Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing

Ayishathahira C H[1] [2]
Calpine Labs[1]
UVJ Technologies, Kochi, India[1]
Dept. of CSE[2]
Govt. Engg. College, Palakkad, India[2]
ayishathahira007@gmail.com

Sreejith C
Calpine Labs
UVJ Technologies, Kochi, India
sreejith.cherikkallil@calpinetech.com

Raseek C
Dept. of CSE
Govt. Engg. College, Palakkad, India
raseek.c@gmail.com

***Abstract*—Resume parsing is a technique to extract useful information from resumes for further processing such as resume ranking and selection. Different companies process thousands of resumes during their recruitment process using traditional methods like manual processing and by providing unique resume templates to applicants. The current job recruitment horizon demands better approaches for efficient resume parsing technologies and methods. Even though there are many elementary techniques for parsing the structured documents, they are not suitable for parsing unstructured documents like resumes. The ongoing approaches for resume parsing mainly use regular expressions, chunking, keyword based models and entity recognition models. Relevant to this context, this paper proposes a system for resume parsing using deep learning models such as the convolutional neural network (CNN), Bi-LSTM (Bidirectional Long Short-Term Memory) and Conditional Random Field (CRF). CNN Model is used for classifying different segments in a resume. CRF and Bi-LSTM-CNN models were used for sequence labeling inorder to tag different entities. Pre-trained Glove model is used for word embedding. The proposed system could classify a resume into three segments and extract 23 fields.**

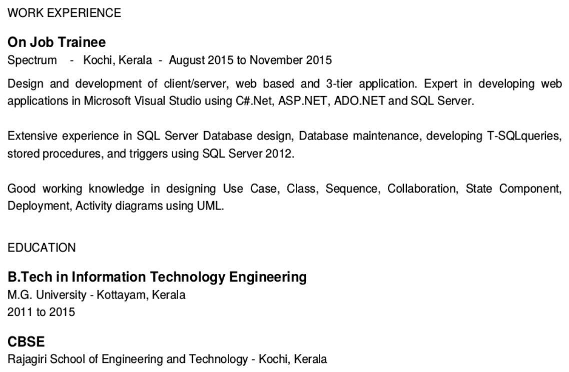
***Index Terms*—Bidirectional long short-term memory, Conditional random field, Convolutional neural network, Deep learning, Glove, Neural network, Segmentation, Word embedding.**

I. Introduction

Document parsing is one of the major steps in information extraction process. There are many approaches [1], [2], [3], [4], [5] to parse a document to get the structured format from its contents. The major challenge in doing document parsing is some of the documents are semi-structured or purely unstructured. Resumes belong to this category of documents. The contents of a resume may be in the form of short texts or tables. If a document is converted into a structured format, it will be comparatively easy to extract various information.

When a computer reads a resume, it simply understands it as a set of numbers, characters, symbols, and punctuations. Hence, we need to build a model that analyzes a resume syntactically as well as semantically. Fig. 1. shows a sample resume section.

One of the major problem faced by the model is the ambiguity of English language. For example, MD is used for representing the term Medical Doctor and also for Managing Director. Another problem is that languages vary infinitely. A date is written as 12th January 2018 or 12-01-18 or in many other formats. Our model resolves all these ambiguities by using Natural Language Processing (NLP) techniques.



WORK EXPERIENCE

On Job Trainee
Spectrum - Kochi, Kerala - August 2015 to November 2015

Design and development of client/server, web based and 3-tier application. Expert in developing web applications in Microsoft Visual Studio using C#.Net, ASP.NET, ADO.NET and SQL Server.

Extensive experience in SQL Server Database design, Database maintenance, developing T-SQLqueries, stored procedures, and triggers using SQL Server 2012.

Good working knowledge in designing Use Case, Class, Sequence, Collaboration, State Component, Deployment, Activity diagrams using UML.

EDUCATION

B.Tech in Information Technology Engineering
M.G. University - Kottayam, Kerala
2011 to 2015

CBSE
Rajagiri School of Engineering and Technology - Kochi, Kerala

Fig. 1. Example of a resume section.

Large organizations deal with thousands of resumes during their recruitment process. Hence, some effective methodology is needed to parse this huge amount of resumes. The resumes may be in different formats like .pdf, .docx, .doc, .odt, etc. and this again may be a critical issue. The conversion of these format types into text format is a very intricate task.

Job recruitment process is an important application of resume parsing. There are many software tools available for automatic job recruitment process. Rchilli [6] and Sovren [7] are the examples of software that performs resume parsing for automatic job recruitment. This software packages will also provide a ranking of resumes based on the parsed content. The output will be in a .json or .xml file format. They use many methods such as pattern matching, machine learning, spatial analysis, fuzzy logic, etc. and resume parsing is the backbone of these kinds of software. Recruiters extract candidates work experience in each company, their gap period and total work experience, etc. by using resume parsing technology.

Fig. 2. is the basic architecture of our model: Extracting Plain Text, Preprocessing, Segmentation and Information Extraction. All these steps are explained in detail in the system design section.

978-1-5386-4966-4/18/$31.00 ©2018 IEEE

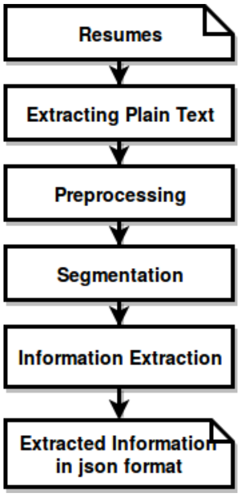


Fig. 2. Basic Architecure of Resume Parser.

This paper is organized as follows: Section II discusses various approaches for resume parsing and named entity recognition. Section III discusses the architecture of our model and next section discusses the experimental evaluation of the proposed method. Section V gives a brief conclusion and describes the future scope of this work.

II. Related Work

This section discusses several existing methodologies for resume parsing, some state of the art deep learning concepts for Named Entity Recognition (NER) and sentence classification.

There are few papers that discusses resume parsing. Sadiq et al [8], proposed a system to extract unstructured information from resumes and convert it into the structured format and do the ranking, based on the retrieved information. Their system architecture divided into 2 groups such as (1) Outer World System, (2) Resume Ranking System. The Outer World System consists of candidates C.Vs and their Social Profiles like Linkedin, Github, etc. The Resume Ranking System has 3 subparts: Parser System, Candidate Skill-set Database and Resume Ranking algorithm. Parser system is the important subpart of the Resume Ranking System. It includes lexical analysis, syntactic analysis, and semantic analysis. At first, the resume is segmented into sections like personal, educational, job-related, etc. by using a data dictionary. This dictionary contains all possible headings in a resume. The system will then search for the heading that matches with any of the headings in the dictionary. When a match is found, then the contents between the matched headings and the next heading are extracted. These segments have a group of chunkers to find named entities from each segment. It will reduce the systems complexity. In the syntactic analysis, a parse tree is generated that will give a structure to the sentence. Semantic analysis deals with the meaning of the sentence. For example, a sentence in a resume is University of ABC. So, it should convert into ABC University and it stores as a JSON file format.

The following two papers discuss some different approaches to resume parsing such as an ontology-based model proposed by Celik D et al [11], [13] and a cascaded hybrid model by Yu K et al [9]. An ontology-based model [11] converts a resume into an ontological structural model. The dataset they used was 205 Turkish resumes. The proposed system works based on Semantic Web approach. First, the system converts the resumes in pdf formats into an HTML format and then remove all HTML tags. Then applies sentence end algorithm which can add some tags, that indicates an end of a sentence and end of a paragraph. After that, split sentences into words and compare every word with the words in ontology knowledge base, to find the category of each sentence. By using some defined rules, extracts information from the sentences. In cascaded hybrid model [9], the model splits a resume into segments by using a Hidden Markov Model (HMM). This paper only concentrated on segmentation, personal information extraction, and educational information extraction. SVM multi-class classification algorithm is used for personal information extraction because of the words in personal information block are independent. The features used in SVM algorithm are the words and their named entities such as location and person. Extracting the educational details from the educational information block is by using HMM algorithm.

Chuang et al [10] introduced a resume parser for Chinese resume analysis. They implemented this parser in a Chinese recruitment website. The parser system has mainly 3 parts such as a Segmentation Module, Identification Module and Feedback Module. The segmentation module determines the text class set, which is divided into two as simple items and complex items. The second step is items identification process, which is an iterative process to get all the items from that resume. Feedback algorithm controls the iteration process by adjusting some parameters until all items are extracted. They used regular expressions for further information extraction. They have taken 5000 resumes for implementation and got 87% accuracy for simple item information extraction and 81% for complex items. The summary of the existing methedologies for document parsing is shown in Table I.

There are many works that have been done in the field of named entity recognition using deep learning concepts. An LSTM-CRF model is introduced by Lample et al [14]. They compared this model with Stack LSTM (S-LSTM) model, it is an LSTM network in a stacked format in which the hidden layers are on top of each other. The detailed description of S-LSTM is discussed in Dyer et al [15]. A bidirectional LSTM with a conditional random field layer outperforms the Stack LSTM. In LSTM-CRF model, first the sentence is split into words and then these words are embedded into d-dimensional vectors. A word representation is produced by using a forward LSTM in which it takes the left context of the sentence at first. Another representation is formed by considering the right context of that word in that sentence. This is done by using a backward LSTM. Hence, the LSTM is called as bidirectional LSTM. After the formation of two type of word representation, it will be concatenated to one. This combined representation is

TABLE I
OVERVIEW OF THE COMPARED METHODOLOGIES AND THE DATASETS USED.

Type of Approach	Model	Dataset	Performance
Combination of HMM and pattern matching approach	Almgren et al (2000) [2]	95 tagged seminar announcements	80%
Cascaded model	Yu et al (2005) [9]	1,200 Chinese resumes	80.44% F-score for personal information and 73.40% for educational information
Pattern matching model	Chuang et al (2009) [10]	5,000 resumes	84%
Deep learning based model	Ronan Collobert (2011) [3]	Penn Treebank	82.8%
Ontology based model	Celik et al (2013) [11]	205 Turkish resumes	80%
Rule based model	Sadiq et al (2016) [8]	Resumes and Candidates Social Profiles	Not specified
Pattern matching model	Anujna M and Ushadevi A (2017) [12]	Hospital staff members data	Not specified

given to the CRF layer for each word. It recognizes the right named entity of each word by including the IOBES (Inside, Outside, Beginning, End, Start) property.

A sentence classification using convolutional neural network (CNN) is proposed by Yang et al [16]. They introduce Hierarchical Attention Network (HAN) which is designed for the representation of relevant words and sentences in a document by considering the context. In HAN, sentences are taken as word by word and annotated using bidirectional GRU (Gated Recurrent Unit). In word attention step, relevant meaningful words are extracted from a sentence. Then aggregating all the extracted words representations to form corresponding sentence representation. Extraction of relevant words from each sentence is done by checking similarity between each word annotation and a word level context vector. This context vector is added randomly to the initial process and then jointly learned during the training process. The same process will occur for document vector creation. Here sentence vectors are used instead of word vectors. They compared their method with many approaches like word based CNN, character-based CNN, a linear classifier based on multinomial logistic regression and SVM and they got 90% accuracy.

A pattern matching model for unstructured document parsing is implemented by Anujna and Ushadevi [12]. They created a structured excel sheet from unstructured documents by using regular expression concept. Mainly, the system extracts name, email id, date of birth and phone number. Belthangadi staff members data is used as the dataset for information extraction. Visual studio as front end and SQL database as the backend are considered for system implementation. In the beginning, the system preprocesses the data by removing stopwords and perform stemming. Porter stemmer is used for the stemming process. Then the preprocessed data is given to the regular expression module.

Jan Luts [17] introduced a Convolutional neural network (CNN) for predicting job title for job descriptions. A job description is a description of a job in one or two sentences like, ”*we are urgently looking for a developer with a C++ background with a ...*”. Here, the job title for that description is a C++ developer. Word2vec is used for word embeddings of job title and job description words. The dataset they used for both training and testing was the job vacancy list for the Canada, UK, New Zealand and Australia. After data cleaning, the dataset contained 10 million job vacancies, and the unique job titles in the dataset were around 90,000.

To sum up, this section discussed some research works about segmentation, information extraction, and named entity extraction related to resume parsing. There have been less works in the field of resume parsing because of it’s purely unstructured format. Since the current methodologies do not give an efficient parser, we introduce a new parser which extracts all possible information from a resume using some deep learning concepts.

III. System Design

The proposed system has four important tasks such as Extracting Plain Text, Preprocessing, Segmentation and Information Extraction. The following two sections discuss the dataset used and the detailed architecture of the system.

A. Dataset

There are no publically available datasets for resumes. We tapped some resumes from Calpine Lab’s resume collection. It contains above 2000 resumes in different file types like .doc, .pdf, .docx, .rtf, etc. and in many formats like table format, plain text format, etc.

B. Architecture

A detailed architecture of resume parser is shown in Fig. 3. The sections are as follows :

1) *Extracting Plain Text*: This section will convert the resumes which are in .pdf, .docx, .odt, .doc or .rtf file format into text format. Apache tika server [18] is used for it.
2) *Preprocessing*: After the conversion of pdf/docx/doc/odt/ rtf to text, the output text will contain many unwanted lines, punctuations, bullets, etc. All these are removed by using string replacement method and regular expressions.
3) *Segmentation*: The Segmentation process is an important step in the resume parser because a better segmentation model will return the sentences with the correct label. In this step, the model divides a whole resume into different segments such as personal, educational occupational and other. The segmentation model is created using a Convolutional Neural Network (CNN) architecture, which segments the information contained in the resume data into four different classes as personal, educational,

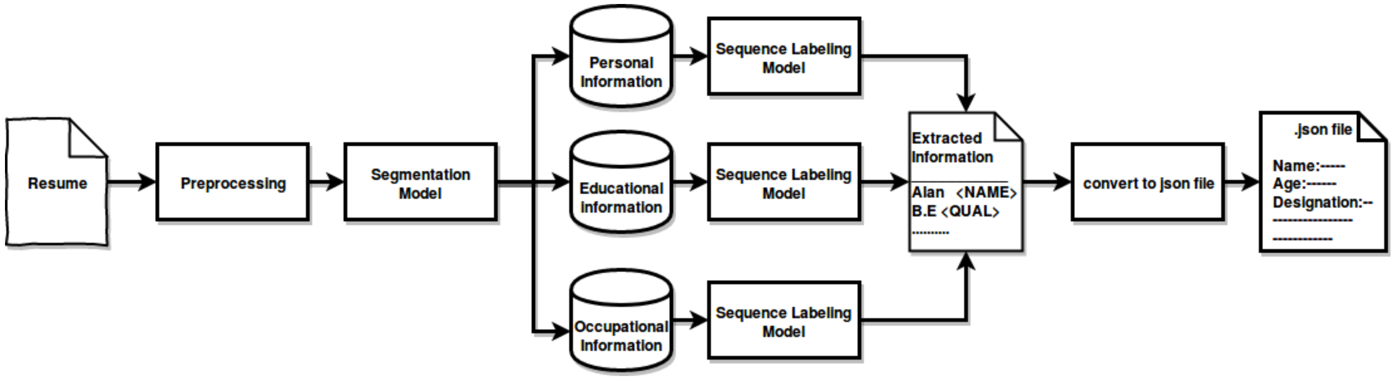


Fig. 3. Detailed System Architecture for Resume Parser

TABLE II
LABELS OF EACH INFORMATION BLOCK

Information Blocks	Labels
Educational	RESULT PLACE INSTITUITION YEAR_OF_STUDY QUALIFICATION UNIVERSITY_OF_GRADUATION
Occupational	DESIGNATION COMPANY_LOCATION ORGANIZATION YEAR_OF_EXPERIENCE TOTAL_EXPERIENCE
Personal	NAME EMAIL CONTACT ADDRESS LOCATION DOB GENDER FATHER_NAME MOTHER_NAME NATIONALITY MARITAL_STATUS PASSPORT_NO

occupational and others. The pre-trained model of Glove [19] is used for the word embedding in CNN.

4) *Information Extraction*: Information extraction is the main task in resume parser, in which they extract useful information from each segment. A CRF based model and a Bi-LSTM-CNN model are created for sequence labeling section. These two models for each segmented block are created and compared. Table II shows the labels used in the three information blocks.

The proposed system reads an input resume which is then converted into text format. This is given to the segmentation model for identifying different sections. The segmented sentences are given to the corresponding sequence labeling models. The labeled information from each model is combined and converted into a JSON file format.

IV. Experimental Setup and Results

This section describes the methods we used to build the two models such as segmentation model and sequence labeling model, and also discusses the comparison of results obtained.

A. *Segmentation Model*

The segmentation model is created using a Convolutional Neural Network (CNN) model and, compared with a Bi-LSTM model.

Fig. 4. shows the CNN model architecture, which contains 16 layers with an input layer, an embedding layer, five convolution layers, five max-pooling layers, a merge layer, one flattening layer and two dense layers. The first layer is the embedding layer, which embeds the words using the pre-trained Glove 100-dimensional word vectors. The output of this layer is given to the first three convolution layers with window sizes 3,4,5 respectively, and the number of filters is 128 for each layer. The features getting from each convolution layers are passed to the corresponding max-pooling layers which have the window size 5.

All feature getting from three pool layers are passed to a merge layer that will concatenate these features and forward to the fourth convolution, which also has 128 convolutions with window size 5, followed by a 1D max-pooling with size 5. The output of this layer is given to the last convolution with max-pooling of size 30. The downsampled features from the last max-pooling layer is given to a flattening layer to get the single structure of feature vectors. Then this final feature vectors are passed to the fully connected layer which has the size 128, and the output of this layer is given to the final dense layer with size four.

The next model is Bi-LSTM model which contains six layers as an input layer, embedding layer, bidirectional lstm layer, dropout layer, flattening layer and dense layer. The purpose of input layer, embedding layer and dense layer are same as CNN model. Bidirectional lstm layer contains 400 lstm units with forward and backward lstm units. Dropout layer is used to reduce overfitting of the model, by removing the neurons whose probability will be less than or equal to dropout rate. The dropout rate used over here is 0.5. We take

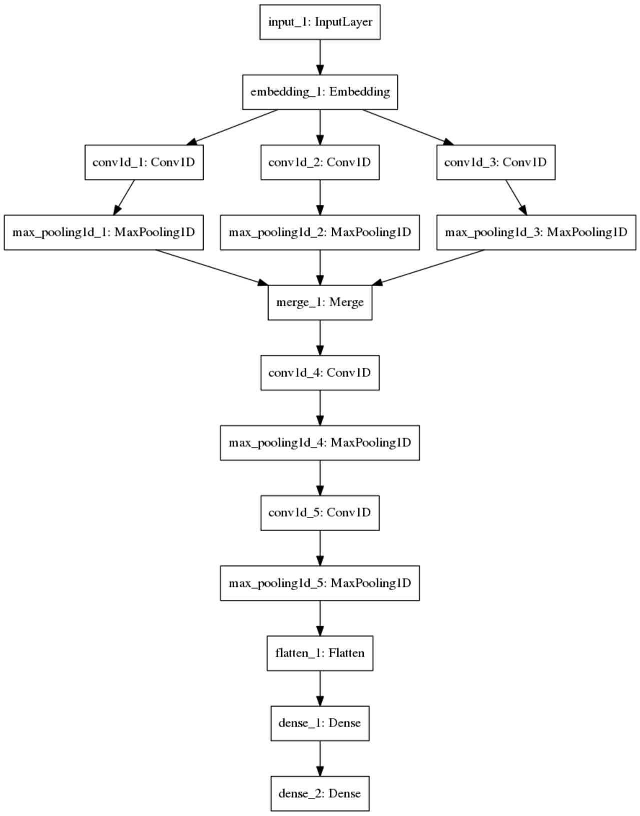


Fig. 4. CNN Model Architecture

TABLE III
RESULT COMPARISON OF SEGMENTATION MODEL

Class	Model	Precision	Recall	F-Score
Education	CNN	0.97	0.92	0.95
	Bi-LSTM	0.38	0.40	0.39
Occupation	CNN	0.78	0.95	0.86
	Bi-LSTM	0.48	0.80	0.60
Personal	CNN	0.93	0.93	0.93
	Bi-LSTM	0.41	0.43	0.42
Others	CNN	0.92	0.77	0.84
	Bi-LSTM	0.59	0.22	0.32

TABLE IV
FEATURES USED IN THE CRF BASED MODEL

	Features
1	Previous word
2	Current word
3	Next word
4	Previous word and current word
5	Current word and next word
6	Pos tag of previous word
7	Pos tag of current word
8	Pos tag of next word
9	Pos tag of current word and next word
10	Pos tag of previous word and current word
11	Pos tag of previous word, current word and next word
12	Bigrams

TABLE V
F-SCORE OF EDUCATIONAL FIELDS WITH DIFFERENT MODELS

Tags	CRF++	Bi-LSTM-CNN
Qualification	0.90	0.76
Institution	0.89	0.73
Place	0.73	0.69
Year of study	0.77	0.74
Result	0.94	0.68
University of graduation	0.90	0.72

precision, recall, and f-measure as the evaluation metrics to check the performance of our system. Table III shows the precision, recall, and f-score obtained for each class labels. We take total 1200 resumes for evaluation, in which 80% for training and 20% for testing, and we got 91% testing accuracy for the CNN model and 43% for the Bi-LSTM model.

B. Sequence Labelling Model

CRF based model and Bi-LSTM-CNN model are used for sequence labeling. Conditional Random Fields (CRF) are the probabilistic graphical model, which is undirected. So we can take features from both directions like past and future features, for the sequence labeling task. There are many tools available for the implementation of CRF based models. CRF++ [20] is one of the popular tools. In CRF++, It automatically iterates during training based on an ”object value”. When the object value converges to a fixed point, it will stop the iteration.

In our experiment, we take 800 resumes in 4:1 ratio. We built three CRF based models for the three information blocks as personal, educational and occupational. In personal information block, the iteration stops in 28 minutes during training and got 94% accuracy. Twenty five minutes is taken for the training part of the occupational information block and got 95% accuracy. Finally, for the educational information, we got 97% accuracy in 22 minutes. Features used in the CRF based model is listed in Table IV.

Chiu and Nichols [21] introduced a Bi-LSTM-CNN model for named entity recognition. Inspired by that model, we build a Bidirectional LSTM-CNN model for sequence labeling. It is a combination of recurrent neural network and convolutional neural network. In our system, we take word embedding, character embedding, and case related features as the input of Bi-LSTM. Here, first, we do character embedding for each word and pass it to a 1D convolution layer, which has a window size 5 and number of filters 40. Then it is forward to the max-pooling layer with window size 52. Then the output of the last drop out layer is concatenated with the corresponding word embeddings and with their case features.

This concatenated values for each word is taken as input to the next LSTM network. The system contains 400 LSTM units in which 200 for forward LSTM and 200 for backward LSTM. The output of the LSTM units is given to the final fully connected layer. We got 70% model accuracy for educational information and 67%, 69% for personal and occupational information blocks respectively. We select rmsprop optimizer

TABLE VI
F-SCORE OF PERSONAL FIELDS WITH DIFFERENT MODELS

Tags	**CRF++**	**Bi-LSTM-CNN**
Name	0.85	0.74
Email	0.75	0.58
Contact	0.85	0.67
Address	0.80	0.70
Gender	1.00	0.69
Marital status	1.00	0.65
Father name	0.79	0.69
Mother name	0.68	0.64
Passport no	0.67	0.60
Nationality	1.00	0.72
Date of birth	0.98	0.75
Location	0.56	0.54

TABLE VII
F-SCORE OF OCCUPATIONAL FIELDS WITH DIFFERENT MODELS

Tags	**CRF++**	**Bi-LSTM-CNN**
Designation	0.92	0.75
Company location	0.80	0.66
Organization	0.85	0.74
Year of experience	0.84	0.70
Total experience	0.78	0.66

and sparse categorical cross-entropy as parameters. Dropout ratio chosen for both CNN and Bi-LSTM models is 0.5.

Tables V, VI and VII shows the performance comparison of different models for information extraction from the three segmented blocks.

V. Conclusion and Future Scope

The resumes or CVs will be written in many file formats. Hence resume parsing is an intricate task in automatic job recruitment tools. In this paper, we used neural networks and CRF to segment and extract various information from resumes. CNN model is used for segmentation and compared with a Bi-LSTM model. A CRF based model is chosen for information extraction and compared with a Bi-LSTM-CNN model. We segmented and extracted several information from personal, educational and occupational blocks. The results are promising and the output JSON file contains 23 data fields.

As a future work, we can enlarge the resume dataset and improve the performance of the proposed system. Also can be extended by including more sections like skills, hobbies, publications, etc.

VI. Acknowledgment

This work was fully funded by Calpine Labs, UVJ Technologies, Kochi, India. We are also immensely grateful to Mr. Bijeesh Devassy, Project Manager, Calpine Labs, for sharing his pearls of wisdom with us during the course of this research. We also thank all of the colleagues from Calpine Labs, UVJ technologies, Kochi, India, who provided insight and expertise that greatly assisted the research.

References

[1] A. Sharma, O. Tuzel, and D. W. Jacobs, “Deep hierarchical parsing for semantic segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 530–538, IEEE, 2015.

[2] M. Almgren and J. Berglund, “Information extraction of seminar information,” *CS224N: Final Project*, pp. 1–12, 2000.

[3] R. Collobert, “Deep learning for efficient discriminative parsing,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 224–232, 2011.

[4] D. Feng, G. Burns, and E. Hovy, “Extracting data records from unstructured biomedical full text,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[5] S. At-Mokhtar, J.-P. Chanod, and C. Roux, “Robustness beyond shallowness: incremental deep parsing,” *Natural Language Engineering*, vol. 8, no. 2-3, 2002.

[6] R. API, “Rchilli resume parser web api reviews and pricing - 2018.” https://www.capterra.com/p/105548/Rchilli-Resume-Parser-Web-API/, 2018.

[7] https://www.sovren.com/, 2018.

[8] G. R. N. M. A. A. P. K. T. M. T. Sayed Zainul Abideen Mohd Sadiq, Juneja Afzal Ayub, “Intelligent hiring with resume parser and ranking using natural language processing and machine learning.”.

[9] K. Yu, G. Guan, and M. Zhou, “Resume information extraction with cascaded hybrid model,” *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL 05*, 2005.

[10] Z. Chuang, W. Ming, L. C. Guang, X. Bo, and L. Zhi-qing, “Resume parser: Semi-structured chinese document analysis,” in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 5, pp. 12–16, IEEE, 2009.

[11] D. Çelik and A. Elçi, “An ontology-based information extraction approach for résumés,” in *Joint International Conference on Pervasive Computing and the Networked World*, pp. 165–179, Springer, 2012.

[12] M. Anujna and A. Ushadevi, “Converting and deploying an unstructured data using pattern matching,” *American Journal of Intelligent Systems*, vol. 7, no. 3, pp. 54–59, 2017.

[13] D. Çelik, A. Karakas, G. Bal, C. Gültunca, A. Elçi, B. Buluz, and M. C. Alevli, “Towards an information extraction system based on ontology to match resumes and jobs,” in *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*, pp. 333–338, IEEE, 2013.

[14] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” *arXiv preprint arXiv:1603.01360*, 2016.

[15] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, “Transition-based dependency parsing with stack long short-term memory,” *arXiv preprint arXiv:1505.08075*, 2015.

[16] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, 2016.

[17] https://www.kdnuggets.com/2017/05/deep-learning-job-descriptions.html, 2018.

[18] https://wiki.apache.org/tika/TikaJAXRS, 2018.

[19] J. Pennington, “Glove: Global vectors for word representation.” https://nlp.stanford.edu/projects/glove/, 2018.

[20] https://taku910.github.io/crfpp/, 2018.

[21] J. P. Chiu and E. Nichols, “Named entity recognition with bidirectional lstm-cnns,” *arXiv preprint arXiv:1511.08308*, 2015.