2023 IEEE 39th International Conference on Data Engineering (ICDE) | 979-8-3503-2227-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDE55515.2023.00242

ResuFormer: Semantic Structure Understanding for Resumes via Multi-Modal Pre-training

Kaichun Yao[1], Jingshuai Zhang[1], Chuan Qin[*,1], Xin Song[1], Peng Wang[1],
Hengshu Zhu[*,†,2] and Hui Xiong[3]
[1]Baidu Talent Intelligence Center, Baidu Inc., [2]Career Science Lab, BOSS Zhipin,
[3]Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou)
yaokaichun@outlook.com, {chuanqin0426, zhuhengshu}@gmail.com,
{zhangjingshuai, songxin06, wangpeng40}@baidu.com, xionghui@ust.hk

***Abstract*—Understanding the semantic structure of resumes plays an important role for various intelligent recruitment related applications. However, due to the unique characteristics of resume documents (e.g., diverse writing styles and multi-page) and the lack of labeled data, it has been a long-standing challenge to effectively extract the structural information of resumes through machine learning models. While considerable efforts have been made in this direction, existing methods only focus on the textual information in the document where the rich multi-modal information (e.g., the visual and layout information) is largely ignored. To this end, in this paper, we propose ResuFormer for understanding the semantic structure of resumes. Specifically, ResuFormer focuses on two typical tasks in this direction, namely resume block classification and intra-block information extraction respectively. For the first task, we propose a multi-modal pre-training model with a hierarchical Transformer encoder, in which we design three self-supervised training objectives, i.e., masked layout-language model, self-supervised contrastive learning and dynamic next-sentence prediction, to pre-train the model parameters, and fine-tune the model only using a small amount of training data. For the second task, we introduce a self-distillation based self-training learning framework to make the distantly supervised model more robust to the noise data. Finally, extensive experiments conducted on real-world resume datasets have clearly validated the performance of our ResuFormer compared with state-of-the-art (SOTA) baselines.**

***Index Terms*—semantic structure extraction, multimodal pre-training, distant supervision, self-training**

I. Introduction

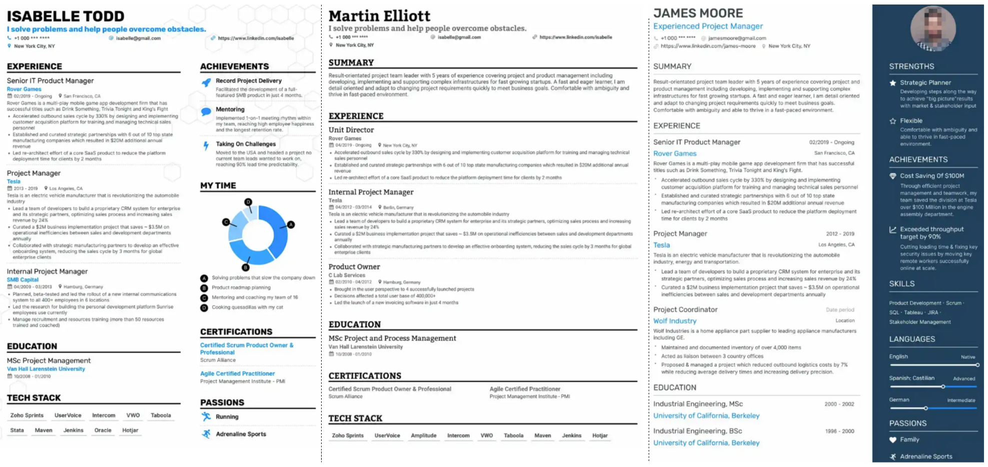
With the rapid development of online recruitment service, the number of electronic resumes has soared. How to efficiently and accurately process these resumes into the structured information becomes a crucial issue that will be beneficial for many downstream applications, such as person-job matching [1, 2, 3, 4], talent identification [5, 6], talent evaluation [7, 8] and job description generation [9].

The resumes are different from other documents like the form and the receipt. They are more text-centric and have more flexible formats with rich semantic structure. Generally, a resume can be segmented into several types of semantic blocks: personal information, education experience, work&project experience, summary and awards, and each block has the corresponding specific information. For example, personal information block often contains name, email, phone number and age, and education experience block often consists of college, major, degree, start date and end date. It is worth noting that education, work and project experience are usually more than one in a resume. We present three styles of resume templates in Figure 1. It is observed that the three resumes have different writing styles and each resume contains several types of text blocks with specific semantic information.

Existing studies for resume semantic structure understanding can be grouped into two categories: resume entity extraction and resume semantic structure extraction. The entity extraction methods only focus on extracting key entities such as name, email and school from resumes. Although lots of named entity recognition (NER) methods [10, 11, 12, 13] achieve good performance on resume entity recognition, They fail to understand the hierarchical semantic structure of resume. For example, a resume usually contains multiple work experiences and each work experience consists of word date, company name, job position and work content. It is hard for the NER methods to obtain the hierarchical structure. To solve this problem, semantic structure extraction methods [14, 15] are proposed to extract the structured information from resumes. Specifically, first, neural networks based models are deployed to segment resume into text blocks and then the typical NER methods are used to recognize entities in each segmented block. However, these approaches only leverage the textual information while ignore the rich multi-modal information (i.e., the visual information and layout information) naturally in the resume document. They achieve the limited performance.

Recently, Transformer [16] based multi-modal pre-training models [17, 18, 19] are widely applied in document understanding. In the pre-training procedure of model, the self-supervised objectives make the model parameters to be trained enough that largely reduce the dependence on annotated training data. Additionally, except for the textual information, visual information and layout information naturally existed in the document are jointly learned and merged as the final semantic representations that makes these pre-training models achieve promising results on many downstream tasks such as form and receipt understanding, document image classification. and document visual question answering.

*Hengshu Zhu and Chuan Qin are corresponding authors.
†Part of the work was done when the author was employed by Baidu Talent Intelligence Center.



ISABELLE TODD
I solve problems and help people overcome obstacles.
+1 000 *** **** isabelle@gmail.com https://www.linkedin.com/isabelle
New York City, NY

EXPERIENCE

Senior IT Product Manager
Rover Games
02/2019 - Ongoing San Francisco, CA
Rover Games is a multi-play mobile game app development firm that has successful titles such as Drink Something, Trivia Tonight and King's Fight
- Accelerated outbound sales cycle by 330% by designing and implementing customer acquisition platform for training and managing technical sales personnel
- Established and curated strategic partnerships with 6 out of 10 top state manufacturing companies which resulted in $20M additional annual revenue
- Led re-architect effort of a core SaaS product to reduce the platform deployment time for clients by 2 months

Project Manager
Tesla
2013 - 2019 Los Angeles, CA
Tesla is an electric vehicle manufacturer that is revolutionizing the automobile industry
- Lead a team of developers to build a proprietary CRM system for enterprise and its strategic partners, optimizing sales process and increasing sales revenue by 24%
- Curated a $2M business implementation project that saves ~ $3.5M on operational inefficiencies between sales and development departments annually
- Collaborated with strategic manufacturing partners to develop an effective onboarding system, reducing the sales cycle by 3 months for global enterprise clients

Internal Project Manager
SMB Capital
04/2009 - 03/2013 Hamburg, Germany
- Planned, beta-tested and led the rollout of a new internal communications system to all 400+ employees in 6 locations
- Led the research for building the personal development platform Sunrise employees use currently
- Manage recruitment and resources training (more than 50 resources trained and coached)

EDUCATION

MSc Project Management
Van Hall Larenstein University
10/2008 - 01/2010

TECH STACK

Zoho Sprints UserVoice Intercom VWO Taboola
Stata Maven Jenkins Oracle Hotjar

ACHIEVEMENTS

Record Project Delivery
Facilitated the development of a full-featured SMB product in just 4 months.

Mentoring
Implemented 1-on-1 meeting rhythm within my team, reaching high employee happiness and the longest retention rate.

Taking On Challenges
Moved to the USA and headed a project no current team leads wanted to work on, reaching 90% lead time predictability.

MY TIME

A Solving problems that slow the company down
B Product roadmap planning
C Mentoring and coaching my team of 16
D Cooking quesadillas with my cat

CERTIFICATIONS

Certified Scrum Product Owner & Professional
Scrum Alliance

Agile Certified Practitioner
Project Management Institute - PMI

PASSIONS

Running
Adrenaline Sports

Martin Elliott
I solve problems and help people overcome obstacles.
+1 000 *** **** isabelle@gmail.com https://www.linkedin.com/isabelle
New York City, NY

SUMMARY

Result-orientated project team leader with 5 years of experience covering project and product management including developing, implementing and supporting complex infrastructures for fast growing startups. A fast and eager learner, I am detail oriented and adapt to changing project requirements quickly to meet business goals. Comfortable with ambiguity and thrive in fast-paced environment.

EXPERIENCE

Unit Director
Rover Games
04/2019 - Ongoing New York City, NY
- Accelerated outbound sales cycle by 330% by designing and implementing customer acquisition platform for training and managing technical sales personnel
- Established and curated strategic partnerships with 6 out of 10 top state manufacturing companies which resulted in $20M additional annual revenue
- Led re-architect effort of a core SaaS product to reduce the platform deployment time for clients by 2 months

Internal Project Manager
Tesla
04/2012 - 03/2014 Berlin, Germany
Tesla is an electric vehicle manufacturer that is revolutionizing the automobile industry
- Lead a team of developers to build a proprietary CRM system for enterprise and its strategic partners, optimizing sales process and increasing sales revenue by 24%
- Curated a $2M business implementation project that saves ~ $3.5M on operational inefficiencies between sales and development departments annually
- Collaborated with strategic manufacturing partners to develop an effective onboarding system, reducing the sales cycle by 3 months for global enterprise clients

Product Owner
C Lab Services
02/2010 - 04/2012 Hamburg, Germany
- Brought in the user perspective to 4 successfully launched projects
- Decisions affected a total user base of 400,000+
- Led the launch of a new invoicing software in just 4 months

EDUCATION

MSc Project and Process Management
Van Hall Larenstein University
10/2008 - 01/2010

CERTIFICATIONS

Certified Scrum Product Owner & Professional
Scrum Alliance

Agile Certified Practitioner
Project Management Institute - PMI

TECH STACK

Zoho Sprints UserVoice Amplitude Intercom VWO Taboola Maven Jenkins Hotjar

JAMES MOORE
Experienced Project Manager
+1 000 *** **** jamesmoore@gmail.com
https://www.linkedin.com/james-moore New York City, NY

SUMMARY

Result-orientated project team leader with 5 years of experience covering project and product management including developing, implementing and supporting complex infrastructures for fast growing startups. A fast and eager learner, I am detail oriented and adapt to changing project requirements quickly to meet business goals. Comfortable with ambiguity and able to thrive in a fast-paced environment.

EXPERIENCE

Senior IT Product Manager 02/2019 - Ongoing
Rover Games San Francisco, CA
Rover Games is a multi-play mobile game app development firm that has successful titles such as Drink Something, Trivia Tonight and King's Fight.
- Accelerated outbound sales cycle by 330% by designing and implementing customer acquisition platform for training and managing technical sales personnel
- Established and curated strategic partnerships with 6 out of 10 top state manufacturing companies which resulted in $20M additional annual revenue
- Led re-architect effort of a core SaaS product to reduce the platform deployment time for clients by 2 months

Project Manager 2012 - 2019
Tesla Los Angeles, CA
Tesla is an electric vehicle manufacturer that is revolutionizing the automobile industry, energy and transportation.
- Lead a team of developers to build a proprietary CRM system for enterprise and its strategic partners, optimizing sales process and increasing sales revenue by 24%
- Curated a $2M business implementation project that saves ~ $3.5M on operational inefficiencies between sales and development departments annually

Project Coordinator Date period
Wolf Industry Location
Wolf Industries is a home appliance part supplier to leading appliance manufacturers including GE.
- Maintained and documented inventory of over 4,000 items
- Acted as liaison between 3 country offices
- Proposed & managed a project which reduced outbound logistics costs by 7% while reducing average delivery times and increasing delivery precision.

EDUCATION

Industrial Engineering, MSc 2000 - 2002
University of California, Berkeley

Industrial Engineering, BSc 1996 - 2000
University of California, Berkeley

STRENGTHS

Strategic Planner
Developing steps along the way to achieve "big picture"results with market & stakeholder input

Flexible
Comfortable with ambiguity and able to thrive in fast-paced environment.

ACHIEVEMENTS

Cost Saving Of $100M
Through efficient project management and teamwork, my team saved the division at Tesla over $100 Million in the engine assembly department.

Exceeded throughput target by 90%
Cutting loading time & fixing key security issues by moving key remote workers successfully online at scale.

SKILLS

Product Development · Scrum · SQL · Tableau · JIRA · Stakeholder Management

LANGUAGES

English Native
Spanish; Castilian Advanced
German Intermediate

PASSIONS

Family
Adrenaline Sports

Fig. 1. Three different styles of resume templates. All contents are fictional to protect the privacy of candidates.

Although the above pre-training strategies can be applied in the resume semantic structure understanding, there still exists several challenges: First, the resume documents are different from forms and invoices. They are more text-centric, and have more contents with several pages. According to the statistics on our resume dataset, the average number of tokens for resumes is more than 1,600, while the above Transformer architecture based pre-training models are hard to handle more than 512 tokens at once. Although the token by token loop processing strategies can solve this problem, high time costs will affect the efficiency of resume parsing. Second, the various writing styles for different resumes increase the difficulty of semantic structure understanding. As shown in Figure 1, the three resumes have different formats and the semantic blocks randomly appear in different positions in the documents. In addition, work experiences and project experiences often exist in different pages of a document. Third, the available fine-grained annotated training dataset is limited for resume semantic structure understanding. Lack of large scale fine-grained annotation training data will result in the poor performance for resume block classification and intra-block information extraction mentioned above.

To address the above challenges, following the above resume semantic structure extraction, we propose ResuFormer for resume semantic structure understanding, which consists of a multi-modal pre-training model with a hierarchical Transformer for resume block classification and a self-distillation based distantly supervised NER approach for intra-block information extraction. Specifically, we first construct a hierarchical transformer for resume block classification. The hierarchical structure is more suitable for long document understanding. Then, we design three self-supervised objectives, i.e, masked layout-language model, self-supervised contrastive learning and dynamical next-sentence prediction, to achieve more effective representation learning for the multi-modal information. Meanwhile, the effective pre-training strategies largely reduce the dependence on labeled training data. After the pre-training, we fine-tune the model on a small amount of annotated data for resume block classification. In addition, due to lack of fine-grained annotated training data for intra-block information extraction, we construct multiple entity dictionaries to automatically annotate the training data, and also adopted a distantly supervised sequence labeling method for intra-block fine-grained information extraction. Finally, we conduct extensive experiments on real-world resume datasets with two typical tasks, namely resume block classification and intra-block information extraction. The experimental results clearly validate the effectiveness of our approaches compared with a number of SOTA baselines.

The main contributions of this paper can be summarized as follows:

- To the best of my knowledge, we are the first to propose the hierarchical multi-modal pre-training model for long document understanding.
- We propose a distantly supervised sequence labeling method trained in the self-distillation based self-training learning framework, which not only overcome the difficulty of lack of labeled data, but also improve the model robustness against the noise data in the distantly supervised scenario.
- We evaluate our ResuFormer with extensive experiments on real-world resume datasets, which consistently outperform a number of SOTA baselines with a significant margin. Specifically, for the task of resume block classification, our model achieves more than 4% improvements on multiple tags than the SOTA baseline. Meanwhile, our method improves about 15 times for the running efficiency than the SOTA baseline. For the task of intra-block information extraction, our method achieves the best performance on all tags.

II. Related Work

The related works can be grouped into two categories, namely *resume information extraction* and *multi-modal pre-training*.

A. Resume Information Extraction

The rapid development of online recruitment results in an increasing amount of resume data. How to effectively analyze each resume document has attracted much attention of researchers. Existing methods for resume information extraction can be grouped into two categories: resume entity extraction and semantic structure extraction.

Resume entity extraction only focuses on extracting key entities, such as name, email, major, school and company, from the resume text. Early studies on them were mainly machine learning based approaches. The traditional machine learning such as hidden Markov model (HMM), conditional random field (CRF), support vector machine (SVM), were used to construct resume information extraction system [20, 21, 22]. Although these models achieve good performance, they bring the cost of feature engineering. In recent years, deep learning based methods have been used for resume information extraction. The typical named entity recognition method, BiLSTM+CRF, was used to extract resume information [10]. The method first used Word2Vec to initialize the word embedding as inputs. Then it utilized a bidirectional long-short term memory (BiLSTM) layer to learn contextual semantic of a word. Finally, the CRF layer outputted the scores of all possible tag for sequences. Different from the above Word2Vec embeddings, Van et al. [23] used a CNN+BiLSTM+CRF structure for various formats of resume information extraction, which used CNN to learn the representation of character and combined deep learning with heuristic rules to achieve superior results than other models. To enhance the semantic representation, Chen et al. [11] fused the Word2Vec [24] features with BiLSTM contextual features and then combine BiLSTM with CRF to parse Chinese resumes. Due to the powerful semantic representations of pre-training model, Li et al. [12] established a BERT+BiLSTM+CRF model for the resume information extraction, in which BERT [25] was used to extract the deep features of the resume text. To exploit 2D layout information, Wei et al. [26] combined the pre-trained model RoBERTa [27] with graph convolutional network [28] (GCN) to extract resume entity information, in which the GCN is used to encode layout and positional information. Although these approaches achieved promising results on entity extraction. They ignored to exact the hierarchical structured information naturally existed in the resume document.

Semantic structure extraction pipelines resume information extraction as the task of resume block classification and intra-block entity extraction, aiming at extracting a hierarchical structured information from a resume. For example, a resume has multiple work experiences and each work experience contains text information like work content and entities such as work date, company and position. For this kind of extractions, Ayishathahira et al. [14] leveraged convolutional neural network (CNN) to classify different text blocks in resumes and then used BiLSTM+CRF model for resume entity recognition. Zu et al. [15] proposed a two-step pipeline approach, which used a neural network text classifier to segment resume text block and used BiLSTM+CNN+CRF to recognize entities in segmented text blocks. However, these models only leveraged the text in the resume to extract information while ignored visual information and layout information naturally in the resume document. Therefore, they achieved the limited performance. In addition, their model performs the token-level classification and it is hard to handle the document-level semantic structure.

In order to extract the hierarchical structure from the resume document, we follow the typical semantic structure extraction pipeline for resume information extraction. We propose a hierarchical multi-modal pre-training model for resume block classification and present a distantly supervised NER approach with self-distillation based self-training learning framework for intra-block entity extraction.

B. Multimodal Pre-training

Multi-modal [29, 30] pre-training has attracted huge attentions due to its success on vision-language representation learning. In recent years, to learn visual and textual representations in a unified framework, Lu et al. [31] proposed a vision-and-language BERT (ViLBERT), which aims at jointly learning representations of image content and natural language. ViLBERT extended the popular BERT architecture to a multi-modal two-steam model, which used a co-attentional transformer layers to accept both visual and textual information in separate streams as inputs. Li et al. [32] stacked multiple Transformer layers as a VisualBERT, which used a self-attention mechanism to align elements of an input text and regions in an associated input. Su et al. [33] introduced a visual-linguistic BERT (VL-BERT) for visual-linguistic tasks, which used a Transformer model as backbone and took both visual and linguistic embedded features as input. Different from the above simultaneous multi-modal inputs, Chen et al. [34] proposed a universal image-text representation (UNITER) model, which can power heterogeneous downstream V+L tasks with joint multi-modal embeddings. It is noted that the approaches above mostly simply concatenate image region features and text features as input to the model to be pre-trained and utilize self-attention to align image-text semantics. Different from their approach, Li et al. [35] proposed an object-semantics aligned pre-training (OSCAR) method, which used object tags detected in images as anchor points to significantly ease the learning of alignments.

The pre-trained models mentioned above paid more attention to the multi-modal learning of videos or images. In order to better understand visually-rich document, Xu et al. proposed LayoutLM [17], LayoutLMv2 [18] and LayoutXLM [19], respectively. LayoutLM aimed at jointly modeling interactions between text and layout information across scanned document images, LayoutLMv2 is the upgraded version LayoutLM, its pre-trained strategies used not only the existing masked visual-language modeling task but also the new text-image alignment and text-image matching tasks. For LayoutXLM, it is based on LayoutLMv2 for multilingual document understanding. Although these approaches achieve promising results on many document understanding tasks, they can not better handle the long document with abundant textual information. Different from them, our multi-modal pre-trained model adopted a hierarchical structure that can better deal with the long document.

III. Problem Definition

Resume semantic structure understanding usually involves two main tasks, namely resume block classification and intra-block information extraction.

A. Resume Block Classification

Resume block classification aims to segment a resume document into a listed of blocks and predict the corresponding label for each block according to the semantic information. Actually, the process can be considered as a sequence labeling task. Different from the token-level classification, here we adopt a sentence-level classification to better cope with long document.

Formally, given a resume document $\mathcal{D}$, it contains the semantic tags $\mathcal{C} \in \{$*PInfo*, *EduExp*, *WorkExp*, *ProjExp*, *Summary*, *Awards*, *SkillDes*, *Title*$\}$ [1]. We first use an open-source PDF parser called PyMuPDF [2] to obtain a list of discrete token set $\mathcal{D} = \{t_0, t_1, ..., t_n\}$ and the page image list $V = \{v_1, v_2, ..., v_p\}$ from the PDF file, where each token $t_i = (w, (x_0, y_0, x_1, y_1), p)$ consists of a word w, its bounding box coordinates (x_0, y_0, x_1, y_1) and its page number p, n is the number of tokens and v_p is the image data corresponding to the p-th page. Then, according to the bounding box coordinates of each token, we concatenate adjacent tokens in t into multiple sentences $\mathcal{D} = \{s_1, s_2, ..., s_m\}$, where $s_i = (seq, (x'_0, y'_0, x'_1, y'_1), p')$ consisting of a sequence of word $seq = \{w_0, w_1, ..., w_{|s_i|}\}$, the corresponding bounding box coordinates (x'_0, y'_0, x'_1, y'_1) and the page number p', and m is the number of sentences. The concatenation for tokens need to be guaranteed that the two tokens are closely spaced and in a row in the document. Thus, the visual information v_i for each sentence s_i can be obtained according to the image data V and the corresponding bounding box coordinates and the page number. It is worth noting that the sentence s_i is not a real semantically complete sentence. It just consists of a sequence of adjacent tokens and then we merge the coordinates of leftmost token and rightmost token into the bounding box coordinates of s_i. Finally, we predict the tag for each sentence s_i following the typical **IOB** tagging scheme, where "B" denotes the beginning of a tag, "I" denotes the continuation of a tag, and "O" corresponds to tokens that are not part of any tag.

B. Intra-block Information Extraction

Resume block classification can obtain a coarse-grained semantic block structure. As shown in Figure 1, the resume includes multiple semantic blocks such as person information, education experience and work experience. Actually, it is also crucial to extract the fine-grained information in each intra-block. For instance, a personal information block usually includes many entities such as name, email and age, and a work experience block usually includes work duration, company name, job position and work content. In order to extract these fine-grained entities, we also consider the extraction process as a sequence labeling task. Different from the sentence-level resume block classification, it adopts a token-level sequence tagging strategy. Noted that we extract these fine-grained entities from the segmented block instead of the whole document because the token sequence of document is too long.

Formally, given a intra-text block $B = \{w_1, w_2, ..., w_{|B|}\}$, where w_j is the j-th token in the block. Our aim is to predict the entity tagging label for each word in the block.

IV. Methodology

In this section, we first present our model architecture for resume block classification. Then, we introduce intra-block information extraction model with a distantly-supervised training samples.

A. Resume Block Classification

The key training procedure is to construct a powerful hierarchical Transformer encoder as the encoder backbone, followed by pre-training the encoder with self-supervised objectives on a large real-word resume corpus. The next step is to fine-tune the encoder with a block classification layer only using a small amount of labeled data. The architecture of the resume block classification is illustrated in Figure 2.

As shown in Figure 2, for the hierarchical transformer encoder, it consists of a sentence-level encoder and a document-level encoder. The two encoders all encode the text and layout information, while the document-level also introduce the visual information. For the downstream task, we use typical BiLSTM+CRF method for resume block classification.

1) Hierarchical Transformer Encoder: Different from the previous document understanding model [17, 18, 19] pre-trained on the single page document like invoice and form document, the resume document is usually a long document with several pages. In order to handle the long document, we adopt a hierarchical transformer encoder architecture as the encoder backbone, which consists of a sentence-level encoder and a document-level encoder. The former encodes a sequence of tokens to produce final representations of sentence and the latter learns the semantic interactions among these sentence representations.

Sentence-level Transformer Encoder. The BERT model [25], based on a multi-layer bidirectional Transformer encoder, has been verified its powerful abilities of semantic representation and knowledge transfer by self-supervised pre-training on large-scale training data. In order to capture the token-level semantic features, we deployed a 6-layers bidirectional Transformer encoder, which accepts a sequence of tokens and stacks six layers to produce final representations. Specifically, given a sentence s_j in $\mathcal{D}$, the input embeddings

[1]These eight tags donate personal information, education experience, work experience, project experience, summary, awards, skill description and section title, respectively.

[2]https://github.com/pymupdf/PyMuPDF

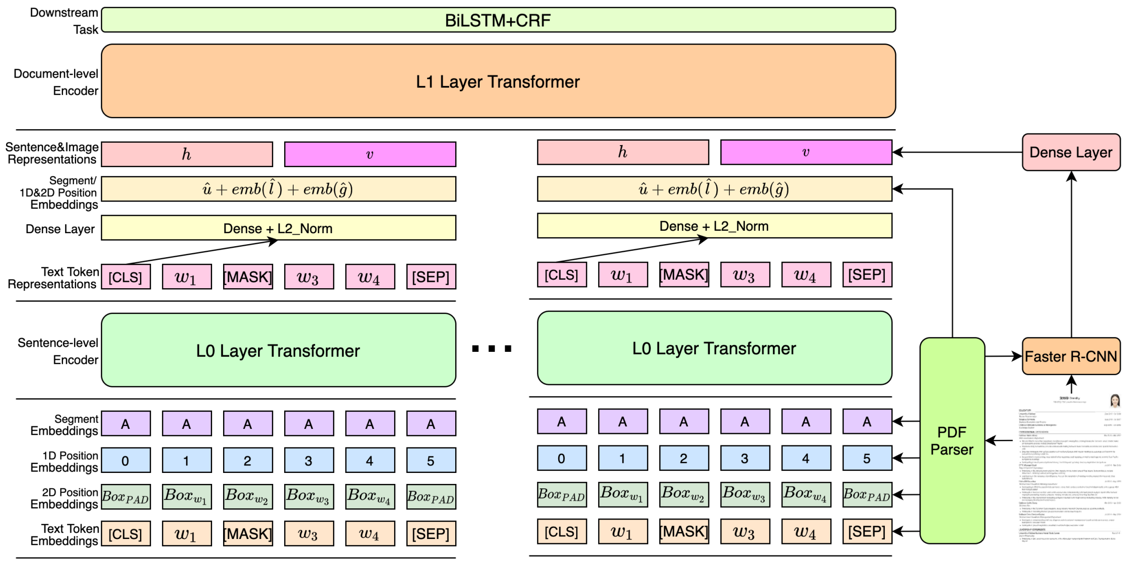


Fig. 2. The framework overview of hierarchical multi-modal pre-training model.

are computed by summing two parts: the text embedding and the layout embedding.

For the text embedding, we follow the common practice in pre-training BERT model. In detail, we first use WordPiece [36] to tokenize the text in the s_j into a sequence of words $\{w_j^0, w_j^1, ..., w_j^{|s_j|}\}$. Then, each word will be assigned a certain segment symbol $g_j^i \in \{[A], [B]\}$, which is used to distinguish different text segments. In addition, the word index will also be used to encode the position information. Finally, the final text embedding o_j^i is the sum of word embedding $emb(w_j^i)$, 1-D positional embedding $emb(l_j^i)$ and segment embedding $emb(g_j^i)$. Formally, the i-th word in s_j can be formulated as below:

$$o_j^i = emb(w_j^i) + emb(l_j^i) + emb(g_j^i). \quad (1)$$

For the layout embedding, in the process of PDF parsing, we obtain the spatial layout information with the format of (x_0, y_0, x_1, y_1, p) for the word w_i^j, where (x_0, y_0) and (x_1, y_1) are the top-left corner coordinate and bottom-right coordinate for the word, respectively, and p is the page index. We followed the work of LayoutLMV2 [18], all coordinates are normalized and discretized to integers in the range [0, 1000]. For i-th word, we first convert its layout information into a seven tuple, i.e., (x_{min}, y_{min}, x_{max}, y_{max}, $width$, $height$ p), where $height$ and $width$ are the height and width of the word. Then the spatial information will be pass through three embedding layers to embed x-axis features, y-axis features and page features, respectively. Thus, the final layout embedding u_j^i for w_j^i can be formulated as

$$\begin{aligned} u_i^j =& [emb_g(p); emb_x(x_{min}, x_{max}, width); \\ & emb_y(y_{min}, y_{max}, height)], \end{aligned} \quad (2)$$

whre "[;]" is the vector concatenation operation.

Finally, these input embeddings, i.e., $o_j + u_j$, are passed through our sentence-level Transformer encoder that can obtain a sequence of contextualized representations for words in the sentence. Following the setting in BERT model, we use [CLS] token as the learned representation of the whole sentence. As shown in Figure 1, we also add another dense layer and perform a L2 normalization on the sentence representation. Finally, we use h_j denote the representation of the j-th sentence s_j.

Document-level transformer encoder. Similar to the sentence-level transformer encoder architecture, the document-level encoder adopts a 4-layers bidirectional Transformer encoder, which accepts a sequence of sentence representations. Actually, the j-input sentence representations contain three parts: the sentence representation h_j derived from the sentence-level encoder, the sentence-level layout information $\hat{u}_j$ and the sentence-level visual information v_j. It is noted that we do not introduce the visual information in the sentence-level encoder because we assume it is not easy to distinguish the word features from the corresponding word image among different words. However, the sentence-level visual information can better reflect an effective feature. For instance, a section title in the resume document usually has different font color or a larger font size.

To utilize the visual feature of a sentence in the document and align it with the corresponding text, for each sentence s_j with the 2-D positional information (x_0', y_0', x_1', y_1'), we still use PyMuPDF to extract document images from the PDF document and then segment the document image into several pieces according to the position information, which correspond to the sentence s_j. Thus, we can directly leverage the pre-trained Faster R-CNN [37] to produce the image region features for these image pieces. After obtaining the visual features v_j for the sentence s_j, we concatenate v_j with the contextualized representations h_j as the two-modal sentence embeddings, i.e., $h_j^\star$ = [h_j; v_j]. In addition, we adopt the same approaches in the sentence-level encoder to obtain 2-

D spatial layout embeddings $\hat{u}_j$, 1-D positional embeddings $emb(\hat{l}_j)$ and segment embeddings $emb(\hat{g}_j)$. These embedding will be summed together as the final input embeddings, which will feed to the document-level encoder to capture the contextualized representations h'_j with an adaptive attention mechanism.

2) Pre-training Objectives: For the model training, we adopt the "pre-training + fine-tuning" paradigm as in BERT. For the pre-training, we elaborately design three self-supervised objectives for our hierarchical Transformer encoder, which can learn effective semantic representation from large-scale unlabeled corpus and reduce the dependence on labeled data in downstream supervised tasks.

Objective #1: Masked Layout-Language Model. Followed by LayoutLM [17], our sentence-level transformer encoder learns the language representation with the clues of 2-D layout embeddings and text embeddings. Due to its textual richness for resume documents, we initialize the parameters of the sentence-level encoder with a pre-trained RoBERTa model with 6 layers so that the external knowledge can transfer to our encoder. Then, we continue to optimize the parameters on the large-scale resume document training data. To be specific, we randomly mask some of the input tokens in a sentence but retain the corresponding 2-D position embeddings, and then train the model to predict the masked tokens given the contexts. Thus, our sentence-level encoder not only understands the language contexts of resume but also utilizes the corresponding 2-D position information. In addition to adding 2-D positional information, masked layout-language model is same with the masking word prediction in the BERT [25]. For convenience, we denote the loss of the masked layout-language model task as $\mathcal{L}_{wp}$. The detailed computation process please refers to the masking word prediction in BERT.

Objective #2: Self-supervised Contrastive Learning. Inspired by the contrastive learning framework [38, 39], we use self-supervised contrastive loss as our objective function to pre-train our document-level transformer encoder. Specifically, given a sequence of sentence representations H_s = $\{h_1^\star$, $h_2^\star$, ..., $h_m^\star\}$ from a resume document, we randomly replace k ($k < m$) sentence representations with a randomly initialized masked vector $\hat{h} \in \mathcal{R}^d$ in current training batch. For instance, the masked document become $\hat{H}_s$ = $\{h_1^\star$, $\hat{h}_2$, $h_3^\star$, $\hat{h}_4$, ..., $h_m^\star\}$ if we randomly mask the 2nd and 4th sentence representation. In each pre-training step, we repeat the dynamic sampling procedure in a batch. Thus, for the same document, the masked sentence may be occur at the different positions in different training steps. Thereby, compared with the static masking, the dynamic masking strategy can obtain more diverse masked sentences in a document to train the model.

After the dynamic sentence masking, the masked sequence $\hat{H}_s$ will be passed through the document-level transformer encoder that can capture contextual block representations H_d = $\{h'_1$, h'_2, ..., $h'_m\}$. Then, we perform the contrastive learning for the masked sentence prediction. Specifically, given a batch of N masked sentences with the generated contextualized sentence representation $h'_d \in \mathcal{R}^{N\times D}$ and the ground truth sentence representation $h_s^\star \in \mathcal{R}^{N\times D}$ where $N = b \times k$. According to the contrastive learning framework, we compute a similarity score between h'_d and $h_s^\star$, which ensures that the sentence representations in the same positions are closer to each other in the latent embedding space. Formally, we can compute a pairwise similarity matrix for every masked sentence pair in the current batch as follow:

$$Sim(h'_d, h_s^\star) = h'_d h_s^{\star T}. \tag{3}$$

In the above equation, Sim($h'_{d,i}$, $h^\star_{s,j}$) is the predicted similarity between the i-th contextual sentence representation and the j-th original sentence representation. We normalize it with a softmax function, and then the self-supervised contrastive loss function is described as follows:

$$\mathcal{L}_{cl} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{1}\{i=j\}log\frac{exp(Sim(h'_{d,i}, h^\star_{s,j})/\tau)}{\sum_{k=1}^{N}exp(Sim(h'_{d,i}, h^\star_{s,k})/\tau)} \tag{4}$$

where $\mathbf{1}\{i = j\}$ evaluate to 1 iff i=j, and τ denotes a temperature parameter.

Objective #3: Dynamic Next-sentence Prediction. Inspired by the next sentence prediction (NSP for short) in BERT [25], we propose the dynamic next sentence prediction (DNSP for short) to learn the dependency relation between two adjacent sentences. Different from the NSP only considering textual semantic relations, DNBP models the dependency relation from the perspectives of visual semantics, text semantics and spatial positions. In more detail, we randomly sample L ($L < m$) sentences $H' \in \mathcal{R}^{L\times D}$ and the corresponding next sentences $H'' \in \mathcal{R}^{L\times D}$ from the contextual sentences representations H_d in each training batch. For instance, the sampled sentence sequence is H' = $\{h'_2$, h'_5, $h'_8\}$ and the adjacent sentence sequence becomes H'' = $\{h'_3$, h'_6, $h'_9\}$ if we sample the 2nd, 5th and 8th sentences from H_d. Then, we use the adjacent sentences to build the ground truth training pairs $\{(h'_2$, $h'_3)$, $(h'_5$, $h'_6)$, $(h'_8$, $h'_9)\}$ for the next-sentence prediction. In addition, we use a parameter matrix $W_d \in \mathcal{R}^{D\times D}$ to model the dependency relation between two adjacent sentences, which can be formulated as follows:

$$M_d(H', H'') = H'W_dH''^T, \tag{5}$$

Similar to the equation 3, $W_d(H'_i, H''_j)$ denotes the interaction between the i-th sentence and j-th sentence, and we also utilize a softmax function to normalize it. Finally, the final loss function is described as below:

$$\mathcal{L}_{ns} = -\frac{1}{L}\sum_{i}^{L}\sum_{j}^{L}\mathbf{1}\{i=j\}log\frac{exp(M_d(H'_i, H''_j))}{\sum_{k=1}^{L}exp(M_d(H'_i, H''_k))} \tag{6}$$

Noted that we also adopt dynamic sampling strategies for next sentence prediction for every document in each training batch. Finally, the overall objective for the pre-training process can be formulated as below:

$$\mathcal{L}_{pretrain} = \lambda_1\mathcal{L}_{wp} + \lambda_2\mathcal{L}_{cl} + \lambda_3\mathcal{L}_{ns} \tag{7}$$

where λ_1, λ_2 and λ_3 are the hyper-parameters, which are used to balance the three pre-training objectives.

3) Fine-tuning Objectives: After the hierarchical encoder is pre-trained based on the three pre-training objectives above, we begin to fine-tune the model on a small amount of training data to adapt the pre-trained model to a new task, i.e., resume block classification. Specifically, we regard it as a sequence labeling task and apply a BiLSTM network stacked on the top of hierarchical transformer encoder. More formally, given a sequence of contextual sentence representation $H_d = \{h'_1, ..., h'_i, ..., h'_m\}$ derived from the document-level encoder, the BiLSTM network operates on h'_i at the i-th step to output h^i_b :

$$\begin{aligned} \overrightarrow{h^i_b} &= \overrightarrow{\text{LSTM}}(h'_i, \overrightarrow{h^{i-1}_b}), \\ \overleftarrow{h^i_b} &= \overleftarrow{\text{LSTM}}(h'_i, \overleftarrow{h^{i+1}_b}), \\ h^i_b &= [\overrightarrow{h^i_s}; \overleftarrow{h^i_s}]. \end{aligned} \tag{8}$$

where $\overrightarrow{\text{LSTM}}$ and $\overleftarrow{\text{LSTM}}$ denote the forward LSTM [40] and the backward LSTM, and "[;]" is the concatenation operation.

Finally, the output of the BiLSTM layer h^i_b is used to predict a score for each possible tag with a multi-layer perceptron (MLP). Accordingly, we add a conditional random field (CRF) layer to compute the sentence CRF loss [41] using the forward-backward algorithm at training time, and using the Viterbi algorithm [42] to find the most likely tag sequence at test time.

4) Knowledge Distillation: Labeling data is expensive for resume block classification task, but unlabeled data is more likely abundant. In addition to the above pre-training methods, we also leverage the knowledge distillation technique [43] to augment training data. Specifically, assuming that we have resume document data $\mathcal{T}$ for the the resume block classification task, which can be divided into the training data $\mathcal{T}_l$ annotated by human experts and the unlabeled data $\mathcal{T}_u$. We first train a teacher model LayoutXLM [19] with $\mathcal{T}_l$, which is a multi-modal pre-trained model for multilingual document understanding and can fulfill the token-level classification for resume blocks. Then, we use the trained LayoutXLM to auto-annotate a large amount of unlabeled data $\mathcal{T}_u$ with (hard) pseudo labels [3]. The augmented data $\mathcal{T}_p$ with the pseudo label can be used train our model. The detailed training process is described at Algorithm 1. Actually, merging the unlabled data with pseudo-labels and the training data with real labels can improve the performance of model [44]. The potential reason is that LayoutXLM is quite different from our hierarchical pre-training model and the semantic knowledge in LayoutXLM can be transferred and leveraged to improve the quality of prediction of our model.

B. Intra-block Information Extraction

Intra-block information extraction is, essentially, a NER problem. After the document is segmented into multiple blocks, we begin to extract the fine-grained entities from each block. Although plenty of state-of-the-art NER approaches [45, 46, 47] have achieved a promising performance, they are fully supervised learning methods, which strongly depend on a large-scale fine-grained annotated data. Due to the lack of labeling data, we resort to a distantly-supervised NER method. Specifically, we first build domain dictionary for each entity class. Then, the training data is annotated by the self-built dictionary automatically. Finally, we adopt a classical "BERT+BiLSTM+MLP" architecture with self-training framework for the entity extraction.

[3]The original labels are predicted for tokens in the document. We need to convert them to the labels of sentence.

Algorithm 1 Training Process of our model.

```
1: Pretrain the hierarchical transformer encoder by Eq. (7);
2: Train the teacher model LayoutXLM using $\mathcal{T}_l$;
3: Use the trained LayoutXLM to auto-generate the pseudo
   label data $\mathcal{T}_p$ based on unlabeled data $\mathcal{T}_u$;
4: Train our model on $\mathcal{T}_p$;
5: Fine-tune our model on $\mathcal{T}_l$;
```

1) Entity Dictionary Construction: As shown in Table IV, the resumes contain rich entity information in each segmented block. In order to generate the training instances, we build multiple dictionaries to automatically annotate data. In more detail, in the personal information, the person name entities are collected from the name database collected from the online websites. In the education experience, we collect college, and major entities from web encyclopedia data. In the work experience, the company entities and job position entities are collected from encyclopedia data and online recruitment websites. In the project experience, the project name entities are extracted from the project information in the candidate database. In addition, some entities like gender and degree have the limited value type, and other entities like phone number, email and date can be recognized by the regular expression.

2) Automatic Data Annotation: After the entity dictionaries are built, we automatically generate training data by matching unlabeled sentences with the dictionaries. The matching can be achieved by string matching [9], regular expressions [8] or heuristic rules. Specifically, we match the entity mentions, such as college name, major and company name, with exactly the same surface names in the dictionaries. The entities, such as email, date and phone number, are directly matched with regular expressions. In addition to the string matching and regular expression matching, some heuristic rules are also designed for labelling training data. For instance, the person name starts with a common family name and usually occurs at the beginning of the document with a bigger font size. Besides, some prefix words or suffix words of entities can indicates their types [4].

To enhance the diversity of training data, we replace the entity mentions in the sentence with other entities in the dictionaries. Besides, the order of entities, such as the company name and the work date in the work experience, and the degree and the major in the education experience can be adjusted for data augmentation.

[4]Some entities usually begin with prefix words or end with suffix words. For instance, the age entity usually has the prefix: "Age: ", the email entity has the prefix: "Email: " and the company entity often ends with "Co. LTD".

3) Distantly-Supervised NER Model: We adopt the typical "BERT+BiLSTM+MLP" architecture for distantly supervised NER. Different from resume block classification, it performs on the word-level and accepts a text sequence as input. Specifically, given a sequence of input words from a segmented block, a pre-trained BERT model first captures the contextualized semantic representation for each word in the input. Then, a BiLSTM layer is stacked on the top of BERT and projects the output features into the label space. Finally, we use a MLP layer to predict the labels of the input sequence.

4) Self-Distillation based Self-Training Learning: The challenge for distant supervision is the training data with noisy label or incomplete label. To reduce the impact of this problem, we proposed a self-training learning approach based on self-distillation framework to improve the performance of model [48]. In more detail, we first train a teacher model $f(X_t;\theta_{tea})$, i.e., BERT+BiLSTM+MLP, using the distantly supervised training set, where θ_{tea} is the model learning parameters and $X_t = \{X_1, ..., X_i, ..., X_M\}$ is the training data with M sentences. Since the teacher model is trained on a noise training set, we optimize its parameters by using Adam algorithm [40] with a early stopping trick, which can prevent the overfitting and improve the generalization of the model. Then, we deploy a student model $f(X_t;\theta_{stu})$ parameterized identically to the teacher model, i.e., $\theta_{stu} = \theta_{tea}$. During each training iteration t, the teacher model generates a batch of samples with (hard) pseudo labels, which will be used to train the student model. At any iteration step, we re-initialize the teacher model with the current parameters of the student model if its performance improves on the validation dataset X_v and then continue train the student with the same procedure. The behind intuition is both the teacher and the student benefit each other that will produce a virtuous cycle. In other words, a better teacher will produce more accurate pseudo-labels, which in turn helps train a better student. Meanwhile, the improved student will be used to re-initialize the teacher. This teacher-student framework enjoys the merit that it progressively improves the model confidence over data.We formally describes our training approach in Algorithm 2.

5) High-Confidence Soft Labels: The training data with the hard pseudo labels generated by the teacher usually exists two problems: 1) The hard label only retains the most confident class for each token but misses the confidence of other classes. 2) The training sample with low confidence will bring greater possible uncertainty for the student model training. Accordingly, we propose to utilize soft labels with re-weighting confidence and propose to filter tokes based on the prediction confidence, respectively.

Specifically, given the j-th word in the i-th sentence X_i at the t-th iteration step. We follow the work of [49]. The soft-pseudo-labels S_i^t over C classes generated by the teacher are calculated as follows:

$$S_{i,j}^t = S_{i,j,c}^t\Big|_{c=1}^{C} = \frac{f_{j,c}^2(X_i;\theta_{tea}^t)/p_c}{\sum_{k=1}^{C} f_{j,k}^2(X_i;\theta_{tea}^t)/p_k}\Bigg|_{c=1}^{C}, \quad (9)$$

where $f_{j,c}(X;\theta_{tea})$ denotes the corresponding output probability score for the c-th class, and $p_c = \sum_{i=1}^{M}\sum_{j=1}^{N} f_{j,c}(X_i;\theta_{tea}^t)$ computes the unnormalized frequency of the tokens corresponding to the c-th class. As can be seen, the above equation adopts a squared re-weighting method that prefer to the classes with higher confidence essentially. Thus, the parameters of the student model is then optimized by minimizing the KL-divergence loss:

Algorithm 2 Self-training learning based on Self-distillation.

Input: M training sentences X_t, the validation set X_v;
Output: θ_{stu}^T;

```
1:  Train a teacher model $f(X_t;\theta_{tea})$ with early stopping;
2:  Initialize a student model $f(\cdot;\theta_{tea})$, i.e., $\theta_{stu} = \theta_{tea}$;
3:  for $t$ = 1, 2, 3, ..., $T$ do
4:      $\theta_{tea}^{(t-1)} = \theta_{tea}$, $\theta_{stu}^{(t-1)} = \theta_{stu}$;
5:      Sample a minibatch $\mathcal{B}_t$ from $X_t$;
6:      Use $f(\mathcal{B}_t;\theta_{tea}^{(t-1)})$ to generate pseudo-labels $\hat{Y}_i\big|_{i\in\mathcal{B}_t}$;
7:      Use Adam algorithm to update the student model, i.e.,
        $\theta_{stu}^{(t)}$ = Adam($\theta_{stu}^{(t-1)}$, $\{(X_i, \hat{Y}_i)\}_i \in \mathcal{B}_t$);
8:      if The performance of $f(X_v;\theta_{stu}^t)$ is improved then
9:          Re-initialize the teacher using the student, i.e., $\theta_{tea}^t$
            = $\theta_{stu}^t$
10:     end if
11: end for
```

$$\theta_{stu}^t = \underset{\theta_{stu}}{argmin}\,\frac{1}{M|X_i|}\sum_{i=1}^{M}\sum_{j=1}^{|X_i|}\sum_{c=1}^{C} -S_{i,j,c}^t log f_{j,c}(X_i;\theta_{stu}). \quad (10)$$

In addition, we further reduce the uncertainty in the training data. To be specific, at the t-th iteration step, we choose a set of high confidence tokens from the i-th sentence by

$$H_i^t = \{j : \underset{c}{max}\, S_{i,j,c}^t > \gamma\}, \quad (11)$$

where $\gamma \in (0,\ 1)$ is a threshold parameter. Thus, we can optimize the parameters of the student model only using the selected training tokens

$$\theta_{stu}^t = \underset{\theta_{stu}}{argmin}\,\frac{1}{M|H_i^t|}\sum_{i=1}^{M}\sum_{j=1}^{|H_i^t|}\sum_{c=1}^{C} -S_{i,j,c}^t log f_{j,c}(X_i;\theta_{stu}). \quad (12)$$

By means of such high confidence token selection, the student model is actually enforced to better fit training tokens with high confidence. Thus, the model robustness against low-confidence tokens can be enhanced.

V. EXPERIMENTS

In this section, we will estimate the performance of ResuFormer based on the extensive experiments, which conducted on the real-world datasets.

A. Evaluation on Resume Block Classification

1) Datasets: Our experiments were conducted on a large-scale resume documents provided by a high-tech company. In order to fulfill resume block classification, we used 80,000 resume documents for model pre-training. In addition, we

TABLE I
SOME IMPORTANT STATISTICS OF RESUME DOCUMENT DATASETS.

Datasets	Pre-training Documents	Finetuning Documents		
		train	validation	test
# of samples	80,000	1,100	500	500
avg # of tokens	1,704.20	1,721.98	1,704.37	1,685.43
avg # of sentences	90.28	90.71	89.57	91.26
avg # of pages	2.1	2.02	2.04	2.23

randomly selected 2,100 resume documents and invited ten experts to annotate them with an open-source PDF annotation tool, i.e., PAWLS [5]. Specifically, each resume document will be divided into several blocks, and each block will be assigned to one label in eight categories [6]. Among the 2,100 annotated documents, 1,100 documents were selected as the training set, 500 documents as the testing set and the rest of them as the validation set. Moreover, we also reported some key statistical information including the average tokens, the average sentences, and the average pages in Table I.

2) Implementation Details: The sentence-level encoder use a 6-layer Transformer encoder with 12 heads and the hidden size is set as 768. Its parameters are initialized with the weight of a 6-layer pre-trained RoBERTa [7]. For the document-level encoder, we use a 4-layer Transformer encoder. The heads and the hidden size are same with the sentence encoder. We apply the Adam optimizer with the weight decay of 0.01 to optimize parameters. During the pre-training stage, the learning rate is set as 5e-5. During the fine-tuning stage, the learning rate for the hierarchical encoder and the BiLSTM with CRF layer are set as 5e-5 and 1e-3, respectively. In the pre-training objectives, the number of masked sentence and next sentence account for 0.2 in all sentences. To limit the size of the model, the maximum number of tokens in a sentence is 55, and the maximum number of sentence in a document is limited to 350. The value for τ is fixed to 0.8. The balance parameters λ_1, λ_2 and λ_3 for three pre-training objectives are set as 0.4, 1.0 and 0.6, respectively.

3) Baselines: To achieve the comprehensive and comparative analysis of our proposed methods, we chose several SOTA methods as baselines:

BERT+CRF [50]: It is a typical information extraction model for text sequence, which is based on the pretraining model BERT with conditional random field (CRF) layer and achieves the resume block classification.

HiBERT+CRF [51]: It is a hierarchical BERT model with CRF layer for sentence by sentence classification, in which the sentence-level BERT initializes its parameters with a pre-trained BERT and the document-level BERT aims to learn the contextual semantic representation.

RoBERTa+GCN [26]: It combines the power of large pre-trained language models with graph neural networks to efficiently encode both textual and positional information for document information extraction.

LayoutXLM [19]: It is a multi-modal pre-trained model for multilingual document understanding, which jointly model the text information, the layout information and the visual information in the large-scale multilingual documents to obtain more effective semantic representation. Due to its pretraining on multilingual documents, it can be easily adapted to resume document and achieve resume block classification with token by token.

In addition, we also compared several variants of our model to examine the relative influences of different modules on performance:

Our Method(w/o KD) is a variant of our method without using knowledge distillation.

Our Method(w/o WMP) is a variant of our method without using word-masking prediction in the pre-training of the masked layout-language model.

Our Method(w/o SCL) is a variant of our method without using dynamic sentence-masking prediction in self-supervised contrastive learning.

Our Method(w/o DNSP) is a variant of our model without using dynamic next sentence prediction mechanism.

4) Evaluation Metrics: Since our model accepts serialized 2-D resume documents as the inputs, we follow the evaluation metrics of document layout analysis task [52] instead of the typical IOB-tagging evaluation. Specifically, we compute *Precision*, *Recall* and *F1* score as metrics for each kind of resume semantic structure, which are defined as follows:

$$P = \frac{\text{Area of Ground truth tokens in Detected tokens}}{\text{Area of all Detected tokens}}, \quad (13)$$

$$R = \frac{\text{Area of Ground truth tokens in Detected tokens}}{\text{Area of all Ground truth tokens}}, \quad (14)$$

$$F1 = \frac{2 * P * R}{P + R}, \quad (15)$$

where P and R denote the precision and recall, respectively.

5) Results: Table II shows the overall performance of our proposed method as well as the baseline methods. Among these baselines, "BERT+CRF" and "HiBERT+CRF" are textual information based non-pretrained models, and the other two baselines are multi-modal pretrained models. From the Table II, we can clearly observe that our model performs the best F1 score except "PInfo" tag. Compared with "LayoutXLM", our model achieves obvious improvements on 6 tags. It improves 7.39%, 6.98%, 6.59%, 4.05%, 4.04% and 3.03% on *WorkExp*, *ProjExp*, *Summary*, *Awards*, *SkillDes*, and *Title* tags, respectively. In addition, we can find that the pre-trained models achieve better performance than non-pretrained models that validates the necessity and importance of self-supervised training. Moreover, we also compare the average running efficiency on one resume. Obviously, "HiBERT+CRF" and our method run more quickly. The main reason is that the two methods work on the sentence-level, while the other three baselines run on the token-level. Compared with the

[5]https://github.com/allenai/pawls

[6]The label category includes *Title*, *PInfo*, *EduExp*, *WorkExp*, *ProjExp*, *SkillDes.*, *Summary* and *Awards*

[7]https://github.com/brightmart/roberta_zh.

TABLE II
THE PERFORMANCE OF RESUME BLOCK CLASSIFICATION: F1 SCORE (RECALL/PRECISION)(IN %)

		BERT+CRF	**HiBERT+CRF**	**RoBERTa+GCN**	**LayoutXLM**	**Our Method**
Tags	PInfo	77.88 (78.03 / 77.74)	73.28 (68.60 / 78.64)	89.95 (92.38 / 87.65))	**92.99** (95.53 / 90.58)	91.75 (95.91 / 87.93)
	EduExp	63.95 (61.47 / 66.64)	60.50 (54.89 / 67.38)	88.68 (91.16 / 86.34)	90.85 (93.06 / 88.73)	**91.00** (91.42 / 90.58)
	WorkExp	60.77 (53.10 / 71.03)	56.25 (53.81 / 58.92)	84.72 (82.11 / 87.50)	86.20 (83.28 / 89.32)	**93.59** (92.52 / 94.68)
	ProjExp	66.51 (81.18 / 56.33)	59.88 (69.16 / 52.79)	85.68 (87.72 / 83.74)	86.25 (88.58 / 84.03)	**93.23** (95.83 / 90.76)
	Summary	43.42 (51.92 / 37.32)	36.60 (28.32 / 51.73)	83.95 (83.72 / 84.19)	85.10 (84.63 / 85.57)	**91.69** (90.28 / 93.14)
	Awards	15.31 (39.18 / 9.51)	10.48 (9.46 / 11.74)	70.12 (77.51 / 64.02)	71.23 (77.22 / 66.09)	**75.28** (74.32 / 76.26)
	SkillDes	40.94(33.07 / 53.72)	35.96 (35.05 / 36.91)	87.01 (86.80 / 87.23)	88.64 (88.71 / 88.58)	**92.68** (92.63 / 92.74)
	Title	43.10 (28.77 / 85.86)	37.25 (26.24 / 64.14)	84.88 (75.86 / 96.33)	84.77 (75.03 / 97.41)	**87.80** (80.93 / 95.94)
Time / Resume		3.26s	**0.19s**	3.46s	3.88s	0.27s

TABLE III
THE RESULT OF ABLATION TEST: F1 SCORE (RECALL/PRECISION)(IN %)

		Our Method	**w/o KD**	**w/o WMP**	**w/o SCL**	**-w/o DNSP**
Tags	PInfo	**91.75** (95.91 / 87.93)	89.91 (94.32 / 85.89)	87.39 (92.73 / 82.64)	78.85 (84.62 / 73.81)	83.30 (89.03 / 78.26)
	EduExp	**91.00** (91.42 / 90.58)	89.35 (89.84 / 88.86)	87.22 (88.23 / 86.24)	79.79 (80.42 / 79.16)	83.93 (84.67 / 83.21)
	WorkExp	**93.59** (92.52 / 94.68)	88.94 (84.21 / 94.23)	86.20 (82.66 / 90.05)	79.81 (76.65 / 83.25)	83.66 (80.62 / 86.94)
	ProjExp	**93.23** (95.83 / 90.76)	88.79 (95.65 / 82.84)	86.05 (92.74 / 80.26)	77.13 (84.42 / 71.01)	82.17 (88.93 / 76.36)
	Summary	**91.69** (90.28 / 93.14)	90.06 (93.21 / 87.12)	88.03 (90.25 / 85.92)	79.01 (81.86 / 76.35)	84.26 (86.93 / 81.74)
	Awards	**75.28** (74.32 / 76.26)	71.91 (79.84 / 65.42)	69.57 (78.22 / 62.64)	60.73 (70.83 / 53.15)	66.03 (75.78 / 58.51)
	SkillDes	**92.68** (92.63 / 92.74)	89.84 (86.35 / 93.62)	88.46 (85.74 / 91.35)	79.34 (76.84 / 82.01)	84.42 (81.48 / 87.57)
	Title	**87.80** (80.93 / 95.94)	85.37 (78.46 / 93.62)	83.85 (77.72 / 91.04)	75.90 (69.75 / 83.25)	80.33 (74.21 / 87.56)

TABLE IV
MAIN RESULTS OF INTRA-BLOCK INFORMATION EXTRACTION: F1 SCORE (RECALL/PRECISION)(IN %)

Blocks	**Tags**	**D&R Match**	**BERT+BiLSTM+CRF**	**BERT+BiLSTM+FCRF**	**AutoNER**	**Our Method**
PInfo	Name	69.59 (56.52 / 90.54)	85.10 (85.27 / 84.93)	93.03 (93.82 / 92.26)	94.38 (95.45 / 93.33)	**97.52** (98.38 / 96.67)
	Gender	92.76 (87.32 / 98.93)	93.00 (94.32 / 91.72)	95.41 (96.67 / 94.18)	96.17 (97.13 / 95.22)	**98.66** (99.41 / 97.92)
	PhoneNum	86.74 (80.25 / 94.37)	91.83 (92.87 / 90.82)	93.88 (95.26 / 92.53)	95.86 (97.32 / 94.45)	**98.51** (99.31 / 97.72)
	Email	87.98 (80.26 / 97.34)	90.95 (91.73 / 90.19)	93.35 (94.28 / 92.43)	95.46 (96.21 / 94.73)	**98.31** (99.07 / 97.56)
	Age	82.06 (73.63 / 92.68)	84.85 (85.53 / 84.18)	87.54 (88.89 / 86.24)	89.48 (90.73 / 88.26)	**92.98** (93.62 / 92.35)
EduExp	College	66.35 (50.45 / 96.88)	71.57 (70.35 / 72.83)	78.10 (77.68 / 78.52)	80.04 (79.93 / 80.16)	**85.59** (85.31 / 85.87)
	Major	66.37 (51.37 / 93.76)	70.97 (71.23 / 70.72)	76.44 (76.37 / 76.52)	78.53 (78.24 / 78.83)	**83.75** (83.77 / 83.72)
	Degree	83.30 (72.69 / 97.53)	88.08 (83.52 / 93.16)	90.23 (85.48 / 95.55)	91.14 (86.24 / 96.62)	**93.55**(89.48 / 98.00)
	Date	82.95 (73.72 / 94.83)	86.73 (85.36 / 88.15)	88.43 (87.21 / 89.69)	90.31 (88.94 / 91.73)	**92.82** (91.07 / 94.63)
WorkExp	Company	60.22 (44.67 / 92.36)	69.35 (68.43 / 70.29)	76.80 (75.94 / 77.67)	77.92 (76.05 / 79.89)	**82.74** (82.13 / 83.36)
	Position	55.42 (39.68 / 91.83)	65.80 (63.37 / 68.42)	74.88 (74.74 / 75.02)	77.13 (76.47 / 77.81)	**83.45** (81.27 / 85.81)
	Date	83.62 (74.28 / 95.65)	86.78 (85.34 / 88.26)	88.74 (87.16 / 90.37)	90.55 (88.92 / 92.25)	**92.76** (90.78 / 94.82)
ProjExp	ProjName	43.23 (28.47 / 89.79)	63.24 (61.36 / 65.24)	73.37 (72.68 / 74.07)	75.53 (75.52 / 75.54)	**80.19** (78.34 / 82.12)
	Date	83.90 (75.22 / 94.84)	86.41 (84.62 / 88.27)	88.20 (86.37 / 90.11)	89.57 (87.74 / 91.48)	**91.78** (89.85 / 93.79)

best baseline, i.e., ”LayoutXLM”, our model improves about 15 times for the running efficiency that meets the latency performance of online model deployment.

6) Ablation Test: We also conduct a ablation test to validate the effectiveness of different modules. The experimental results are shown in Table III. We can note that the performance of our model decreases more or less if we remove each kind of module. Particularly, the performance decreases the most and the second when we respectively remove self-supervised contrastive learning (SCL) and dynamical next sentence prediction (DNSP) that validate the importance and effectiveness in our self-supervised training objectives for the document-level encoder. In addition, we can find that the knowledge distillation (KD) trick increases the performance of our model due to the enhancement of training data.

B. Evaluation on Intra-block Information Extraction

1) Datasets: Based on the above resume document dataset used in the task of resume block classification, we first use the PDF parsing tool to extract the pure text from these PDF documents. Then, the extracted text are used to build distantly-supervised training data with the method in Section IV-B2. It is worth noting that the average number of tokens of a resume document are more than 1,600. Therefore, we utilize the resume block classification approach to segment a training document into several blocks and the text in each block will be a training instance. We select 20,000 training samples as the final training dataset, in which each training instance has

TABLE V
ABLATION TEST FOR INTRA-BLOCK INFORMATION EXTRACTION: F1 SCORE(IN %)

Blocks	Tags	Our Method	w/o HCS	w/o SL	w/o SD
PInfo	Name	**97.52**	95.87	94.56	85.10
	Gender	**98.66**	97.54	96.23	93.00
	PhoneNum	**98.51**	97.25	96.11	91.83
	Email	**98.31**	97.12	96.08	90.95
	Age	**92.98**	91.77	90.42	84.85
EduExp	College	**85.89**	83.68	81.28	71.57
	Major	**83.75**	81.83	80.14	70.97
	Degree	**93.55**	92.74	91.47	88.08
	Date	**92.82**	91.53	90.46	86.73
WorkExp	Company	**82.74**	80.53	78.36	69.35
	Position	**83.45**	81.57	79.62	65.80
	Date	**92.76**	91.32	90.25	86.78
ProjExp	ProjName	**80.19**	78.67	76.62	63.24
	Date	**91.78**	90.35	89.87	86.41

TABLE VI
SOME IMPORTANT STATISTICS OF INTRA-BLOCK INFORMATION EXTRACTION DATASETS.

Datasets	# of samples	avg # of tokens	avg # of entities
Train Set	20,000	362	3.5
Validation Set	400	359	4.1
Test Set	600	381	4.3

at least one matched entity mention. In addition, we also invite 10 experts to annotate 1,000 samples, in which 400 samples will randomly selected as the validation dataset and the other 600 samples are the test dataset. Some important statistics are summarized in Table VI.

2) Implementation Details: For the encoder backbone BERT, we used a pre-trained model, RoBERTa-zh-base [8], to initialize its parameters, which has 12 layers with 768 hidden dimensions and 12 attention heads. For BiLSTM, we set the hidden units to 256 and theirs hidden states are initialized as zeros. The learning rate for BERT was set to 1e-5, but for the BiLSTM layer and MLP layer, the learning rate was 1e-3. In addition, we set the mini-batch size to 128 in the model training. The threshold for the high-confidence token selection is set to 0.8.

3) Baselines: We compare our model with several baseline methods.

D&R Match: It performs string matching with the self-built entity dictionaries and regular matching to obtain entity mentions from the input text.

BERT+BiLSTM+CRF [50]: It deploys a typical bidirectional LSTM networks with a CRF layer on the top of a pre-trained BERT model, which is trained on the distant labels automatically annotated by the methods in Section IV-B2.

BERT+BiLSTM+FCRF: Compared with the above "BERT+BiLSTM+CRF", we replace CRF layer with a fuzzy CRF layer [53], which are more suitable for distantly-supervised training data.

AutoNER [53]: Different from typical IOB tags, it adopts a novel "Tie or Break" tagging scheme, and assigns ambiguous tokens with all possible labels.

[8]https://github.com/brightmart/roberta_zh

Noted that we deployed our method based on the structure of "BERT+BiLSTM+MLP" with self-distillation framework, in which the parameters are trained with the soft labels by high-confidence token selections in Section IV-B5. In addition, we also deployed several variants of our method to validate the effectiveness of different modules:

Our Method (w/o HCS) is a variant of our method without using high-confidence selection, only using soft pseudo labels.

Our Method (w/o SL) is a variant of our method only using hard pseudo labels instead of soft labels.

Our Method (w/o SD) is a variant of our method without using self-distillation framework, which is trained by the distantly supervised training data with early stopping.

4) Evaluation Metrics: Following the typical IOB-tagging evaluation metrics, we adopt Precision, Recall, F1 score to evaluate the performance of proposed methods. Different from the above metrics for resume block classification, the precision returns a positive predictive rate, and the recall gives a true positive rate. These can be defined as:

$$\text{Precision} = \frac{\text{The number of true positive prediction}}{\text{Total number of positive prediction}}, \quad (16)$$

$$\text{Recall} = \frac{\text{The number of true positive prediction}}{\text{Total number of actual positives}}. \quad (17)$$

F1 measure is a combination of precision and recall, and gives the harmonic mean of them. It can be defined by:

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (18)$$

5) Results: The overall performance for intra-block information extraction is shown in Table IV. Clearly, we observe that our method achieves the best performance on all tags, which verifies the effectiveness of our method by combining self-training learning with the distant supervision. In addition, we find that the F1 scores for some tags, such as gender, email, date, and degree, are more than 90% that indicates these information are more easily extracted since some of them have the fixed formats and the others have the finite number of values. Meanwhile, we also note that "D&R Match" method achieves very high precision score but low recall score, resulting in a worst F1 score on most tags. It is worth noting that "BERT+BiLSTM+CRF" are more suitable for fully-supervised scenario. Therefore, it achieves worse performances in the distantly supervised setting.

6) Ablation Study: To evaluate the relative influences of different modules of our distantly supervised model, we conducted ablation test for several variants as we mentioned before. The experimental results are reported in Table V. It is observed that our method achieves the best F1 score when we use the soft labels with high confidence token selection in the self-distillation framework. It is noted that the performance for our method (w/o SD) decreases the most that denotes the importance of self-distillation learning for distantly supervised training. Obviously, it makes the model robust to

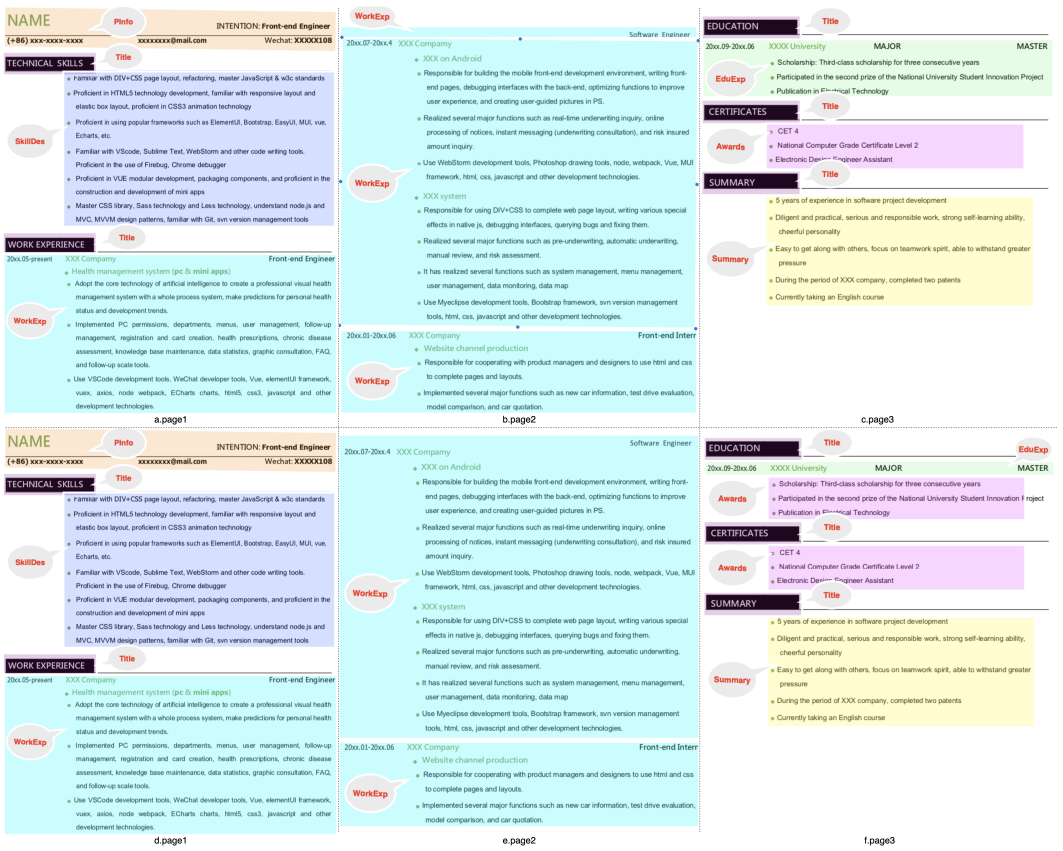


Fig. 3. The case study for LayoutXLM (subfigure a, b and c) and our method (subfigure d, e and f) on a real-world resume with three pages.

the training data with noise. In addition, we observe that soft label performs better than hard label slightly that indicates the soft label has much more effective information than the hard label. Moreover, high confidence token selection also improve the performance of model that denotes such selection process further enhances the model robustness to noise.

7) Case Study: To further illustrate the effectiveness of our hierarchical multimodal pre-training model, we perform a case study on resume block classification. As shown in Figure 3, we present a real-word resume with three pages. To protect the privacy of the candidate, we delete the personal information and mask the sensitive information. In Figure 3, the subfigure a, b and c are the results of resume block classification generated by the best baseline LayoutXLM, and the subfigure d, e and f are from our model. We observe that there exists two differences for the two methods. The first difference is *Awards* tag. In the original resume, the candidate inserts some scholarship information into the education experience. LayoutXLM classified all contents into *EduExp* tag, but our method recognized the scholarship information as *Awards* tag. Obviously, our approach is more rational. The second difference is *WorkExp* tag. LayoutXLM recognized four work experiences while our method gave three work experiences. Clearly, LayoutXLM made a mistake at the second work experience at page 2. The potential reason is that LayoutXLM can not process all content in the resume at once so that it is easy to miss the contextual information when it accepts the local content as the input. In addition, It took 4.28s for LayoutXLM to classify the resume block, while our method only spent 0.29s. It improved about 15 times for the running efficiency. Due to the effectiveness of our resume semantic structure extraction, we have deployed our method on Baidu Cloud as a solution for intelligent talent recruitment [9].

VI. Conclusion

In this paper, we divided the resume semantic structure understanding as two typical tasks, i.e., resume block classification and intra-block information extraction. For the first task, we proposed a multi-modal pre-training model based on the hierarchical Transformer encoder, in which three self-supervised objectives are designed to effectively train model parameters that reduce the dependence on the labeled data. For the second task, we introduced a distantly supervised sequence labelling method trained in self-distillation based self-training framework, which can improve the robustness of the model against the noise data in the distant supervision scenario. Finally, extensive experiments on the real-world resume datasets clearly demonstrated the effectiveness of our approaches. In the future work, we will explore to apply our hierarchical pre-training model in the other kinds of documents.

[9]https://cloud.baidu.com/solution/recruitment

References

[1] C. Qin, H. Zhu, T. Xu, C. Zhu, C. Ma, E. Chen, and H. Xiong, "An enhanced neural network approach to person-job fit in talent recruitment," *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 2, pp. 1–33, 2020.

[2] Y. Luo, H. Zhang, Y. Wen, and X. Zhang, "Resumegan: an optimized deep representation learning framework for talent-job fit via adversarial learning," in *CIKM*, 2019, pp. 1101–1110.

[3] K. Yao, J. Zhang, C. Qin, P. Wang, H. Zhu, and H. Xiong, "Knowledge enhanced person-job fit for talent recruitment," in *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 2022, pp. 3467–3480.

[4] C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 2018, pp. 25–34.

[5] Y. Ye, H. Zhu, T. Xu, F. Zhuang, R. Yu, and H. Xiong, "Identifying high potential talent: A neural network based dynamic social profiling approach," in *ICDM*. IEEE, 2019, pp. 718–727.

[6] Y. Yang, J. Zhang, F. Gao, X. Gao, and H. Zhu, "DOMFN: A divergence-orientated multi-modal fusion network for resume assessment," in *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*. ACM, 2022, pp. 1612–1620.

[7] D. Shen, C. Qin, H. Zhu, T. Xu, E. Chen, and H. Xiong, "Joint representation learning with relation-enhanced topic models for intelligent job interview assessment," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 1, pp. 1–36, 2021.

[8] C. Qin, H. Zhu, C. Zhu, T. Xu, F. Zhuang, C. Ma, J. Zhang, and H. Xiong, "Duerquiz: A personalized question recommender system for intelligent job interview," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2165–2173.

[9] C. Qin, K. Yao, H. Zhu, T. Xu, D. Shen, E. Chen, and H. Xiong, "Towards automatic job description generation with capability-aware neural networks," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[10] H. Sheng *et al.*, "Entity extraction method of resume information based on deep learning," *Computer engineering and design*, vol. 39, no. 12, pp. 3873–3878, 2018.

[11] J. Chen, L. Gao, and Z. Tang, "Information extraction from resume documents in pdf format," *Electronic Imaging*, vol. 2016, no. 17, pp. 1–8, 2016.

[12] X. Li, H. Shu, Y. Zhai, and Z. Lin, "A method for resume information extraction using bert-bilstm-crf," in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*. IEEE, 2021, pp. 1437–1442.

[13] K. Yao, C. Qin, H. Zhu, C. Ma, J. Zhang, Y. Du, and H. Xiong, "An interactive neural network approach to keyphrase extraction in talent recruitment," in *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 2021, pp. 2383–2393.

[14] C. Ayishathahira, C. Sreejith, and C. Raseek, "Combination of neural networks and conditional random fields for efficient resume parsing," in *2018 International CET Conference on Control, Communication, and Computing (IC4)*. IEEE, 2018, pp. 388–393.

[15] S. Zu and X. Wang, "Resume information extraction with a novel text block segmentation algorithm," *Int J Nat Lang Comput*, vol. 8, pp. 29–48, 2019.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *KDD*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020, pp. 1192–1200.

[18] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. A. F. Florêncio, C. Zhang, W. Che, M. Zhang, and L. Zhou, "Layoutlmv2: Multi-modal pre-training for visually-rich document understanding," in *ACL/IJCNLP*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Association for Computational Linguistics, 2021, pp. 2579–2591.

[19] Y. Xu, T. Lv, L. Cui, G. Wang, Y. Lu, D. Florêncio, C. Zhang, and F. Wei, "Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding," *CoRR*, vol. abs/2104.08836, 2021.

[20] K. Yu, G. Guan, and M. Zhou, "Resume information extraction with cascaded hybrid model," in *ACL*, K. Knight, H. T. Ng, and K. Oflazer, Eds. The Association for Computer Linguistics, 2005, pp. 499–506.

[21] S. Pawar, R. Srivastava, and G. K. Palshikar, "Automatic gazette creation for named entity recognition and application to resume processing," in *Proceedings of the 5th ACM COMPUTE Conference: Intelligent & scalable system technologies, Pune, India, January 23-24, 2012*, R. K. Shyamasundar and L. Shastri, Eds. ACM, 2012, p. 15.

[22] Y. Wentan and Q. Yupeng, "Chinese resume information extraction based on semi-structured text," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 11 177–11 182.

[23] L. Pham Van, S. Vu Ngoc, and V. Nguyen Van, "Study of information extraction in resume." Conference, 2018.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[25] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-

training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[26] M. Wei, Y. He, and Q. Zhang, "Robust layout-aware ie for visually rich documents with pre-trained language models," in *SIGIR*, 2020, pp. 2367–2376.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[28] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 7370–7377.

[29] Y. Yang, K. Wang, D. Zhan, H. Xiong, and Y. Jiang, "Comprehensive semi-supervised multi-modal learning," in *IJCAI*. ijcai.org, 2019, pp. 4092–4098.

[30] Y. Yang, Y. Wu, D. Zhan, Z. Liu, and Y. Jiang, "Deep robust unsupervised multi-modal network," in *AAAI*. AAAI Press, 2019, pp. 5652–5659.

[31] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[32] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang, "Visualbert: A simple and performant baseline for vision and language," *CoRR*, vol. abs/1908.03557, 2019.

[33] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: pre-training of generic visual-linguistic representations," in *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.

[34] Y. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "UNITER: universal image-text representation learning," in *ECCV*, vol. 12375. Springer, 2020, pp. 104–120.

[35] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 12375. Springer, 2020, pp. 121–137.

[36] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[38] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[39] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*, Y. Bengio and Y. LeCun, Eds., 2015.

[41] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, C. E. Brodley and A. P. Danyluk, Eds. Morgan Kaufmann, 2001, pp. 282–289.

[42] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[43] S. Mukherjee and A. H. Awadallah, "Distilling transformers into simple neural networks with unlabeled transfer data," *CoRR*, vol. abs/1910.01769, 2019.

[44] B. Zoph, G. Ghiasi, T. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," in *NeurIPS 2020*, 2020.

[45] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, "Empower sequence labeling with task-aware neural language model," in *AAAI*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 5253–5260.

[46] Y. Zhang and J. Yang, "Chinese NER using lattice LSTM," in *ACL 2018*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018, pp. 1554–1564.

[47] X. Ma and E. H. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *ACL 2016*. The Association for Computer Linguistics, 2016.

[48] J. Du, E. Grave, B. Gunel, V. Chaudhary, O. Celebi, M. Auli, V. Stoyanov, and A. Conneau, "Self-training improves pre-training for natural language understanding," in *NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 5408–5418.

[49] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*. PMLR, 2016, pp. 478–487.

[50] X. Li, L. Bing, W. Zhang, and W. Lam, "Exploiting bert for end-to-end aspect-based sentiment analysis," *arXiv preprint arXiv:1910.00883*, 2019.

[51] E. Chapuis, P. Colombo, M. Manica, M. Labeau, and C. Clavel, "Hierarchical pre-training for sequence labelling in spoken dialog," *arXiv preprint arXiv:2009.11152*, 2020.

[52] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, "Docbank: A benchmark dataset for document layout analysis," in *COLING 2020*, D. Scott, N. Bel, and C. Zong, Eds. International Committee on Computational Linguistics, 2020, pp. 949–960.

[53] J. Shang, L. Liu, X. Gu, X. Ren, T. Ren, and J. Han, "Learning named entity tagger using domain-specific dictionary," in *EMNLP*. Association for Computational Linguistics, 2018, pp. 2054–2064.