and F1-score, which significantly enhances the efficiency of recruitment information mining and intelligent talent-job matching.

Keywords: XPath; Named Entity Recognition; Job Postings; Information Extraction; Deep Learning

1. Introduction

With the rapid development of online recruitment platforms, vast amounts of job postings are distributed across various websites in heterogeneous formats [1]. These postings hold significant value for labor market analysis, talent supply-demand forecasting, and occupational career planning. Nevertheless, the multi-source and heterogeneous characteristics of recruitment data impose considerable difficulties for automated information extraction. Various recruitment platforms adopt disparate page layouts and data formatting schemes, while critical entities including skill requirements and working experience are frequently embedded in unstructured descriptive texts, thereby impeding straightforward structured parsing [2].

Web information extraction technology provides a feasible solution to the aforementioned challenges. Rule-based approaches employ XPath, CSS selectors, or regular expressions to locate target elements, yielding high execution efficiency. Nevertheless, these rules demand manual coding and demonstrate limited generalization capabilities. Named Entity Recognition (NER) is proficient in identifying specific entity types within unstructured text, and the recent integration of deep learning models has remarkably enhanced its recognition performance. However, relying solely on NER models to process entire job posting pages results in high computational overhead and introduces redundant processing for fields that are already labeled with explicit structural tags.

To address the aforementioned challenges, this paper proposes a two-stage recruitment information extraction method that integrates XPath with NER. This approach follows the principle of "coarse-grained locating, fine-grained extraction": the first stage rapidly locates target regions using XPath, while the second stage applies an NER model to identify entities such as company names, job titles, and salary ranges. This effectively combines the efficiency of rule-based methods with the generalization capability of learning-based approaches.

2. Related Work

2.1 Web Information Extraction

Web information extraction refers to the

automatic identification and retrieval of target data from semi-structured or unstructured web documents [2]. Based on different technical approaches, existing methods can be broadly categorized into: rule-based methods, visual-based methods, statistical-based methods, and machine learning and deep learning-based methods.

Rule-based methods were the earliest widely adopted techniques for web information extraction, mainly encompassing XPath, CSS selectors, and regular expressions. These methods exhibit high execution efficiency and extraction accuracy, achieving excellent performance when web page structures remain stable. However, their rules rely on manual coding, which leads to poor generalization capabilities. When web page structures change, the rules have to be re-adjusted, resulting in high maintenance costs. For complex or dynamically generated webpages, the difficulty of rule writing also increases significantly.

Visual-based methods extract information by exploiting the visual features rendered from web pages. A representative example is the VIPS algorithm [3], which partitions a web page into semantically coherent blocks based on its visual layout. Other approaches, such as VisualExtract, further combine visual cues with DOM structural features to enhance extraction performance. Although these methods are robust to structural variations in web pages, they entail considerable computational cost and perform poorly on text-dense pages.

Statistical methods perform information extraction by analyzing the statistical characteristics of a web page and employing statistical algorithms to locate the main text. While these methods are easy to implement and free from the need for hand-crafted rules, their applicability is largely restricted to text-rich pages, limiting their generalizability [4].

Traditional machine learning approaches model information extraction as sequence labeling or classification tasks, with representative techniques including Hidden Markov Models (HMMs) [5], Conditional Random Fields (CRFs) [6], and Support Vector Machines (SVMs) [7]. In recent years, deep learning methods have achieved significant progress: CNNs automatically extract local features from text and DOM structures; RNNs and LSTMs capture long-range dependencies within sequences [8]; Transformers yield more expressive feature representations through self-attention mechanisms; and Graph Neural Networks (GNNs) naturally model the hierarchical structure of DOM trees. Compared with traditional methods, deep learning approaches exhibit superior generalization capabilities and robustness, albeit at the cost of substantial computational resources.

2.2 Named Entity Recognition

Named Entity Recognition (NER) is a fundamental task in natural language processing that aims to identify entities of predefined categories—such as person names, locations, and organizations—from unstructured text. The development of NER techniques has evolved from rule-based approaches through statistical methods to modern deep learning-based paradigms.

Early NER methods relied primarily on manually crafted rules and domain-specific dictionaries. Although effective within specific domains, these approaches suffered from limited generalizability and high maintenance costs [9]. Statistical machine learning methods subsequently rose to prominence, including Hidden Markov Models (HMMs), Maximum Entropy Markov Models (MEMMs), and Conditional Random Fields (CRFs) [9]. Among these, CRFs have demonstrated outstanding performance on NER tasks owing to their ability to jointly model label transition probabilities and observation features, and they remain widely adopted to this day.

In recent years, deep learning methods have achieved breakthrough progress on NER tasks. The BiLSTM-CRF model combines a bidirectional long short-term memory network with a CRF layer, enabling the automatic learning of sequential textual features while explicitly modeling dependencies among labels. The emergence of pre-trained language models, notably Bidirectional Encoder Representations from Transformers (BERT), has further advanced NER performance. By pre-training on large-scale corpora, BERT acquires rich linguistic knowledge and provides powerful contextualized representations for downstream NER tasks.

3. Hybrid Extraction Framework Design

3.1 Framework Architecture

The hybrid extraction framework proposed in

this paper employs a two-stage strategy of coarse-grained location and fine-grained extraction. It primarily consists of six main modules: data collection, web page preprocessing, XPath location, text extraction, NER extraction, and post-processing.

The text extraction module extracts plain text content from the located DOM nodes, performing sentence segmentation and word segmentation. The NER extraction module employs a BERT-BiLSTM-CRF model for named entity recognition, extracting key entities relevant to the recruitment domain. The post-processing module validates, duplicates, and formats the NER results, outputting the final structured recruitment information.

The data collection module is responsible for retrieving raw HTML pages from job recruitment websites. Built on a distributed crawler architecture, it enables parallel collection across multiple platforms and incorporates anti-crawling countermeasures. The web page preprocessing module cleans and normalizes the raw HTML through encoding conversion, tag repair, and the removal of comments and scripts. The XPath localization module applies predefined XPath rules to identify target regions containing job postings within the preprocessed HTML, thereby filtering out irrelevant content. The text extraction module then extracts plain text from the localized DOM nodes and performs sentence and word segmentation. The NER extraction module employs a BERT-BiLSTM-CRF model to recognize named entities relevant to the recruitment domain. Finally, the post-processing module validates, deduplicates, and formats the NER results to produce the structured recruitment information.

3.2 XPath Locator Module Design

The XPath localization module is a key component for achieving coarse-grained positioning. It is designed to accurately and efficiently locate the core content regions containing job details across different recruitment platforms, while filtering out irrelevant elements such as navigation bars, sidebars, advertisements, and footers.

3.2.1 Rule library construction

For mainstream recruitment platforms, this paper constructs a multi-level XPath rule repository organized in a three-tier hierarchical structure: "Platform–Page Type–Field". The first tier (Platform Level) defines entry rules for different recruitment platforms to identify the platform to which a given webpage belongs. The second tier (Page Type Level) defines layout rules for different page types within the same platform (e.g., listing pages and detail pages) to locate the relevant content regions. The third tier (Field Level) defines extraction rules for specific fields to retrieve particular information items.

3.2.2 Rule matching strategies

In practice, a single page may match multiple rules simultaneously. To handle this, the proposed method adopts a priority-based matching strategy: platform-specific rules are attempted first, and if no match is found, the system falls back to general-purpose rules; when multiple rules match concurrently, the one with the highest priority is selected. Rule priorities are dynamically adjusted based on their specificity and historical matching success rates. In addition, to address the issue of rules becoming obsolete due to changes in webpage structure, this paper proposes a mechanism for automatically updating rules. The system periodically validates the rule database, flags obsolete rules, and automatically generates candidate rules for manual review by analyzing new webpage samples.

3.3 Rule Matching Strategies

The NER extraction module is responsible for identifying named entities in the recruitment domain from text segments localized via XPath. This paper adopts a BERT-BiLSTM-CRF architecture, which has been specifically optimized for the recruitment domain.

3.3.1 Model architecture

The BERT-BiLSTM-CRF model used in this paper primarily consists of the following layers:

(1) Input layer

The input layer is responsible for converting raw text into a format that the model can process. For an input text sequence $X = \{x_1, x_2, ..., x\_n\}$, the text is first tokenized to produce a sequence of tokens. Then, special tokens [CLS] and [SEP] are added to the beginning and end of the sequence, respectively.

(2) BERT Encoding Layer

The BERT encoding layer uses a pre-trained Chinese BERT model (bert-base-chinese) as a feature extractor. BERT encodes the input sequence using a multi-layer Transformer encoder, outputting context-relevant

representation vectors for each position:

$$H^{BERT}=BERT(X)\in\mathbb{R}^{n\times d_{bert}} \quad (1)$$

Here, d_{bert} refers to the dimension of BERT's hidden layers (768 dimensions).

(3) BiLSTM Layer

The BiLSTM layer is used to further capture long-range dependencies in the sequence. The output of BERT is fed into a bidirectional LSTM network:

$$\overrightarrow{h_t}=LSTM\left(\overrightarrow{h_{t-1}},H_t^{BERT}\right) \quad (2)$$

$$\overleftarrow{h_t}=LSTM\left(\overleftarrow{h_{t+1}},H_t^{BERT}\right) \quad (3)$$

$$h_t=\left[\overrightarrow{h_t};\overleftarrow{h_t}\right] \quad (4)$$

A forward LSTM processes the sequence from left to right, while a backward LSTM processes it from right to left; the hidden states from both directions are concatenated to obtain the final feature representation.

(4) CRF Layer

The CRF layer is used to perform global optimization on the label sequence and to model transition constraints between adjacent labels. Given a feature sequence $H = \{h_1, h_2, ..., h\_n\}$ and a label sequence $Y = \{y_1, y_2, ..., y\_n\}$, the conditional probability defined by the CRF is:

$$P(Y|H)=\frac{\exp(Score(H,Y))}{\sum_{Y'}\exp(Score(H,Y'))} \quad (5)$$

The scoring function is defined as follows:

$$Score(H,Y)=\sum_{i=1}^{n}\left(A_{y_{i-1},y_i}+E_{y_i,h_i}\right) \quad (6)$$

Let A be the label transition matrix and E be the emission matrix. During training, the log-likelihood of the correct label sequence is maximized; during prediction, the Viterbi algorithm is used to decode the optimal label sequence.

3.3.2 Domain-specific optimization

To improve the model's recognition performance in the recruitment domain, the following optimizations are implemented:

(1) Domain Pre-training

Building upon the general-purpose BERT model, we conduct further pre-training on a large-scale recruitment corpus consisting of approximately one million job descriptions collected from major recruitment websites. Through domain-specific pre-training, the model learns the vocabulary and expression patterns characteristic of the recruitment domain.

(2) Multi-feature fusion

Drawing on the work of Yang et al. [10], we introduce a lexical feature enhancement mechanism. Specifically, a recruitment-domain dictionary is constructed, containing entries such as company names, job titles, and skill names. During the encoding stage, dictionary matching results are used to inject lexical boundary features into the input sequence, thereby helping the model identify entity boundaries more accurately.

(3) Tag constraints

Domain-specific tagging constraints are incorporated into the CRF layer. For example, an education entity cannot immediately follow a salary entity, whereas skill entities may appear consecutively. These constraints are enforced by setting the scores of invalid transitions in the transition matrix to negative infinity.

4. Experiment

4.1 Dataset

Current publicly accessible corpora for job posting named entity recognition remain insufficient. To address this gap, this paper develops a task-specific annotated dataset for job posting information extraction.

We construct a dataset of 5,000 annotated samples collected from mainstream platforms such as Zhaopin and Boss Zhipin, covering seven entity types with an annotation consistency of 0.89.

4.2 Evaluation Criteria

We use precision (P), recall (R), and F1 score as evaluation metrics:

$$P=\frac{TP}{TP+FP} \quad (7)$$

$$R=\frac{TP}{TP+FN} \quad (8)$$

$$F1=\frac{2\times P\times R}{P+R} \quad (9)$$

Here, TP represents the number of correctly identified entities, FP represents the number of incorrectly identified entities, and FN represents the number of missed entities.

4.3 Baseline Method

To comprehensively evaluate the effectiveness of our method, we selected the following baseline methods for comparison:

(1) Pure rule-based method: Performs information extraction using only XPath rules, without a NER model.

(2) BiLSTM-CRF: A classic sequence labeling model that uses randomly initialized word vectors.

(3) BERT-CRF: Uses BERT for feature encoding, followed directly by a CRF layer for label prediction.

(4) BERT-BiLSTM-CRF (General): Uses a general-purpose pre-trained BERT without domain adaptation.
(5) Our Method: A complete model utilizing domain-specific pre-trained BERT and multi-feature fusion.

4.4 Experimental Results

Experimental results show that our method achieves an F1 score of 92.3%, surpassing the strongest baseline by 2.5 percentage points. It delivers the best performance across all seven entity types, with the highest recognition accuracy observed for salary ranges and work locations, while skill requirements prove the most challenging due to the diversity of their expressions. Ablation studies confirm the effectiveness of domain-specific pre-training, BiLSTM layers, lexical features, and label constraints, among which domain-specific pre-training and BiLSTM layers contribute most substantially.

Efficiency analysis indicates that the proposed framework effectively reduces the text volume processed by the NER model through XPath pre-filtering, achieving a favorable balance between recognition accuracy and processing efficiency.

5. Conclusions

To address the limitations of single-method approaches in job posting information extraction, this paper proposes a hybrid framework that integrates XPath rules with NER. The framework adopts a two-stage strategy of "coarse-grained localization followed by fine-grained extraction": in the first stage, XPath rules are applied to efficiently locate target regions containing job postings; in the second stage, a BERT-BiLSTM-CRF-based NER model identifies seven entity types, including company names, job titles, and salary ranges. By incorporating domain-specific pre-training, lexical feature fusion, and label constraints, the proposed framework substantially improves the recognition accuracy of job-related entities.

References

[1] Ren J. Z. A Study on Text Classification of Job Postings Based on Deep Learning. Journal of Hubei University of Arts and Sciences, 2023, 44(11): 21–27.
[2] Guo W.J, Lv N., Ji S.J, et al. Recruitment information extraction model based on parallel multi-scale features learning. Journal of Shandong University of Science and Technology, 2025, 44 (03): 97-106.
[3] Song R. International Conference on World Wide Web: Learning block importance models for web pages. ACM, 2004:203-211.
[4] Zhou Y., Yin X.J and Yan J.C. RETRACTION: An Information Extraction Method Based on Improved Mixed Text Density Web Pages. Expert Systems, 2024, 42(2): e13796-e13796.
[5] Freitag D, Mccallum A. Information extraction with HMMs and shrinkage. Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction. 1999: 31-36.
[6] Lafferty J, Mccallum A, Pereira FCN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the 18th International Conference on Machine Learning. 2001: 282-289.
[7] Isozaki H, Kazawa H. Efficient support vector classifiers for named entity recognition. Proceedings of the 19th International Conference on Computational Linguistics. 2002: 1-7.
[8] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Net works, 2005, 18(5-6):602-610.
[9] Zhang M.G, Wei G.H. A Review of Information Extraction Research in the Medical Field. Computer Engineering and Applications, 2026:1-38.
[10] Yang S.Y, Li G.H, Dong J, et al. AGMF-NER: A Chinese Named Entity Recognition Model with Adaptive Feature Fusion. Computer Engineering and Applications, 2026:1-17.

