RESUME INFORMATION EXTRACTION WITH A NOVEL TEXT BLOCK SEGMENTATION ALGORITHM

Shicheng Zu and Xiulai Wang

Post-doctoral Scientific Research Station in East War District General Hospital,
Nanjing, Jiangsu 210000, China

ABSTRACT

In recent years, we have witnessed the rapid development of deep neural networks and distributed representations in natural language processing. However, the applications of neural networks in resume parsing lack systematic investigation. In this study, we proposed an end-to-end pipeline for resume parsing based on neural networks-based classifiers and distributed embeddings. This pipeline leverages the position-wise line information and integrated meanings of each text block. The coordinated line classification by both line type classifier and line label classifier effectively segment a resume into predefined text blocks. Our proposed pipeline joints the text block segmentation with the identification of resume facts in which various sequence labelling classifiers perform named entity recognition within labelled text blocks. Comparative evaluation of four sequence labelling classifiers confirmed BLSTM-CNNs-CRF's superiority in named entity recognition task. Further comparison among three publicized resume parsers also determined the effectiveness of our text block classification method.

KEYWORDS

Resume Parsing, Word Embeddings, Named Entity Recognition, Text Classifier, Neural Networks

1. INTRODUCTION

The recent decade witnessed the rapid evolution of recruitment from traditional job fairs to web-based e-recruiting platforms. The resume is a formal document used by the job seekers to exhibit their experiences and capabilities to the Human Resources of the targeted companies or head-hunters for landing their desired jobs. Statistical analytics shows that the well-known 3rd-party e-recruiting portals, i.e., Monster.com, and LinkedIn.com, are inundated by more than 300 million personal resumes uploaded every year. The vast amounts of personal data draw attention from the researchers because of its immense potential applications, i.e., resume routing, resume ontology creation, automatic database construction, applicants modelling, and resume management. Most e-recruiting portals require the job seekers to reformat their resumes according to their specified formats during the profile creation. However, the applicants often upload the resumes onto the portals' repositories that are under their compositions. The diverse formats include varying font size, font colour, font type, and typesetting.

The applicants usually compose the resumes in structured tables or plain texts. Moreover, the writing styles diversify across different resumes exemplified by various synonyms or word combinations for the same section titles and the arbitrary order to arrange the text blocks. Besides, job seekers save their resumes in different file types, i.e., txt, pdf, and Docx. The uncertainties associated with the resume layouts pose a significant challenge to an efficient resume parsing and reduce the accuracy for subsequent candidate recommendation. A common consequence will be that most job seekers fail to understand why their resumes not getting

DOI: 10.5121/ijnlc.2019.8503

shortlisted. The reason could be the format of their resumes rather than their qualifications.

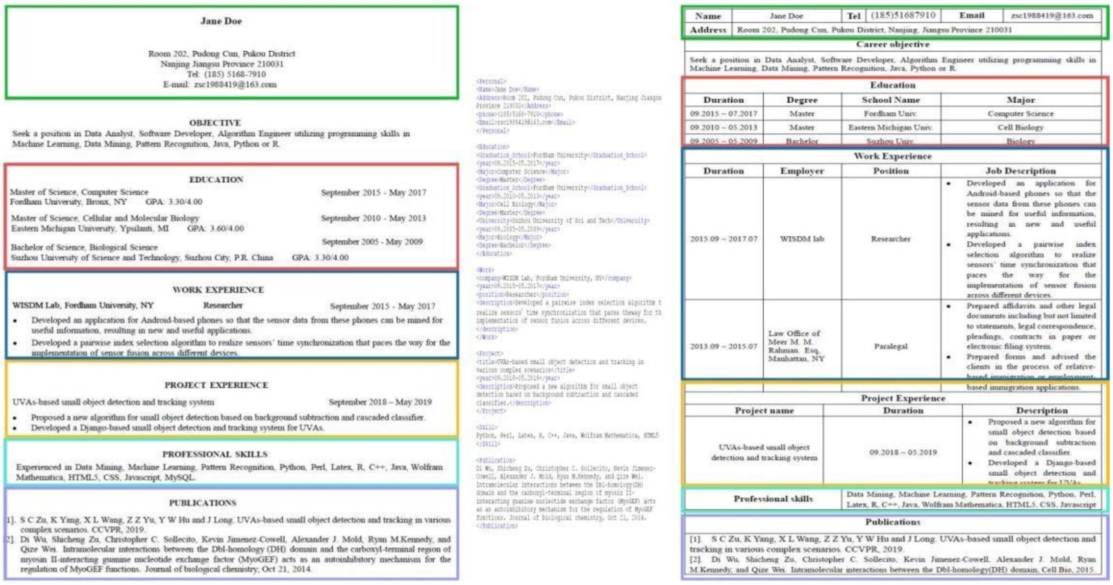
A typical resume usually adopts a document-level hierarchical structure where correlative concepts occur within the discrete text blocks. The job seekers arrange the text blocks of varying information categories in consecutive arbitrary order. The general information categories include personal contacts, career objective, self-evaluation, education background, work experiences, project experiences, professional & language skills, interests & hobbies, honours & achievements, leadership, publications, and referrers. Based on the specific text blocks, we can extract detailed facts, i.e., the phone number in personal contacts or the graduate school in education background. The left panel and right panel of Figure 1 show a representative resume composed in plain text and structured table, respectively. As shown in the left panel, the resume based on plain text adopts a hierarchical structure. The table resume, as shown in the right panel, displays the resume information in tabular form. For small items, i.e., name, Tel, and Email, the table columns usually arrange them in even number in order to form the key-value pairs. For large items, i.e., Publications and Work Experience, the titles typically span the whole columns with the detailed description listed in the next row. The order to arrange the text blocks is not fixed but follows some conventional rules. For example, the resume usually places the personal information at the top header with referrers at the footnotes. The middle panel shows an XML-formatted structured data after information extraction. The extracted items of the XML file will be transformed into a standard format for database storage or front-end rendering.

We normalize the resume parsing process by focusing on six general information fields. These general information fields are personal information, education, work experience, project experience, professional skills, and publications. We assume these fields quintessentially reflects the talent and experience of a person. Other trivial information, i.e., interests & hobbies, leadership, and referrers, vary across different resumes which are not covered in our research. Table 1 summarizes the six general information fields and the nineteen specific information fields. Most mainstream resume parsers utilize keywords to segment resumes into text blocks first. Based on each text block, detailed facts can then be derived through different features combination, i.e., lexicon features, text features, visual features, and features conjunction. Our research proposes a neural network-based approach for resume parsing that does not require extra feature engineering. Our study makes three innovative contributions to resume information extraction as follows:

➢ First, we present a novel approach for resume text block segmentation based on position- wise line information and integrated word representations within each text block. Our proposed approach employs neural network-based text classifiers and distributed embeddings. Distributed representations of words alleviate the data sparseness and conveys meaningful syntactic and semantic regularities. Our proposed method dramatically facilitates the evaluation of various neural networks-based text classifiers by waiving the labour-intensive process of constructing hand-crafted features. The coordination between the line type and the line label classifications effectively segments a resume into text blocks according to predefined labels. Quantitative comparison between five proposed text classifiers determined Attention BLSTM's superiority in text block classification and robustness over both short and long sentences (Zhou et al. 2016). Comparative evaluation among three publicized resume parsers also confirmed the effectiveness of our text block classification approach.

- Second, we present an end-to-end resume information extraction pipeline that connects the text block classification with the resume facts identification. Comparative evaluation of four neural networks-based sequence labelling classifiers indicated that BLSTM-CNNs- CRF was effective in performing named entity recognition task [2]. Based on our proposed resume information extraction pipeline, we developed an online resume parser that functions well in practice. We also show how to build an online resume information extraction system by presenting its systematic architecture.

- Third, aside from BLSTM-CNNs-CRF, most neural networks-based sequence labelling classifiers require extra engineered features to complement the word vectors. We speculated that the CNN layer of BLSTM-CNNs-CRF naturally played a role in text feature extraction. We performed the ablation experiment by removing the CNN layer from BLSTM-CNNs-CRF. The truncated BLSTM-CNNs-CRF achieved sequence labelling performance that was comparable to Bi-LSTM-CRF without text features. Finally, we carried out a comparative evaluation of different word embeddings testifying that the word representations were indispensable for named entity recognition.



Jane Doe

Room 202, Pudong Cun, Pukou District
Nanjing Jiangsu Province 210031
Tel: (185) 5168-7910
E-mail: zsc1988419@163.com

OBJECTIVE

Seek a position in Data Analyst, Software Developer, Algorithm Engineer utilizing programming skills in Machine Learning, Data Mining, Pattern Recognition, Java, Python or R.

EDUCATION

Master of Science, Computer Science September 2015 - May 2017
Fordham University, Bronx, NY GPA: 3.30/4.00

Master of Science, Cellular and Molecular Biology September 2010 - May 2013
Eastern Michigan University, Ypsilanti, MI GPA: 3.60/4.00

Bachelor of Science, Biological Science September 2005 - May 2009
Suzhou University of Science and Technology, Suzhou City, P.R. China GPA: 3.30/4.00

WORK EXPERIENCE

WISDM Lab, Fordham University, NY Researcher September 2015 - May 2017

- Developed an application for Android-based phones so that the sensor data from these phones can be mined for useful information, resulting in new and useful applications.
- Developed a pairwise index selection algorithm to realize sensors' time synchronization that paces the way for the implementation of sensor fusion across different devices.

PROJECT EXPERIENCE

UVAs-based small object detection and tracking system September 2018 – May 2019

- Proposed a new algorithm for small object detection based on background subtraction and cascaded classifier.
- Developed a Django-based small object detection and tracking system for UVAs.

PROFESSIONAL SKILLS

Experienced in Data Mining, Machine Learning, Pattern Recognition, Python, Perl, Latex, R, C++, Java, Wolfram Mathematica, HTML5, CSS, Javascript, MySQL.

PUBLICATIONS

1]. S C Zu, K Yang, X L Wang, Z Z Yu, Y W Hu and J Long. UVAs-based small object detection and tracking in various complex scenarios. CCVPR, 2019.
2]. Di Wu, Shicheng Zu, Christopher C. Sollecito, Kevin Jimenez-Cowell, Alexander J. Mold, Ryan M.Kennedy, and Qize Wei. Intramolecular interactions between the Dbl-homology (DH) domain and the carboxyl-terminal region of myosin II-interacting guanine nucleotide exchange factor (MyoGEF) acts as an autoinhibitory mechanism for the regulation of MyoGEF functions. Journal of biological chemistry, Oct 21, 2014.

```
<Personal>
<Name>Jane Doe</Name>
<Address>Room 202, Pudong Cun, Pukou District, Nanjing Jiangsu
Province 210031</Address>
<phone>(185)5168-7910</phone>
<Email>zsc1988419@163.com</Email>
</Personal>

<Education>
<Graduation_School>Fordham University</Graduation_School>
<year>09.2015-05.2017</year>
<Major>Computer Science</Major>
<Degree>Master</Degree>
<Graduation_School>Fordham University</Graduation_School>
<year>09.2010-05.2013</year>
<Major>Cell Biology</Major>
<Degree>Master</Degree>
<University>Suzhou University of Sci and Tech</University>
<year>09.2005-05.2009</year>
<Major>Biology</Major>
<Degree>Bachelor</Degree>
</Education>

<Work>
<company>WISDM Lab, Fordham University, NY</company>
<year>09.2015-05.2017</year>
<position>Researcher</position>
<description>Developed a pairwise index selection algorithm t
realize sensors' time synchronization that paces theway for th
implementation of sensor fusion across different devices.
</description>
</Work>

<Project>
<title>UVAs-based small object detection and tracking in
Various complex scenarios</title>
<year>09.2018-05.2019</year>
<description>Proposed a new algrithm for small object
detection based on background subtraction and cascaded
classifier.</description>
</Project>

<Skill>
Python, Perl, Latex, R, C++, Java, Wolfram Mathematica, HTML5
</Skill>

<Publication>
Di Wu, Shicheng Zu, Christopher C. Sollecito, Kevin Jimenez-
Cowell, Alexander J. Mold, Ryan M.Kennedy, and Qize Wei.
Intramolecular interactions between the Dbl-homology(DH)
domain and the carboxyl-terminal region of myosin II-
interacting guanine nucleotide exchange factor (MyoGEF) acts
as an autoinhibitory mechanism for the regulation of MyoGEF
functions. Journal of biological chemistry, Oct 21, 2014.
</Publication>
```

Name	Jane Doe	Tel	(185)51687910	Email	zsc1988419@163.com
Address	Room 202, Pudong Cun, Pukou District, Nanjing, Jiangsu Province 210031				

Career objective

Seek a position in Data Analyst, Software Developer, Algorithm Engineer utilizing programming skills in Machine Learning, Data Mining, Pattern Recognition, Java, Python or R.

Education

Duration	Degree	School Name	Major
09.2015 ~ 07.2017	Master	Fordham Univ.	Computer Science
09.2010 ~ 05.2013	Master	Eastern Michigan Univ.	Cell Biology
09.2005 ~ 05.2009	Bachelor	Suzhou Univ.	Biology

Work Experience

Duration	Employer	Position	Job Description
2015.09 ~ 2017.07	WISDM lab	Researcher	• Developed an application for Android-based phones so that the sensor data from these phones can be mined for useful information, resulting in new and useful applications. • Developed a pairwise index selection algorithm to realize sensors' time synchronization that paces the way for the implementation of sensor fusion across different devices.
2013.09 ~ 2015.07	Law Office of Meer M. M. Rahman, Esq, Manhattan, NY	Paralegal	• Prepared affidavits and other legal documents including but not limited to statements, legal correspondence, pleadings, contracts in paper or electronic filing system. • Prepared forms and advised the clients in the process of relative-based immigration or employment-based immigration applications.

Project Experience

Project name	Duration	Description
UVAs-based small object detection and tracking system	09.2018 ~ 05.2019	• Proposed a new algorithm for small object detection based on background subtraction and cascaded classifier. • Developed a Django-based small object detection and tracking system for UVAs.

Professional skills	Data Mining, Machine Learning, Pattern Recognition, Python, Perl, Latex, R, C++, Java, Wolfram Mathematica, HTML5, CSS, Javascript

Publications

[1]. S C Zu, K Yang, X L Wang, Z Z Yu, Y W Hu and J Long. UVAs-based small object detection and tracking in various complex scenarios. CCVPR, 2019.
[2]. Di Wu, Shicheng Zu, Christopher C. Sollecito, Kevin Jimenez-Cowell, Alexander J. Mold, Ryan M.Kennedy, and Qize Wei. Intramolecular interactions between the Dbl-homology(DH) domain. Cell Bio, 2015.

Figure 1. The left and right panels show a representative resume in plain text and structured table, respectively. The personal information, education, work experience, project experience, skill, and publication are outlined by the green box, the red box, the blue box, the yellow box, the cyan box, and the violet box, respectively. The middle panel is the generated XML structured data after resume parsing.

Table 1. Predefined Information fields for Resume Information Extraction.

Personal	Education	Work	Project	Skill	Publication
Name;	University;	Company;	Title;	Language;	Reference;
Address;	Graduate school;	Job title;	Project Period;	Computer ;	
Phone;	Graduation Date;	Work Period;	Project Description;		
Email;	Major;	Job Description;			
	Degree;				

2. Related Works

The resumes accessible from the Internet can be classified into two categories: the free plain texts and structured marked-up texts. The Information extraction algorithms primarily designed for plain texts tend to avoid any generalization over structured marked-up texts. It is because these algorithms are based on lexicons and grammars and do not make use of the extralinguistic elements such as the HTML tags. On the other hand, the Information extraction algorithms designed for structured marked-up texts are ineffective over plain texts. It is because they are unable to overcome the data sparseness because of the high flexibility of natural language.

2.1. Resume Parsing Based On Plain Text

Approaching the resume information as a hierarchical structure rather than a flat structure, Yu et al. presented a semantic-based cascaded hybrid model for resume entity extraction [3]. In the first pass, the Hidden Markov model (HMM) segmented a resume into consecutive blocks of diverse information types [3]. Then based on the classification, the second pass used HMM to extract the detailed educational information and SVM to obtain the detailed personal information [3]. By treating the resume information as a flat structure, Chen et al. proposed a two-step resume information extraction algorithm [4]. One novel contribution was the introduction of a syntax feature, the Writing Style, to model each sentence of a resume [4]. In the first step, lines of raw text were classified into semi-structured text blocks [4]. In the second step, the algorithm utilized the naïve Bayes classif ier to identify facts from the text blocks based on the Writing Style [4]. Forgoing the step of segmenting the resumes into text blocks, Chen et al. later presented a knowledge extraction framework for resumes parsing based on text classifiers [5]. Chen et al. classified the composition of each text line into three labels: Simple, Key-Value, and Complex [5]. The resulting semi-structured data was then utilized to extract facts with text classifiers [5]. Han et al. proposed a Support Vector Machine based metadata extraction [6]. This method performed independent line classification followed by an iterative contextual line classification to improve the classification accuracy using the neighbour lines' predicted labels [6]. The structured pattern of the data, the domain-specific word clustering, and feature normalization improved the metadata extraction performance [6]. In many situations, the resumes also present themselves in tabular form. The complex and ambiguous table elements pose difficulties for traditional sequence labelling techniques. By employing Conditional Random Field as the classifier, Pinto et al. successfully classified each constituent table line with a predefined tag indicating its corresponding function, i.e., table header, separator and data row [7]. PROSPECT is a recruitment support system that allows the screeners to quickly select candidates based on specified filtering criteria or combinations of multiple queries [8]. PROSPECT's resume parser is comprised of three parts: Table Analyzer (TA), Resume Segmenter (RS), and Concept Recognizer (CR) [8]. TA is responsible for structurally dissecting tables into categories and extracting information from them [8]. RS segments resume into predefined, homogeneous, consecutive textual blocks [8]. The segmented textual sections are then parsed by CRF-based extractors to derive named entities [8]. AIDAS is an intelligent approach to interpret logical document structures from PDF documents [9]. Based on the shallow grammars, AIDAS assigns a set of functions to each layout object. These functions incrementally fragment each layout object into smaller objects via a bottom-up fashion until every item is annotated by a domain ontology [9]. In terms of keywords-based resume parsing, Maheshwari et al. built a query to improve the efficiency of candidate ranking by filtering specific skill types and values [10]. The skill information was employed to extract Skill Type Feature Set (STFS) and Skill Value Feature Set (SVFS) [10]. The Degree of Specialness (DS) can be calculated on these two feature sets to screen out the most suitable candidates.

2.2. Resume Parsing based on Websites

Since most 3rd-party recruitment portals use web pages to exhibit the resumes, the researchers also investigate various parsing techniques for web-related resumes. Ji et al. applied a tag tree algorithm to extract records and schema from resume web pages based on the Document Object Model (DOM) [11]. In DOM, the internal nodes represent the attributes, while the leaf nodes consist of the detailed facts. This algorithm parses different web pages into the tag trees from which the tag tree templates can be generated via a cost-based tree similarity metric [11]. The tag tree templates can parse exclusive contents from each resume [11]. The facts can be derived from the exclusive contents by finding repeated patterns or heuristic rules [11]. EXPERT is one of the existing resume recommendation systems that leverage the ontology to model the profiles of job seekers [12]. This system constructs the ontology documents for the features of collected resumes and job postings, respectively [12]. Then EXPERT retrieves the eligible candidates by mapping the job requirement ontology onto the candidate ontology via similarity measurement [12]. Cravegna et al. presented a rule-based adaptive parser for web-related text based on Learning Pattern by Language Processing (LP)2 algorithm [13, 14]. This algorithm learns rules by generalizing over a set of instances in a training corpus marked by XML tags [13, 14]. This algorithm performs training by inducing a set of tagging rules, followed by tagging imprecision correction [13, 14]. The shallow NLP is utilized to generalize rules beyond the flat word sequences since it limits the data sparseness and solves the overfitting issue [13, 14].

3. Neural Networks-Based Resume Parsing Algorithm

3.1. Text Block Segmentation

Whereas most job seekers composed their resumes with diverse formats, they usually order the text blocks by following the conventional rule. The job seekers typically place their personal information at the top part, followed by education background, work experience, project experience, professional skills, and publications. Besides that, the items of integrating meanings are normally grouped within each text block. The position-wise line information and correlated word representations within each text block provide vital clues for text block segmentation. However, most prevalent resume parsers use tools such as Tika to obliterate the layout information during the pre-processing step and perform text block classification on the extracted text using regular expressions or hand-crafted rules. One disadvantage of format removal is that significant amounts of positional information are lost which is supposed to provide extra discrimination. Another shortcoming is that the regular expressions and hand-crafted rules cannot generalize well if not properly defined, resulting in limited usage of these methods. The machine learning methods, however, are robust and adaptable, usually achieving higher recalls than fuzzy keyword matching. By leveraging the resumes' layout information and integrated word representations within each text block, we proposed a novel text block classification method based on neural networks dispensing the feature construction step required by machine learning algorithms such as SVM and naïve Bayes. Three reasons motivate us to propose this model. First, neural network-based feature extraction outperforms hand-crafted features in capturing more semantics from texts, i.e., contextual information and word order in texts and suffering less from the data sparseness problem. Second, word embeddings served as “universal feature extractors” can better represent words than human-designed features. Third, pre-trained word embeddings are convenient to use. The text classifiers only need to fine-tune the hyperparameters for the specific downstream classification tasks after applying to the word vectors.

Our proposed text block classification method trains two kinds of line classifiers, a line type classifier, and a line label classifier. The line type classifier roughly divides the resumes into general sections based on four types of generic layouts, which are header, content, metadata, and footer. The header occupies the topmost part of a document usually containing the page numbers, the company logos, the section titles, and the communication addresses. In resumes, job seekers usually include their personal information at the top header. The content is the resume entity. The metadata is about the file-specific information, including authors, file creators, file creation date, file modification date, and producers. The footer occupies the bottom part of a document usually containing the page numbers and the referrers.

The rough segmentation is further refined by exquisite line label classification based on six general information fields: personal, education, work, project, skill, and publication. We expect the coordinated line classification performed by these two classifiers will generate contiguous line label clusters. In each line cluster, the neighbor lines share the same line labels indicative of boundaries for text blocks.

To feed the training dataset with appropriate format into the neural network, we converted the line lists in the text resumes into word vectors. To realize this, we built the pre-trained word embeddings specific for the resume-related domain. The sentences from all collected resumes were gathered by concatenating the line lists from all text resumes. We unified the sentence punctuations by replacing different kinds of punctuations with spaces. The total number of sentences amounted to 75000. We utilized the Word2Vec [19] model from the gensim tool as the default model to train word embeddings by applying to the sentence collections or corpora essentially. Other word representation models were also evaluated, such as GloVe [20], and BERT [21]. These embeddings would be fine-tuned during training. We set the word embeddings dimension to 300 and saved the generated word embeddings as .bin file. We iteratively split each line to retrieve the line type, line label, and the line content based on the tab spaces. After removing the stop words, every word in line segments was tokenized and transformed into word vector $e_i = W^{wrd} v^i$ by looking up its vocabulary index in word embeddings where ranged from 0 to . The sentence-level word vectors were represented as $emb_s = \{e_1, e_2, \ldots, e_T\}$ where the T was the sentence length. We placed various text classifiers on top of the word vectors to optimize their hyper-parameters in accordance with the predefined categories. To classify each line into the correct category, we considered five text classifiers with a summary for each one as follows.

- ★ **Text-CNN**. To perform line classification, a simple CNN was trained with its one convolutional layer to sit on top of word vectors [22]. We kept the word vectors static and optimized other parameters according to the predefined categories through backpropagation [22]. Text-CNN used fixed-size windows to capture contextual features and max-pooling layers to determine discriminative features on feature maps [22]. The sliding of the convolutional kernel resembled the n-grams in a sense that the multi-scales n-grams correspond to different window sizes [22].

- ★ **RCNN**. When learning word representations, RCNN used its bi-directional recurrent structure to capture more contextual information compared to the traditional window-based CNN. RCNN also preserved CNN's discriminative capability by using its max-pooling layers to determine key semantic factors in a text [23].

- ★ **Adversarial LSTM**. Adversarial training and virtual adversarial training have been proven to be effective regularization strategies. Miyato et al. applied perturbations to the initial pre-trained word embeddings in conventional LSTM [24]. We expected that adversarial training and virtual adversarial training improved not only the model's robustness against overfitting but also the quality of the original word embeddings after training the modified LSTM [24].

- ★ **Attention BLSTM**. Since the unidirectional LSTM only processes the word sequences in a forward pass, Bi-LSTM compensates for this disadvantage by introducing a second SLTM layer [1]. For each word, the two hidden states flow in opposite directions outputting a concatenated hidden state. As a result, the contextual information from the past and the future can be learned [1]. Besides, an attention mechanism was involved in capturing the most decisive words in a sentence for text classification by merging word-level features into a sentence-level feature vector through weight vector multiplication [1].

- ★ **Transformer**. The transformer is a sequence transduction model connecting the encoder and the decoder through a Multi-Head Attention mechanism to draw global dependencies between input and output [25]. Self-attention correlates different parts of a single sequence in order to compute a sentence representation [25]. The number of operations for relaying signals from two arbitrary input is a constant number admitting parallel computing [25]. It is worth noting that the transformer should be placed on top of the one-hot positional embeddings rather than the word embeddings.

3.2. Resume Facts Identification

3.2.1. Text Sequence Labelling

The coordination between line type classification and line label classification can determine the boundaries for text blocks. Our subsequent task is to derive facts from each text block via named entity recognition (NER). NER tags phrases in a sentence with predefined named entity keys such as street addresses, university/graduate school names, majors, degrees, departments, company names, job positions, computer skills, and language names. A massive effort has been taken to collect the standard dictionaries for the named entities. We obtained the data from various media on the Internet. The accredited university/graduate school names, the degrees conferred, and the registered majors can be acquired from the official websites of the Ministry of Education. The gazetteers are updated by the Civil Administration department regularly. The Industrial and Commercial Bureau curates the official company names while job position names and computer skills can be extracted from the websites of the 3rd-party recruitment portals. The collected named entity features were used to train our sequence labelling classifiers with BIO annotation. When the classifiers performed the NER, the classifiers assigned a probability distribution to each phrase on different classes. To map the named entity candidates to the standard attribute names, we employed the k-means algorithm to cluster the identified named entities by computing the cosine similarities between them based on Term Frequency–Inverse Document Frequency (TFIDF).

In our study, we evaluated four prevalent sequence labelling classifiers, namely Bi-LSTM-CRF [29], Bi-GRU-CRF [30], IDCNN-CRF [31], and BLSTM-CNNs-CRF [2] in terms of NER performance and decoding speed. A summary for each classifier is as follows.

- ★ **Bi-LSTM-CRF**. This model combines a bidirectional LSTM and a CRF to form a Bi-LSTM-CRF network. Word sequences are projected into dense vectors and then concatenated with extra features. The concatenated vectors make up the input for recurrent bidirectional LSTM layer. Outputs of forward and backward layers are concatenated and projected to score each label. A final CRF layer is used to overcome the label bias problem.

- ★ **Bi-GRU-CRF**. In this model, the stacked Bi-GRUs of reverse directions takes word vectors and engineered features as input. The stacked Bi-GRUs are coupled with a CRF layer. The concatenated output from Bi-GRUs is fed into the CRF layer to decode the label sequences jointly.

- **IDCNN-CRF**. CNN has limited representations for large context due to its fixed-size windows. Iterated Dilated CNNs (ID-CNNs) fix this problem by stacking layers of dilated convolutions of exponentially increasing dilation width. These dilated convolutions allow incorporation of global context without losing resolutions. In contrast to CNN in which convolutions transform adjacent input, the ID-CNNs transform a wider input by skipping over δ input where δ is the dilation width. Besides, the ID-CNNs enable convolutions to run in parallel across the entire documents.

- **BLSTM-CNNs-CRF**. BLSTM-CNNs-CRF is an end-to-end neural network for sequence labelling requiring no engineered features or data pre-processing beyond word vectors. The model makes use of both word-level and character-level representation that will be fed into the bi-directional LSTM to model context information of each word. The CRF then uses the output from the BLSTM to jointly predict labels for the whole sequence.

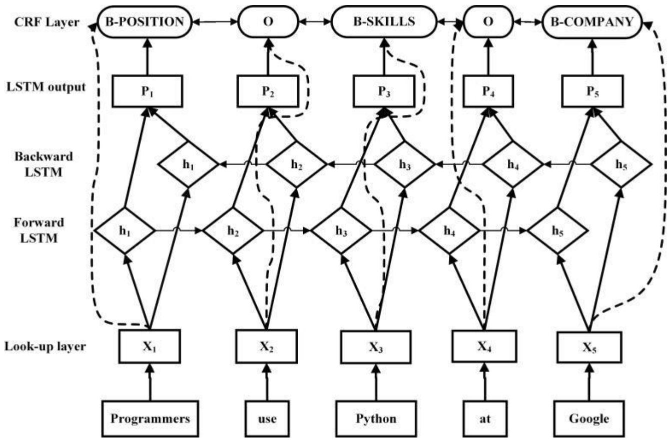


Figure 2. Feature Connection for Sequence Labelling Classifiers

3.2.2. Text Features

Bi-LSTM-CRF, Bi-GRU-CRF, IDCNN-CRF can also use text features as input besides the word vectors. Text features capture variations of the word itself. In this section, we describe various text features that we extracted to complement the word vectors for NER performance. The text features used in our research are as follows:

- Starts with a capital letter.
- All letters are capitalized.
- Whether has noninitial capital letters.
- Starts with a digit.
- All characters are digits.
- Mix with letters and digits.
- Whether has punctuations.
- Whether has symbols.
- Whether has apostrophe end (’s).
- Whether has initials such as I. B. M.

We extracted 30K text features for the NER task. In the process of training models, we treated the text features the same as the word vectors. Therefore, the input of networks contained both word vectors and text features. Zhiheng Huang reported that the direct connections from word spelling and context features to the output layer expedited the training process without compromising the sequence labelling accuracy [29]. In our study, we also employed this feature connection technique to connect text features with the CRF output layer directly to avoid potential feature collisions. Figure 2 provides an instance to illustrate the feature connection. The input word sequences are “Programmers use Python at Google” where “Programmers,” “Python,” and “Google” are three named entities to be identified. We used standard BIO format

to annotate the NER where B- stands for Beginning-, I- stands for Inside-, and O stands for Others. Instead of feeding the features into the forward LSTM/GRU layer like the word vectors (solid straight arrow), a direct connection between the text features and the CRF output layer is made (dotted curved arrows). Whereas we allow for full connections between features and outputs, this direct connection accelerates the training of the classifiers without losing the sequence labelling accuracy.

3.3. Resume Information Extraction Approach

By consolidating the concepts from text block classification and resume facts identification, we present the whole procedure for resume information extraction. Figure 3 illustrates the pipeline for our proposed resume parsing algorithm. Suppose we crawl down a resume from the Internet, we use the pdfminer and docx tools to convert it into the text file by removing all the layouts. We implement data cleaning on the text resume by unifying different punctuations, removing stop words, and low-frequency words. After that, we append each line of the text resume to a line list. We iteratively convert the lines from the line list into the word vectors. We tokenize every word from each line and map it into the word vector by looking up its vocabulary index in the pre-trained word embeddings. For line type and line label classification, the input of the line type and line label classifiers do not require any engineered features beyond the word vectors. The line type classifier will categorize each line into four generic layouts. The line label classifier further refines the rough classification by classifying each line into six general information fields. This classification cascade will generate successive line clusters with shared labels indicative of the boundaries for text blocks. As a result, we segment a new resume into text blocks with predefined labels. For resume facts identification, we iteratively apply the sequence labelling classifiers to the word vectors of each text block in conjunction with the text features we design in advance. The sequence labelling classifiers will identify any named entities they can recognize. To match the named entities to the standard attribute names, the k-means algorithm is used to do attribute clustering by computing the TFIDF-based cosine similarities between the named entity candidates. After that, every named entity in clusters will be assigned a standard attribute name. At this point, we parse the detailed resume information as structured data. Finally, we transform the key-value pairs from the identified named entities to an XML file for front-end rendering or database storage.

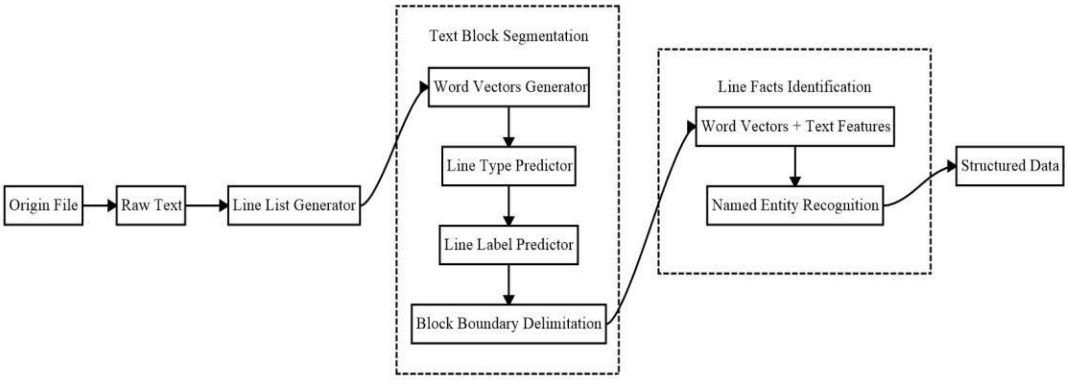


Figure 3. Pipeline for Proposed Resume Information Extraction Algorithm

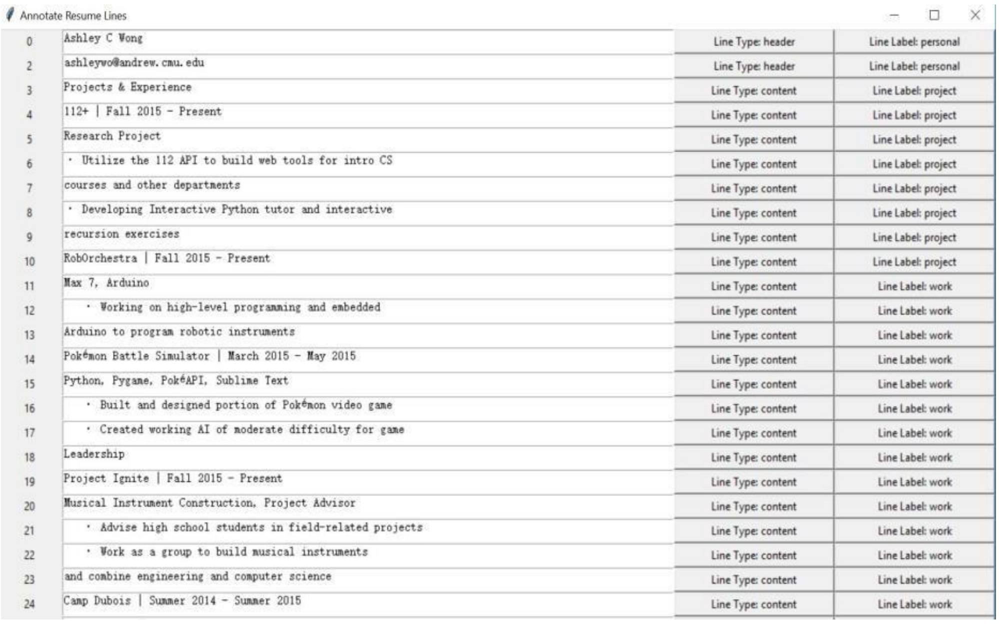
4. EXPERIMENTAL METHODOLOGY

4.1. Dataset Setup and Evaluation Metrics

To train our proposed line classifiers, we collected 5000 English resumes as our data set. To obtain the vast amount of personal data, we developed a crawler to crawl resumes of diverse formats from the English version of zhaopin.com (https://www.zhaopin.com), the largest recruitment portal in China. The collected resumes covered multiple industries. 4000 resumes were in pdf format with 1000 resumes in Microsoft Docx format. We utilized the pdfminer and docx tools to convert the pdf resumes, and Docx resumes into txt files, respectively. After that, we cleaned the txt files by replacing extra carriage return with empty space. The generated text resumes will remove any visual layouts from their original ones.

We developed a custom-make line annotation GUI as shown in Figure 4. This line annotation tool annotates each line with two categories, one for line type and one for line label. After line annotation was finished, the annotation tool saved the results as .txt files. Each line contained the line type, the line label, and the line content separated by tabs. We split the prepared dataset with 3/4 of them reserved for training and others for testing. In term of classification validation, we applied four-fold cross-validation on the training dataset.

In NLP application, precision and recall are commonly adopted to measure the performance of the classifiers. We define the Precision (P) as the proportion of all named entities that the classifiers recognize to be correct. We define the Recall (R) as the percentage of the correct named entities recognition that the classifiers achieve. The two metrics can be deemed as a measure of completeness and correctness, respectively. The F-1 measure is defined as the harmonic mean of precision and recall. It lies between the precision and recall. In our study, we used these three metrics to evaluate our proposed classifiers.



0	Ashley C Wong	Line Type: header	Line Label: personal
2	ashleywo@andrew.cmu.edu	Line Type: header	Line Label: personal
3	Projects & Experience	Line Type: content	Line Label: project
4	112+ | Fall 2015 - Present	Line Type: content	Line Label: project
5	Research Project	Line Type: content	Line Label: project
6	· Utilize the 112 API to build web tools for intro CS	Line Type: content	Line Label: project
7	courses and other departments	Line Type: content	Line Label: project
8	· Developing Interactive Python tutor and interactive	Line Type: content	Line Label: project
9	recursion exercises	Line Type: content	Line Label: project
10	RobOrchestra | Fall 2015 - Present	Line Type: content	Line Label: project
11	Max 7, Arduino	Line Type: content	Line Label: work
12	· Working on high-level programming and embedded	Line Type: content	Line Label: work
13	Arduino to program robotic instruments	Line Type: content	Line Label: work
14	Pokémon Battle Simulator | March 2015 - May 2015	Line Type: content	Line Label: work
15	Python, Pygame, PokéAPI, Sublime Text	Line Type: content	Line Label: work
16	· Built and designed portion of Pokémon video game	Line Type: content	Line Label: work
17	· Created working AI of moderate difficulty for game	Line Type: content	Line Label: work
18	Leadership	Line Type: content	Line Label: work
19	Project Ignite | Fall 2015 - Present	Line Type: content	Line Label: work
20	Musical Instrument Construction, Project Advisor	Line Type: content	Line Label: work
21	· Advise high school students in field-related projects	Line Type: content	Line Label: work
22	· Work as a group to build musical instruments	Line Type: content	Line Label: work
23	and combine engineering and computer science	Line Type: content	Line Label: work
24	Camp Dubois | Summer 2014 - Summer 2015	Line Type: content	Line Label: work

Figure 4. The custom-made line annotation tool to label lines for a resume-related dataset

4.2. Implementation of Web-based Resume Parser

Based on our proposed resume information extraction approach, we developed a Django-based resume information extraction system installed at our big data computing institute for resumes collection and statistical analysis. Thus far, we have realized three useful functions. They are resume information extraction, reformatted resume downloading, and resume filtering. We utilized the MySQL clusters for structured data storage, i.e., the operating records of the users, the resumes uploading timestamps. The resume crawler executed a night job in charging of crawling down thousands of resumes from designated recruitment portals. The system stored the original resumes in the HDFS file system with their file information curated in MySQL Clusters indexed by their resume IDs. The Django framework is based on the Model-View-Controller (MVC) model. When a registered client initiates his/her resume parsing request, the application server will pass the inquired resume ID onto the MySQL Clusters. If the MySQL clusters have previous parsing records for the inquired resume, the application server will ask for the previous parsing results from the Mongo database. For every resume, we store the extracted key-value pairs in the Mongo database as one document. The diversification of the identified named entities across different resumes makes it impossible for storing the extracted information in structured tables. The controller will use the retrieved parsing results for front-end rendering. If the MySQL clusters do not have previous parsing records, the HDFS file system will return the inquired original resume. The system will employ the proposed resume information extraction algorithm to parse the resume and save the results in MongoDB. For resume filtering module, we created a full-text index to enable text search on candidate resumes. The Lucene is an open-source text search engine specific for creating a full-text index over resumes. In similar to web-based template rendering, we can also reformat the original resumes into standard formats by rendering the predefined Docx template. Figure 5 shows a descriptive system architectural network for our online resume parser.

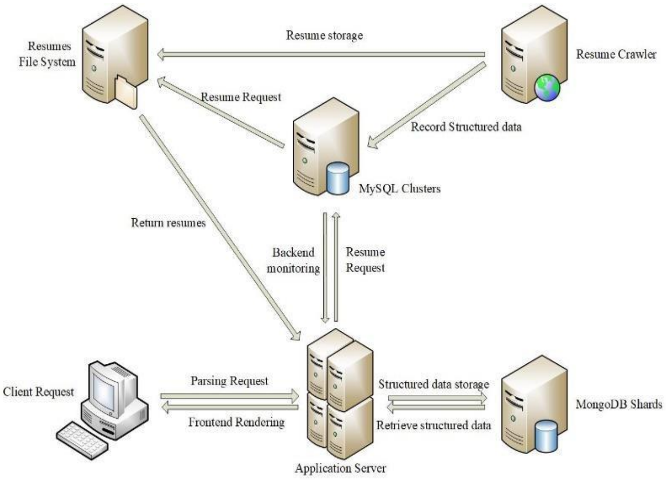


Figure 5. The systematic architecture for our web-based parser

5. RESULTS

5.1.Evaluation of Text Block Segmentation

With regards to line type classification, overall, the four generic layouts can be differentiated from each other with high classification accuracy by the five proposed text classifiers. Figure 6 shows the comparative analysis of the five text classifiers based on the line type classification. The reason for overall superior classification performance is that the four generic layouts occupy different positions in a resume. We noticed that all five text classifiers predicted the content with slightly lower recalls and F-1 measures. The reason could be that many job seekers use the header to make their personal contacts more noticeable. Hence, the lines supposed to be in the content were wrongfully classified to be in the header. As shown in Figure 6, Attention BLSTM outperformed other text classifiers by achieving a F-1 measure of 0.96, 0.93, 0.96, and 0.97 for header, content, metadata and footer, respectively.

With regards to line label classification for the six general information fields, Figure 7 shows the comparative analysis of five proposed text classifiers based on the line label classification. We draw three conclusions on the performance of the evaluated text classifiers. The first conclusion is that Attention BLSTM and Adversarial LSTM outperform other classifiers in classifying long sentences with higher recalls and F-1 measures. This observation was made by classifying long sentences in work experience, project experience, and publication. As shown in Figure 7, for the work experience, Attention BLSTM yielded a recall of 0.80 with an F-1 measure of 0.82, whereas Text-CNN achieved a recall of 0.70 with an F-1 measure of 0.73. For the project experience, Attention BLSTM obtained a recall of 0.81 with an F-1 measure of 0.83 while Text-CNN yielded a recall of 0.71 with an F-1 measure of 0.74. The performance of Adversarial LSTM and Attention LSTM was comparable. Compared to paragraphs, lines are of short or medium lengths which are not long enough to provide contextual information from previous and future states. The second conclusion is that for short phrases such as detailed personal information, Text-CNN outperforms the other text classifiers because of the independent occurrence of detailed information. Text-CNN can better capture the semantic of short phrases by utilizing its discriminative window to learn the character-level representations of the words. As shown in Figure 7, Text-CNN achieved a recall of 0.84 with an F-1 measure of 0.88 for personal information block whereas Attention BLSTM yielded a recall of 0.82 with an F1- measure of 0.86 which are slightly lower. The third conclusion is that RCNN achieves better classification performance over Text-CNN in terms of long sentences. It is because the

RCNN applies a recurrent structure to capture more contextual information compared to traditional window-based CNN. Notably, RCNN's recurrent structure does not depend on window-related convolutional kernels. The classification performance of Transformer was mediocre. We suspect the reason could be a reduced resolution due to averaging attention-weighted positions. Attention BLSTM eclipsed other text classifiers by achieving a F-1 measure of 0.86, 0.84, 0.82, 0.83, 0.86, and 0.85 for personal, educational, work, project, skills and publications, respectively. Given Attention BLSTM's favourable classification performance and strong robustness against both short and long sentences, we decided to use Attention BLSTM to segment text blocks in practical implementation.

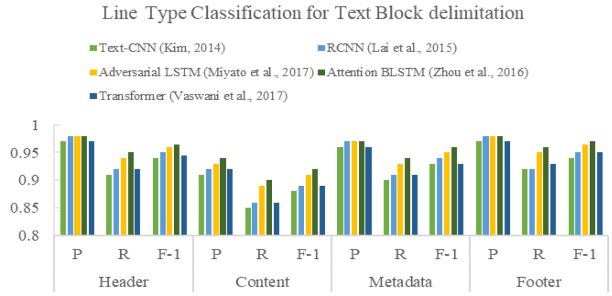


Figure 6. Precision (P), Recall (R), and F-1 Measure (F-1) of line type classification performed by five text classifiers for four generic layouts.

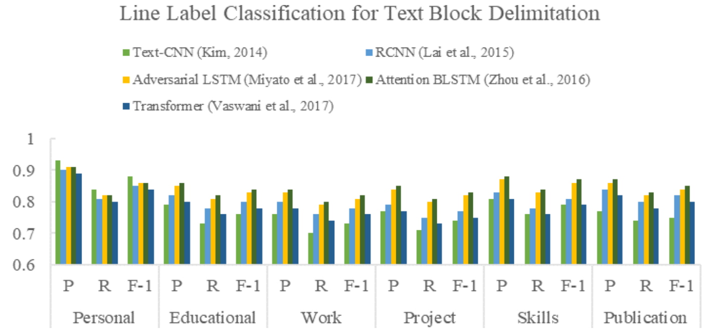


Figure 7. Precision (P), Recall (R), and F-1 Measure (F-1) of line label classification performed by five text classifiers for six general information fields.

5.2. Evaluation of Resume Facts Identification

Table 2 shows the comparative results for the resume facts identification performed by four proposed sequence labelling classifiers. For personal information, the person names have distinct spelling characteristics usually containing the first names and the last names with two capitalized initials. This feature explains its high F-1 measures. In terms of addresses, we typically write addresses conforming to the conventional rule with specific addresses at front and zip code at last. The enriched variation within its long sentence accounts for its subtly decreased F-1 measures. The phone numbers and email addresses have distinct patterns. The phone numbers contain a fixed number of digits with area codes placed in parenthesis. We inevitably write the email addresses with a username followed by a symbol "@" and a domain name. These format constraints explain why these two fields have relatively high F-1 measures. For the school names and the degrees, these fields are relatively fixed such that their F-1 measures are comparatively high. However, the writing formats for these fields can be different. For instance, some job seekers prefer to use the acronym to designate their Alma Maters. However, our collected named entity dictionaries did not include the abbreviations of school names, causing some of them missed by the classifiers. The same also applied to the degree name since graduate students frequently write M.S. rather than Master of Science. Compared to the school names, the major names have reduced F-1 measures since names for the same majors may vary at different schools. In China, colleges use bioscience and biotechnology interchangeably to designate the biology major. The F-1 measures for the graduate date are significantly reduced due to its enriched variations, such as 2010-05-19, 05/19/2010, and 2010/05/19. In terms of company name, most job seekers write their ex-employer names in full names for web cross references. Under this circumstance, the company names can be retrieved from the collected named entity dictionaries with high similarities. The job titles are hard to identify because they largely depend on the requirement of the employers. For the same position, different companies may use different job titles that are harmful to correct recognition. The F-1 measures for job description are relatively higher than job title because the job description consists of successive long sentences. These sentences are longer than others and full of technical details such as symbols and digits. These characteristics facilitate the classification of the job description. The only challenge in classifying the job description is to determine the beginning and end of the description. The reason for low F-1 measures of work period is the same as the graduate date. The project experience and work experience share the same writing format. The F-1 measures of project title, project description, and project period resemble the F- 1 measures of job title, job description, and job period, respectively. The language and computer skills are similar to university and degree since they are fixed in collected named entity dictionaries. Therefore, their F-1 measures are relatively high. When the reference is considered, it also has a distinct writing pattern. The scholars usually adopt the Harvard/Vancouver system to prepare their references. The writing format for references is authors followed by publication year, article title, journal title, volume number, and page numbers. This regex rule explains its high F-1 measures.

It is worth noting that Bi-LSTM-CRF, the Bi-GRU-CRF, and the IDCNN-CRF also employed the text features to recognize named entities. BLSTM-CNNs-CRF only utilized the word vectors for sequence labelling [2]. Based on the comparative analysis of the four sequence labelling systems, we made three conclusions. First, overall, BLSTM-CNNs-CRF outperformed the other three sequence labelling classifiers. It is because BLSTM-CNNs-CRF uses the concatenation of word vectors and character representations to model context information of each word [2]. Compared to Chinese characters, an English word is comprised of letters of finer granularity [33]. CNNs can effectively extract morphological information (such as the prefix or suffix of a word) from word characters and convolute it into neural representations. Our experimental data confirm the conclusion made by Ma et al. that character-level representations are essential for sequence labelling tasks [2]. Second, the greedy ID-CNN outperformed both Bi-LSTM and Bi-GRU when paired with Viterbi-decoding. The reason is that ID-CNNs are better token encoders than Bi-LSTMs at representing broad context without losing resolution [31]. ID-CNNs stack dilated convolutions of increasing width to incorporate global context from a whole document [31]. ID-CNNs can learn a feature function better suited for representing large text, in contrast with Bi-LSTM that encodes long memories of sequences [31]. Third, Bi-GRU slightly outperformed Bi-LSTM in our NER task when paired with the CRF. Both LSTM and GRU are variants of RNN capable of modelling long-term dependencies and solving gradient vanishing or exploding [32]. Because our dataset was not large and Bi-GRU had fewer hyper-parameters to optimize, it was easier for Bi-GRU to reach convergence compared to Bi-LSTM [32].

With regards to the decoding speed, we set the decoding speed of BLSTM-CNNs-CRF to be the baseline and compared the decoding speeds of various classifiers to that of BLSTM-CNNs-CRF. We found out IDCNN-CRF had the fastest decoding speed. When executing sequence labelling, LSTMs require $O(N)$ time complexity on sentences N of length N under GPU parallelism. For IDCNN-CRF, however, the per-token logits produced by fixed-length convolutions enable predictions to be run in parallel across entire documents. The reason why

Bi-GRU-CRF was faster than Bi-LSTM-CRF is that GRU had a simpler structure and fewer hyper-parameters to optimize.

Table 2. The F-1 Measures for resume facts identification by four sequence labelling classifiers.

Field	Bi-LSTM-CRF	Bi-GRU-CRF	IDCNN-CRF	BLSTM-CNNs-CRF
Name	0.937	0.939	0.942	0.945
Address	0.844	0.845	0.847	0.850
Phone	0.967	0.969	0.971	0.975
Email	0.963	0.965	0.968	0.971
University	0.906	0.908	0.912	0.916
Graduation School	0.904	0.907	0.910	0.915
Graduation Date	0.821	0.823	0.828	0.835
Major	0.851	0.855	0.862	0.866
Degree	0.898	0.901	0.906	0.911
Company Name	0.873	0.875	0.881	0.888
Job Title	0.843	0.844	0.850	0.853
Job Description	0.872	0.873	0.880	0.882
Work Period	0.820	0.821	0.826	0.832
Project Title	0.842	0.843	0.848	0.851
Project Description	0.873	0.873	0.881	0.883
Project Period	0.818	0.820	0.824	0.830
Language	0.908	0.910	0.911	0.913
Computer Skills	0.902	0.903	0.906	0.910
Reference	0.848	0.850	0.852	0.860
Avg.	0.878	0.880	0.885	0.889
Speed	1.13 x	1.30 x	1.70 x	1 x

5.3. BLSTM-CNNS-CRF's CNN Layer is an Effective Text Feature Extractor

Most neural networks-based sequence labelling systems utilize various features to augment rather than replace the word vectors. We managed to run the ablation study to testify that the CNN layer of BLSTM-CNNs-CRF naturally serves as the text feature extractor. The automatic extraction of text features simulates the processing of constructing various text features manually. We applied Bi-LSTM-CRF solely on the word vectors and calculated the F-1 measures for identifying various resume facts. We also removed the CNN layer from BLSTM-CNNs-CRF and let the truncated classifier perform the NER. Table 3 shows the results for the ablation experiment. We discover that the sequence labelling performance of Bi-LSTM-CRF without text features was comparable to that of the truncated BLSTM-CNNs-CRF. We choose some fields to explain our observation since we have discussed most of them in the previous section. In terms of the phone number, we designed a text feature, all letters are digits, that can account for its unique textual characteristics. Therefore, after we removed this text feature, the F-1 measures for phone number identification were significantly reduced. When the computer skill is considered, the text features, starts with a capitalized letter, and mix with letters and digits, contribute to its recognition. However, when these two text features were removed, the sequence labelling performance of Bi-LSTM-CRF was vastly degraded. In terms of email addresses, we designed the text feature, whether has symbols, to capture its distinct format. When we removed this text feature, Bi-LSTM-CRF had decreased NER performance. Given that the NER performance of Bi-LSTM-CRF was on par with that of the truncated BLSTM-CNNs-CRF, we conclude that CNN is an effective approach to extract text features or character-level information of a word.

Table 3. The F-1 measures for resume facts identification in ablation study

Field	Bi-LSTM-CRF		BLSTM-CNNs-CRF	
	+	-	+	-
Name	0.937	0.897	0.945	0.901
Address	0.844	0.812	0.850	0.813
Phone	0.967	0.918	0.975	0.920
Email	0.963	0.915	0.971	0.917
University	0.906	0.857	0.916	0.859
Graduation School	0.904	0.851	0.915	0.853
Graduation Date	0.821	0.792	0.835	0.793
Major	0.851	0.817	0.866	0.818
Degree	0.898	0.855	0.911	0.856
Company Name	0.873	0.826	0.888	0.829
Job Title	0.843	0.810	0.853	0.812
Job Description	0.872	0.823	0.882	0.824
Work Period	0.820	0.791	0.832	0.794
Project Title	0.842	0.807	0.851	0.809
Project Description	0.873	0.825	0.883	0.827
Project Period	0.818	0.787	0.830	0.789
Language	0.908	0.864	0.913	0.866
Computer Skills	0.902	0.861	0.910	0.863
Reference	0.848	0.812	0.860	0.814
Avg.	0.878	0.838	0.889	0.839

5.4. Comparative Evaluation between Four Resume Parsers

In this section, we mainly focus on comparing the text block classification performance of our resume parser to that of three publicly published resume parsers in terms of personal information, education, and work experience. Literature only provides the F1-measures for these three text blocks. These three resume parsers are PROSPECT [8], CHM [3], and Writing-Style [4]. Table 4 illustrates the F-1 measures for comparative evaluation between four resume parsers. PROSPECT, developed by IBM Research in India, aims at screening the resumes of software engineers and shortlisting the appropriate candidates for IT companies. The resumes of IT professionals always cover a shortlist of major and degree types. CHM, developed by Microsoft Research, employs HMM to segment the entire resume into consecutive blocks in the first pass. In the second pass, CHM utilizes HMM to extract detailed education information and SVM to recognize detailed personal information. Writing-Style, developed by Beijing Institute of Technology, employs the Writing Style syntactic feature to identify the appropriate blocks of the semi-structured text and different items from the same block. Overall, our proposed approach outperformed Writing-Style and CHM With regards to the classification of personal, education, and work experience. We reason it is the position-wise line information and integrated meaning of individual block that account for its superiority in text block classification. The automatic extraction of text features by neural networks can better capture the semantic features for text block delimitation. The only exception was the outstanding F-1 measure achieved by PROSPECT for education background. The reason is that PROSPECT exclusively gears toward the IT professionals resulting in limited selections of majors and degrees that help increase the precision and the recall of the classifier.

Table 4. The F-1 measures for comparative evaluation between four resume parsers.

Text block	PROSPECT	CHM	Writing-Style	Our approach
Personal	-	0.804	0.823	0.862
Education	0.921	0.730	0.792	0.841
Work experience	0.785	-	0.789	0.820

5.5. Comparative Evaluation of Different Word Embeddings

In this section, we performed a comparative study to confirm the importance of pre-trained word embeddings. We compared four algorithms for words representations, which are randomized initialization, Word2Vec [19], GloVe [20], and BERT [21]. Using a simple single-layer architecture, Word2Vec can train both CBOW and Skip-gram models on corpora by preserving the linear regularities among words [19]. Word2Vec can capture very subtle semantic relationships between words such as a city and the country it belongs to [19]. In our research, we used the Skip-gram to model the local context of a word. GloVe is a global log-bilinear regression model that leverages both matrix factorization and local context window [20]. GloVe makes use of the global corpus statistics while simultaneously capturing the meaningful linear substructures prevalent in Word2Vec [20]. Thus far, BERT is the most advanced word representation algorithm which stands for Bidirectional Encoder Representation from Transformers [21]. BERT uses masked language models to train deep bidirectional representation of words by jointly conditioning on both left and right corpora context [21]. Table 5 shows the average F-1 measures for resume facts identification achieved by BLSTM-CNNS-CRF based on these four kinds of word embeddings. The result in Table 5 indicates that the pre-trained word embeddings are indispensable for the downstream NER task. BLSTM-CNNS-CRF using pre-trained word embeddings obtained a significant classification improvement over the one using randomized initialization. For different pre-trained embeddings, Google's BERT 300 dimensional embeddings achieved the best results on the NER task. However, Word2Vec's performance was not as good as GloVe and BERT. There are two reasons which can explain its poor performance. First, Word2Vec rarely utilizes the statistics of the corpus since they train on local context windows instead of on global co-occurrence counts [20]. Second, Word2Vec embeddings are trained in a case-sensitive manner, excluding many common symbols such as punctuations and digits, resulting in vocabulary mismatch [20].

Table 5. Comparative evaluation between four algorithms for distributed embeddings.

Embedding	Vector Dimension	Avg. F-1 Measures
Random	100	0.746
Word2Vec	300	0.889
GloVe	300	0.898
BERT	300	0.903

6. CONCLUSIONS

In summary, we systematically studied the resume information extraction based on the latest techniques in NLP. Most prevalent resume parsers use regular expression or fuzzy keyword matching to segment the resumes into consecutive text blocks. Based on each text block, various machine learning classifiers like SVM and naïve Baye s perform resume facts identification. In our study, we proposed a novel end-to-end pipeline for resume information extraction based on distributed embeddings and neural networks-based classifiers. This pipeline dispenses the time-consuming process of constructing various hand-crafted features manually. The second contribution we made is a new approach for text block segmentation. This approach incorporates both position-wise line information and integrated meanings within each text block. Compared to hand-crafted features, the automatic feature extraction by neural networks can better capture the subtle semantic features for delimiting the text blocks. Quantitative comparison between five proposed text classifiers suggested that Attention BLSTM was effective in text block classification and robust over both short and long sentences. Comparative evaluation between four publicly published resume parsers confirmed the superiority of our text block classification algorithm. We believe that the iterative contextual line classification can further improve the independent line classification performed by coordination between line type classifier and line label classifier. For resume facts identification, we quantitively compared four kinds of sequence labelling classifiers. Experimental data indicated that BLSTM-CNNs-CRF was effective in performing named entity recognition task. Based on our proposed resume information extraction method, we developed an online resume parser. This system runs well in real condition. Most neural networks-based sequence labelling classifiers require extra engineered features to augment their sequence labelling performance except for BLSTM-CNNs-CRF. We performed the ablation study to verify that the CNN layer of BLSTM-CNNs-CRF was effective in extracting text features. CNNs are useful in capturing the morphological information of words simulating the process of designing various text features manually. Besides, comparative evaluation of different word embeddings suggested that the word representations were essential for named entity recognition task. For future study, we expect to enrich the functions of our online resume parser by incorporating the ontology concept. Through constructing the ontology for each person, we hope to develop a talent recommendation system.

REFERENCES

[1] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu (2016) “Attention-based Bidirectional Long Short-term Memory Networks for Relation Classification”, In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL’16), Berlin, Germany, August 7-12, 2016, pp 207-212.

[2] Xuezhe Ma, & Eduard Hovy (2016) “End-to-End Sequence Labelling via Bi-directional LSTM-CNNs-CRF”, In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL’16), Berlin, Germany, August 7-12, 2016, pp 1064-1074.

[3] Kun Yu, Gang Guan, and Ming Zhou (2005) “Resume Information Extraction with Cascaded Hybrid Model” In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), Stroudsburg, PA, USA, June 2005, pp 499-506.

[4] Jie Chen, Chunxia Zhang, and Zhendong Niu (2018) “A Two-Step Resume Information Extraction Algorithm” Mathematical Problems in Engineering pp1-8.

[5] Jie Chen, Zhendong Niu, and Hongping Fu (2015) “A Novel Knowledge Extraction Framework for Resumes Based on Text Classifier” In: Dong X., Yu X., Li J., Sun Y. (eds) Web-Age Information Management (WAIM 2015) Lecture Notes in Computer Science, Vol. 9098, Springer, Cham.

[6] Hui Han, C. Lee Giles, Eren Manavoglu, HongYuan Zha (2003) “Automatic Document Metadata Extraction using Support Vector Machine” In Proceedings of the 2003 Joint Conference on Digital Libraries, Houston, TX, USA, pp 37-48.

[7] David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft (2003) “Table Extraction Using Conditional Random Field” In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, Toronto, Canada, pp 235- 242.

[8] Amit Singh, Catherine Rose, Karthik Visweswariah, Enara Vijil, and Nandakishore Kambhatla (2010) “PROSPECT: A system for screening candidates for recruitment” In Proceedings of the 19th ACM international conference on Information and knowledge management, (CIKM’10), Toronto, ON, Canada, October 2010, pp 659-668.

[9] Anjo Anjewierden (2001) "AIDAS: Incremental Logical Structure Discovery in PDF Documents" In Proceedings of 6th International Conference on Document Analysis and Recognition (ICDAR'01) pp 374-378.

[10] Sumit Maheshwari, Abhishek Sainani, and P. Krishna Reddy (2010) "An Approach to Extract Special Skills to Improve the Performance of Resume Selection" Databases in Networked Information Systems, Vol. 5999 of Lecture Notes in Computer Science, Springer, Berlin, Germany, 2010, pp 256-273.

[11] Xiangwen Ji, Jianping Zeng, Shiyong Zhang, Chenrong Wu (2010) "Tag tree template for Web information and schema extraction" Expert Systems with Applications Vol. 37, No.12, pp 8492-8498.

[12] V. Senthil Kumaran and A. Sankar (2013) "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT)" International Journal of Metadata, Semantics and Ontologies, Vol. 8, No. 1, pp 56-64.

[13] Fabio Ciravegna (2001) "(LP)2, an Adaptive Algorithm for Information Extraction from Web-related Texts" In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining. Seattle, WA.

[14] Fabio Ciravegna, and Alberto Lavelli (2004) "LearningPinocchio: adaptive information extraction for real world applications" Journal of Natural Language Engineering Vol. 10, No. 2, pp145- 165.

[15] Yan Wentan, and Qiao Yupeng (2017) "Chinese resume information extraction based on semi-structure text" In 36th Chinese Control Conference (CCC), Dalian, China.

[16] Zhang Chuang, Wu Ming, Li Chun Guang, Xiao Bo, and Lin Zhi-qing (2009) "Resume Parser: Semi-structured Chinese document analysis" In Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, USA, Vol. 5 pp 12-16.

[17] Zhixiang Jiang, Chuang Zhang, Bo Xiao, and Zhiqing Lin (2009) "Research and Implementation of Intelligent Chinese resume Parsing" In 2009 WRI International Conference on Communications and Mobile Computing, Yunan, China, Vol. 3 pp 588-593.

[18] Duygu Çelik, Askýn Karakas, Gülsen Bal , Cem Gültunca , Atilla Elçi , Basak Buluz, and Murat Can Alevli (2013) "Towards an Information Extraction System based on Ontology to Match Resumes and Jobs" In Proceedings of the 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops, Japan, pp 333-338.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013) "Efficient Estimation of Word Representations in Vector Space" Computer Science, arXiv preprint arxiv:1301.3781.

[20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning (2014) "GloVe: Global Vectors for Word Representation" In Empirical Methods in Natural Language Processing (EMNLP) pp 1532-1543.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019) "BERT: Pre- training of Deep Bidirectional Transformers for Language Understanding" arxiv:1810.04805.

[22] Yoon Kim (2014) "Convolutional Neural Networks for Sentence Classification" In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) pp 1746-1751.

[23] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao (2005) "Recurrent Convolutional Neural Networks for Text Classification" In Proceedings of Conference of the Association for the Advancement of Artificial Intelligence Vol. 333 pp 2267-2273.

[24] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow (2017) "Adversarial Training Methods for Semi-supervised Text Classification" In ICLR 2017.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017) "Attention Is All You Need" In 31st Conference on Neural Information Processing Systems (NIPS' 2017), Long Beach, CA, USA.

[26] Zoubin Ghahramani, and Michael I. Jordan (1997) "Factorial Hidden Markov Model" Machine Learning Vol. 29 No. 2-3, pp 245-273.

[27] Andrew McCallum, Dayne Freitag, and Fernando Pereira (2000) "Maximum Entropy Markov Models for Information Extraction and Segmentation" In Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00) pp 591-598.

[28] John Lafferty, Andrew McCallum, and Fernando Pereira (2001) "Conditional Random Fields: Probabilistic Models for Segmenting and Labelling Sequence Data" In Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01) Vol. 3 No. 2, pp 282-289.

[29] Zhiheng Huang, Wei Xu, and Kai Yu (2015) "Bidirectional LSTM-CRF Models for Sequence Tagging" arXiv preprint arXiv:1508.01991, 2015.

[30] Zhenyu Jiao, Shuqi Sun, and Ke Sun (2018) “Chinese Lexical Analysis with Deep Bi-GRU-CRF Network” arXiv preprint arXiv:1807.01882.

[31] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum (2017) “Fast and Accurate Entity Recognition with Iterated Dilated Convolutions” In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing arXiv preprint arXiv: 1702.02098.

[32] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio (2014) “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling” arXiv preprint aeXiv:1412.3555, 2014.

[33] Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di (2016) “Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition” International Conference on Computer Processing of Oriental Languages Springer International Publishing pp 239-250.

[34] Sanyal, S., Hazra, S., Adhikary, S., & Ghosh, N. (2017) “Resume Parser with Natural Language Processing” International Journal of Engineering Science, 4484.