Extraction of Complex Semantic Relations from Resumes

Sachin Pawar , Devavrat Thosar , Nitin Ramrakhiyani , Girish K. Palshikar , Anindita Sinha , Rajiv Srivastava

TCS Research, Tata Consultancy Services, Pune, India.

{sachin7.p, d.thosar, nitin.ramrakhiyani, gk.palshikar, anindita.sinha2, rajiv.srivastava}@tcs.com

Abstract

In this paper, we discuss a resume information extraction system that extracts important information from resumes such as career, and education details. We model extraction of these details as a problem of extracting *complex* semantic relations where each relation type has the following characteristics: (i) it may have more than 2 entity arguments (*N-ary*), (ii) argument entity mentions may span multiple sentences (*cross-sentence*), and (iii) some argument entity mentions may be absent (*partial mentions*). Extraction of such complex semantic relations is a challenging problem in general and more so for resumes which can have lots of variations in contents and writing style. We propose a novel approach for extraction of such complex relations which is based on sectioning a document into meaningful chunks of consecutive sentences. Each such chunk is expected to capture all the argument entity mentions of a single N-ary cross-sentence relation mention. Moreover, we jointly model the tasks of extraction of entity mentions (as word-level sequence labelling) and identification of sentence chunks corresponding to a single relation mention (as sentence-level sequence labelling). We evaluate our approach on a dataset of 175 resumes and also describe our deployment experience.

1 Introduction

Recruitment is a complex and important process in human resources (HR) management, responsible for attracting, identifying and selecting qualified, suitably experienced, and skilled personnel to meet the business needs of organizations. Resumes play an important role in recruitment. Candidates give extensive details about themselves in their resumes: personal details, education, work history, skills, roles, projects, tasks handled, trainings, certifications, publications, patents, awards, achievements and much more. There is tremendous variation in the structure, contents, and styles of resumes, across languages, countries, functional areas (e.g., engineering, finance, marketing, sales, HR etc.), and industrial domains (e.g, banking, IT, pharma, manufacturing etc.). Resumes are important not only for job sites (e.g., `monster.com`), social networking sites (e.g., `LinkedIn.com`), but also the end employers (e.g., large multinational IT company) which actually recruit people for specific positions. Such organizations often collect large volumes of resumes. Recruitment executives need to perform several tasks that need a detailed analysis of information present in resumes within a repository; e.g., making queries on resume contents, creating a ranked shortlist of candidates for a given job position, identifying experts etc.

Given the practical importance of resumes, it is not surprising that there has been work in applying *information extraction (IE)* techniques to resumes in a repository: [Yu *et al.*, 2005; Singh *et al.*, 2010; Maheshwari *et al.*, 2010; Celik and Elci, 2012; Kumaran and Sankar, 2013; Chen *et al.*, 2015; Chen *et al.*, 2018; Palshikar *et al.*, 2018]. The goal is to extract specific types of information from resumes and store it in a structured repository (e.g., relational tables) for further processing. A well-known formulation of information in documents is in terms of entities and relations. An *entity type* refers to a set of real-world objects, and an *entity mention* refers to a specific instance of that entity type that occurs in a document; e.g., PERSON, ORG, LOCATION, DATE are entity types and `Yann LeCun`, `University of Toronto`, `Paris` and `July 8, 1960` are respective mentions of these entity types. Much of the information in a resume can be thought of as mentions of various entity types: EMPLOYER, DESIGNATION, DEGREE, INSTITUTE etc. Many entity extraction techniques [Palshikar, 2012; Li *et al.*, 2020] have been developed, which can be applied to extract such mentions from resumes.

A *relation type* defines a specific kind of semantic relationship that may hold between two or more entity types and a *relation mention* relates the mentions of corresponding entity types in a document. E.g., the relation BORN_IN may hold between entities of type PERSON and LOCATION. Relations serve to model complex and structured facts. While there has been much work in extracting entity mentions from resumes, there is relatively little work in applying relation extraction techniques [Pawar *et al.*, 2017] to resumes. In this paper, we discuss a deployed resume information extraction system that extracts two key types of semantic relations that occur in resumes which cover education and career details. These are *complex* relations of *arity* greater than 2 (N-ary), some arguments may be missing in a particular relation mention (*partial mention*) and all arguments of a mention need not be in the same sentence (*cross-sentence*). To the best of our knowledge this is the first paper that explores the use of relation extraction techniques for complex N-ary cross-sentence partial relations in resumes. Relations are realized through identification of chunks of consecutive sentences where each chunk leads to extraction of a single relation mention. Moreover, we jointly model extraction of entity mentions and identification of such relation indicating sentence chunks. Here, extraction of entity mentions is realized by word-level sequence labelling similar to traditional Named Entity Recognition (NER). Identification of sentence chunks is realized through sentence-level sequence labelling (where a document is a sequence of sentences). The intuition is that the knowledge of entity mentions in a sentence is useful for predicting the sentence label as well as the knowledge of the sentence label is useful for identifying entity mentions within that sentence. To the best of our knowledge, this is the first attempt where these two tasks of word-level sequence labelling and sentence-level sequence labelling are being modelled jointly.

We extract the following two relation types: (i) CAREER – a ternary relation indicating that the candidate has worked for an Employer with a specific Designation for a specific Duration, (ii) EDU – a 4-ary relation indicating that the candidate has obtained a Degree from an Institute in a specific YearOfPassing and with specific Marks. Table 1 shows a few example relation mentions. Similarly, there are other relation types of interest in resumes of IT professionals such as PROJECT (which captures details of a particular project in terms of entity mentions of types ProjectTitle, Client, Duration, Role, and Skills used in the project) and CERTIFICATION (which captures details of a certification in terms of entity mentions of types CertificationName, CertifyingInstitute, and CertificationYear). However, in this paper, we only focus on the relation types EDU and CAREER which are present in resumes across various domains. We define one or more key entity arguments of a relation type as *pivot* arguments. Any valid relation mention cannot have missing entity mentions corresponding to these *pivot* arguments. Degree and Employer are pivot entity arguments for the mentions of EDU and CAREER, respectively.

Till recently the problem of N-ary cross-sentence relation extraction received a little attention. A few recent approaches [Peng *et al.*, 2017; Mandya *et al.*, 2018; Jia *et al.*, 2019] propose deep learning based techniques for extracting ternary cross-sentence relations, but these are not applicable directly for our relations in resumes because: (i) *partial* relation mentions with missing entity arguments are not considered. This is not applicable for our relations, e.g., for an EDU relation mention, Marks may not be mentioned (2^{nd} example in Table 1) , (ii) Any candidate relation mention is created such that entity mention arguments are spanned within 3 sentences or within a discourse structure like *paragraph*. However, for resumes, entity mention arguments in a relation mention can be spread farther than just 3 sentences. Also, because of arbitrary presence of blank lines, *paragraph* structure can not be defined easily for resumes. Moreover, all of these N-ary cross-sentence relation extraction approaches assume that the gold-standard entity mentions are already available. However, in practice, it is necessary to solve the end-to-end problem where gold-standard entity mentions are not available. Hence, we design a joint model for extracting entity mentions as well as identifying relation mentions through document sections. [Li *et al.*, 2019] used multi-turn question answering for extracting relations from resumes. However, the proposed technique identifies relations from one sentence at a time and hence can not extract cross-sentence relation mentions. Also, their resume text resembles biographical description of people as against any arbitrarily structured resumes in our case.

2 Problem Definition

We define the problem of extracting semantic relations as –

Input: Resume document X^{test}

Output: List of entity mentions and relation mentions extracted from X^{test}

Training regime: n training resumes $\{\langle X_1^{train}, L_{h_1}, L_{v_1}\rangle, \cdots \langle X_n^{train}, L_{h_n}, L_{v_n}\rangle\}$. L_{h_i} are the word-level labels (using BIO encoding, e.g., B-Employer, B-Degree, I-Degree, O) for each word in each sentence in X_i^{train}. L_{v_i} are the sentence-level labels (e.g., B-CAREER, B-EDU, I-EDU, O) for each sentence in X_i^{train} (see Figure 1). The word-level labels capture the information about entity mentions whereas the sentence-level labels capture the information about sentence chunks where each chunk of consecutive sentences covers a single relation mention (tuple).

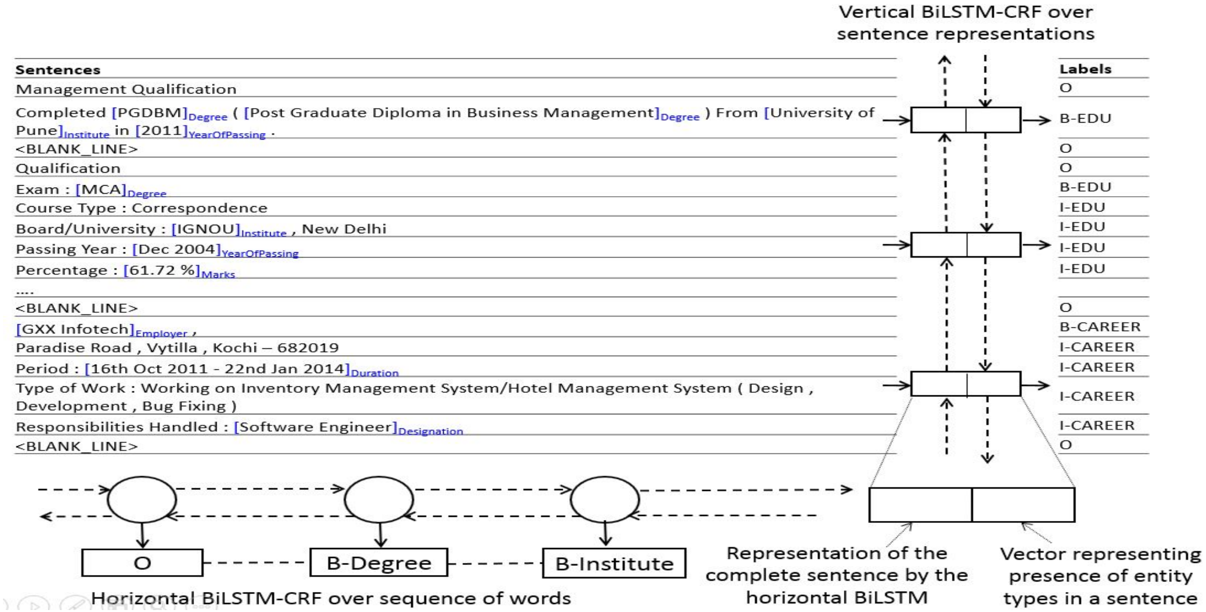
Optional Inputs: We assume that two independent entity extraction techniques are available (described briefly in the next section) - (i) E_{rules} which identifies entity mentions using linguistic rules and gazetteers of known degrees, designations, employers and educational institutes. It does not require any training. (ii) E_{CRF} which is a traditional CRF-based entity extractor based on manually engineered features [Lafferty *et al.*, 2001]. It requires training data having word-level annotations similar to L_{h_i}.

3 Proposed Approach

We propose a document sectioning-based approach for extraction of N-ary and cross-sentence relations. A document is sectioned into meaningful chunks of consecutive sentences such that each chunk is expected to capture complete details about a single N-ary cross-sentence relation mention in terms of the argument entity mentions. Our approach is based on a joint neural model comprising of two sequence labelling layers – i) a horizontal BiLSTM-CRF layer over words in resume sentences (similar to [Huang *et al.*, 2015]), and ii) a vertical BiLSTM-CRF layer over sentences in a resume (Figure 1). Similar hierarchical LSTM based architecures has been reported in the literature for various problems such as aspect-based sentiment analysis [Ruder *et al.*, 2016]. However, to the best of our knowledge, ours is the first attempt which allows predictions of two different sets of labels from two different levels of the hierarchical LSTM structure. The model architecture is described below.

Relation type	Relation mention
EDU	⟨ MCA, 61.72 %, IGNOU, Dec 2004 ⟩
EDU	⟨ Post Graduate Diploma in Business Management, NA, University of Pune, 2011 ⟩
CAREER	⟨ GXX Infotech, 16th Oct 2011 - 22nd Jan 2014, Software Engineer ⟩

Table 1: Examples of relation mentions of the relation types EDU (Degree, Marks, Institute, YearOfPassing) and CAREER (Employer, Duration, Designation), occurring in the resume shown in Figure 1.



Sentences	Labels
Management Qualification	O
Completed [PGDBM]$_{Degree}$ ([Post Graduate Diploma in Business Management]$_{Degree}$) From [University of Pune]$_{Institute}$ in [2011]$_{YearOfPassing}$.	B-EDU
<BLANK_LINE>	O
Qualification	O
Exam : [MCA]$_{Degree}$	B-EDU
Course Type : Correspondence	I-EDU
Board/University : [IGNOU]$_{Institute}$, New Delhi	I-EDU
Passing Year : [Dec 2004]$_{YearOfPassing}$	I-EDU
Percentage : [61.72 %]$_{Marks}$	I-EDU
....	
<BLANK_LINE>	O
[GXX Infotech]$_{Employer}$,	B-CAREER
Paradise Road , Vytilla , Kochi – 682019	I-CAREER
Period : [16th Oct 2011 - 22nd Jan 2014]$_{Duration}$	I-CAREER
Type of Work : Working on Inventory Management System/Hotel Management System (Design , Development , Bug Fixing)	I-CAREER
Responsibilities Handled : [Software Engineer]$_{Designation}$	I-CAREER
<BLANK_LINE>	O

Vertical BiLSTM-CRF over sentence representations
O
B-Degree
B-Institute
Horizontal BiLSTM-CRF over sequence of words
Representation of the complete sentence by the horizontal BiLSTM
Vector representing presence of entity types in a sentence

Figure 1: Overall architecture of our proposed technique. All the layers are not shown for better clarity. The entity mentions and their types are highlighted using blue markers. Information about new lines and blank lines (using a special sentence <BLANK_LINE>) within original resumes is preserved.

Word representation: Each word in each sentence of a resume is represented as a $(d_w + d_p + d_e)$ dimensional vector which is a concatenation of d_w-dim pre-trained word vector (x_w), d_p-dim POS tag embedding of the word (x_p) and d_e-dim NER tag embedding of the word (x_e). Here, spaCy [Honnibal and Montani, 2017] is used for POS and NER information, GloVe word embeddings [Pennington *et al.*, 2014] are used as pre-trained word vectors and POS and NER embeddings are initialized randomly and fine-tuned during training. Hence, overall representation of each word is: $x = [x_w; x_p; x_e]$

Horizontal layers: A complete resume is represented as $X \in \mathbb{R}^{M \times N \times d}$ where $d = d_w + d_p + d_e$ is the overall dimension of word representation, N is maximum number of words in a sentence in a resume and M is number of sentences in the resume. Let $L_h \in \mathbb{N}^{M \times N}$ and $L_v \in \mathbb{N}^{M}$ represent the gold-standard labels for words (for identifying entity mentions) and sentences (for identifying sentence labels), respectively. Each sentence in a resume is passed through a bidirectional LSTM layer.

$$H_w, H_s = BiLSTM_h(X) \tag{1}$$

where $H_w \in \mathbb{R}^{M \times N \times 2d_h}$ and $H_s \in \mathbb{R}^{M \times 2d_h}$ represent context representation for each word and each sentence by the horizontal BiLSTM layer, respectively. Here, d_h is the number of hidden units in each direction of the horizontal BiLSTM layer. Each word representation is then passed through a feed-forward neural network layer.

$$H'_w = FeedForward_h(H_w) \tag{2}$$

where $H'_w \in \mathbb{R}^{M \times N \times n_E}$ and n_E is number of distinct entity labels. We compute two losses for word-level predictions using the gold-standard entity labels – i) negative log likelihood of the whole label sequence predicted by the CRF layer for each sentence, and ii) cross-entropy loss for label predictions at each word in each sentence. Both the losses are normalized based on the number of words in the resume.

$$l^h_{word-seq} = CRF^h_{viterbi}(ReLU(H'_w), L_h) \tag{3}$$

$$l^h_{per-word} = CrossEntropy(Softmax(H'_w), L_h) \tag{4}$$

Vertical layers: For vertical layers, a resume is treated as a sequence of sentences where each sentence is represented by a concatenation of two vectors. The first vector is the sentence representation outputted by the horizontal BiLSTM layer for each sentence - H_s. And the second vector represents the presence of various entity types in the sentence. To capture the entity types which are being predicted for each sentence, we apply max-pooling over its word-level predictions.

$$H_E = MaxPool(Softmax(H'_w)) \tag{5}$$

Here, $H_E \in \mathbb{R}^{M \times n_E}$ will have values close to 1 corresponding to the entity types which are being predicted for at least one word in the sentence. Now, each sentence is represented by concatenating the representation given by the horizontal BiLSTM layer (H_s) and H_E, and then fed into the vertical BiLSTM layer.

$$H'_s = BiLSTM_v([H_s; H_E]) \tag{6}$$

$H'_s \in \mathbb{R}^{M \times 2d_v}$ represents context representation for each sentence as outputted by the vertical BiLSTM layer. Here, d_v is the number of hidden units in each direction of the vertical BiLSTM layer. Each sentence representation is then passed through a feed-forward neural network layer.

$$H''_s = FeedForward_v(H'_s) \tag{7}$$

where $H''_s \in \mathbb{R}^{M \times n_S}$ and n_S is number of distinct sentence labels. We compute two losses for sentence-level predictions using the gold-standard sentence labels – i) negative log likelihood of the whole label sequence predicted by the CRF layer for entire resume, and ii) cross-entropy loss for label predictions at each sentence in the resume. Both the losses are normalized based on the number of sentences in the resume.

$$l^v_{sent-seq} = CRF^v_{viterbi}(ReLU(H''_s), L_v) \tag{8}$$

$$l^v_{per-sent} = CrossEntropy(Softmax(H''_s), L_v) \tag{9}$$

Autoencoder component: Learning informative sentence representations is key for our joint model because these are fed to the vertical BiLSTM layer for predicting appropriate sentence labels. Hence, we also introduce a reconstruction loss using a sequence autoencoder. This autoencoder shares the same horizontal BiLSTM as its encoder layer. Its decoder LSTM accepts the sentence representation outputted by the encoder layer (H_s) as input for each time step and tries to reconstruct the original word representations.

$$X' = LSTM_{decoder}(H^e_s) \tag{10}$$

where $H^e_s \in \mathbb{R}^{M \times N \times 2d_h}$ is an expanded view of H_s (representation of each sentence is copied N times) and $X' \in \mathbb{R}^{M \times N \times d}$ contains the re-constructed word representations as outputted by the decoder LSTM layer. The re-construction loss is computed using Mean Squared Error (MSE) averaged over the number of words in the resume.

$$l_{AE} = MSE(X', X) \tag{11}$$

The total loss for the joint model is sum of all the individual losses.

$$l_{total} = l_{AE} + l^h_{word-seq} + l^h_{per-word} + l^v_{sent-seq} + l^v_{per-sent} \tag{12}$$

Training process: First, only the autoencoder component is trained to optimize l_{AE} on a large corpus of around 241,132 sentences in 2248 resumes, as this does not require any annotated dataset. Then, the horizontal BiLSTM-CRF layers are trained to optimize for $l_{AE} + l^h_{word-seq} + l^h_{per-word}$ on dataset of 1256 resumes which are annotated with only entity labels. Finally, the complete joint model is trained to optimize for l_{total} using 642 resumes dataset which are annotated with both entity as well as sentence labels.

Inference process: During inference, for a new resume X^{test}, it is passed through the joint model and the per-word entity labels are obtained using the horizontal BiLSTM-CRF layer using Viterbi decoding. Also, the per-sentence section labels are obtained using the vertical BiLSTM-CRF layer using Viterbi decoding. Using these sentence-level section labels, the sentence chunks for each relation type are identified. A relation mention of a relation type is formed for each predicted sentence chunk for that type. If no pivot entity mention exists within the chunk (Degree for EDU and Employer for CAREER), no relation mention is identified for that chunk. Otherwise, a relation mention is formed by choosing the highest confidence entity mention for each argument entity type. Here, arguments other than the pivot entity types may be absent within the chunk. Thus, our approach is able to identify partial relation mentions with possibly empty entity mentions for non-pivot arguments. Optionally, in addition to entity mentions identified by the joint model, we also consider the entity extraction output of two entity extraction models - E_{rules} and E_{CRF}. Hence, for forming relation tuples, ensemble output of 3 entity extraction techniques (joint model, E_{rules} and E_{CRF}) is considered.

Pipeline model: We also explore a special case of our joint model which is a *pipeline* model. Here, the horizontal and vertical layers of the joint model are not trained simultaneously but trained in a sequential manner. First, only horizontal layers are trained to learn an entity extractor. Then, only vertical layers are trained to learn a sentence label identifier. For the vertical layer, the input sentence representation is constructed in a similar manner as the joint model. The only difference being that the part of the sentence representation which represents the presence of entity types in a sentence (H_E) is constructed using predicted entity mentions given by the entity extractor realized using the horizontal BiLSTM-CRF layer. Here also, we consider an ensemble with the other two entity extraction techniques - E_{rules} and E_{CRF}.

Rule-based entity extractor E_{rules}: This is a rule-based entity extractor which identifies entity mentions using linguistic rules, gazetteers and combination of both. Here, gazetteers are simply lists of known degrees, designations, employers and educational institutes. The advantage is that no annotated training data is needed, but it requires some domain-expertise for designing linguistic rules and external resources for gazetteers. For constructing gazetteers, in addition to external public resources, we also use a semi-supervised gazette creation algorithm which needs a small set of seed examples and a large unlabelled resumes corpus. These automatically created gazetteers require manual verification to remove incorrect entries. The linguistic rules use certain linguistic properties of entity mentions of certain type and their context to define extraction patterns using regular expressions. Following are some examples of such patterns:

P1 `(Bachelor|Master) of <NP>` : This pattern matches entity mentions of type Degree. Here, `<NP>` matches any base noun phrase[1].

[1] noun phrase which does not contain any other noun phrase

P2 `work(ed|ing) at <NP1> as a <NP2>` : This pattern identifies entity mentions of types Employer and Designation as `<NP1>` and `<NP2>`, respectively.

CRF-based entity extractor E_{CRF}: This is a traditional machine learning model which uses CRF [Lafferty *et al.*, 2001] for sequence labelling based on manually engineered features. It assigns entity type labels for each word in a sentence using BIO encoding (e.g., B-Employer, I-Degree, O) exactly in a similar way as our joint model described earlier. Following are some key features that this model uses for representing any word w in a sentence S:

- w itself, root word (lemma) for w
- previous and next words for w
- Part-of-speech tag of w itself, previous and next words
- word structure of w, i.e., whether w is in upper case, lower case or title case, whether it contains numeric characters, etc.
- word which is parent of w in the dependency tree of S, dependency relation type with the parent word
- words which are children of w in the dependency tree of S
- boolean features indicating if w is the first or last word in S
- any verb which precedes or follows w in S

This is a supervised approach and it needs annotated sentences for training where each word is labelled with an appropriate entity type indicating label.

4 Experimental Analysis

Dataset: We used 2248 resumes for training our sequence auto-encoder, 1256 resumes for training E_{CRF} and pre-training of horizontal BiLSTM-CRF layer in our joint model, and 642 resumes for training our complete joint model for jointly identifying entity mentions and sentence chunks. We evaluated our approach on a dataset of 175 resumes containing 597 and 648 gold-standard relation mentions of EDU and CAREER, respectively.

Evaluation: Any gold-standard relation mention of type r is counted as a true positive if there is a "matching" predicted relation mention of type r, otherwise it is counted as a false negative for type r. Here, two relation mentions are considered to be "matching" only if ALL (strict evaluation) of their corresponding entity mention arguments are matching with at least 80% string similarity between them. All the remaining predicted relation mentions of type r which are not true positives, are counted as false positives for r.

Baseline: We used a rule-based baseline approach for extracting relation mentions of EDU and CAREER. This approach assumes that entity mentions have been already extracted. Here, we use the ensemble of 3 entity extractors - only horizontal BiLSTM-CRF layers of the joint model, E_{rules} and E_{CRF}. This approach starts from an entity mention which is a pivot entity argument for a relation type and then attaches entity mentions of other entity arguments in the vicinity (±4 sentences) to construct a relation mention. However, there are several constraints and exceptions incorporated in this attachment decision. Similar to an expert system, this effort-intensive approach has been developed over time by incorporating several human observations regarding how career and education details are mentioned in resumes.

Results and analysis: Table 2 depicts the performance of our proposed techniques as compared to the baseline, for the test dataset of 175 resumes. Our proposed techniques - joint model and its variant pipeline model, both perform considerably better than the baseline, achieving almost 10% higher macro-F1 score. For EDU, the pipeline model achieves the highest F1-score whereas for CAREER, the joint model achieves the highest F1-score. We plan to develop deeper insights regarding the nature and types of relations for which joint model performs better than the pipeline, as a future work. Table 2 also shows that the two independent entity extraction techniques E_{rules} and E_{CRF} help in improving the F1-score. This highlights the importance of traditional features-based machine learning, linguistic rules and gazetteers as well as their complementary contribution to deep learning based techniques for a real-life domain-specific IE system.

Implementation details: We use 100 dimensional pretrained GloVe word embeddings ($d_w = 100$) and use 20 dimensional POS and NER tags embeddings ($d_p = d_e = 20$). Number of hidden units in each direction of horizontal BiLSTM layer, i.e., d_h is 300. Number of hidden units in each direction of vertical BiLSTM layer, i.e., d_v is 500. We use dropout layers for regularization where dropout with probability 0.1 is applied on input of both horizontal and vertical BiLSTM layers. Similarly, dropout with probability 0.4 is applied on output for both horizontal and vertical layers. We use *Adam* optimizer with the learning rate of 0.001. The joint model is trained in 4 steps - (i) only autoencoder part is trained for 7 epochs for optimizing l_{AE} using batch size of 32 sentences, (ii) only horizontal BiLSTM-CRF layer is trained for 7 epochs for optimizing $l_{AE} + l^h_{word-seq} + l^h_{per-word}$ using batch size of 32 sentences, (iii) only vertical BiLSTM-CRF layer is trained for 5 epochs for optimizing $l^v_{sent-seq} + l^h_{per-sent}$ using batch size of 4 documents (resumes), while keeping horizontal BiLSTM-CRF layer weights frozen, and (iv) finally all the layers are trained jointly for optimizing l_{total} for 5 epochs using batch size of 4 resumes. All the hyperparameters are tuned on a random subset of 20% training instances held out as validation set. Once the best hyperparameters are found, then the complete training dataset is used to train the final joint model.

Entity type	Precision	Recall	F1
Degree	0.874	0.805	0.838
Marks	0.940	0.853	0.894
Institute	0.890	0.827	0.857
YearOfPassing	0.959	0.894	0.925
Employer	0.937	0.813	0.871
Duration	0.922	0.753	0.829
Designation	0.877	0.720	0.791

Table 3: Entity extraction performance on the test dataset for the "Joint model" setting in Table 2

	EDU			CAREER			Overall
	P	**R**	**F1**	**P**	**R**	**F1**	**Macro-F1**
Baseline	0.633	0.566	0.598	0.520	0.511	0.516	0.557
Baseline without E_{rules} and E_{CRF}	0.612	0.514	0.559	0.569	0.444	0.499	0.529
Pipeline model	0.707	0.672	**0.689**	0.673	0.582	0.624	0.657
Pipeline model without E_{rules} and E_{CRF}	0.620	0.533	0.573	0.622	0.478	0.541	0.557
Joint model	0.714	0.656	0.684	0.706	0.585	**0.640**	**0.662**
Joint model without E_{rules}	0.708	0.62	0.661	0.693	0.542	0.608	0.635
Joint model without E_{CRF}	0.709	0.648	0.677	0.695	0.556	0.618	0.648
Joint model without E_{rules} and E_{CRF}	0.648	0.533	0.585	0.641	0.442	0.522	0.554

Table 2: Relation extraction performance on the test dataset of 175 resumes (averaged over 3 runs) (Table 3 shows the F1-scores for individual entity types, corresponding to the "Joint model" setting)

5 Deployment Experience

We discuss three key aspects of our experience in deploying our relation extraction technique: handling drift in content and style of resumes, handling noise in entity extraction and not opting for transformers-based models for encoding sentences. In order to handle the drift, we employ a strategy of iterative deployment based on revised models through active learning. E.g., in the previous deployment cycle, we used the joint model trained on 542 resumes. While this model was in use, we also generated an uncertainty score for each resume which was processed through our system, using uncertainty sampling based active learning [Settles and Craven, 2008]. After a month of deployment, we chose 100 resumes with the highest uncertainty scores from the thousands of resumes which were processed. Human annotators then provided the correct entity and section labels for these resumes and a new joint model was re-trained on 642 resumes. We observed the improvements of 2.2 and 0.9 in F1-scores of EDU and CAREER relation mentions respectively on the same test dataset of 175 resumes. On similar lines, the sequence autoencoder is re-trained by adding the new resumes periodically.

Unlike existing N-ary cross-sentence relation extraction techniques such as [Peng *et al.*, 2017], our techniques do not rely on availability of gold-standard entity mentions. In a real-life scenario, we are restricted to use predicted entity mentions and hence our technique needs to be robust to possible noise in the predicted entity mentions. We observed that our approach is tolerant to some of the errors in entity extraction. E.g., even if an entity mention is a false positive, it does not lead to a false positive relation mention unless the vertical BiLSTM-CRF layer identifies the corresponding sentence as a part of a relation indicating sentence chunk. Hence, we trained the joint model on a dataset where for a subset of resumes *predicted* entity labels are used rather than gold-standard entity labels. This makes the joint model more tolerant towards errors in entity extraction during inference.

We did not opt for transformers-based models like BERT [Devlin *et al.*, 2018] for encoding resume sentences and rather chose to use BiLSTM layer along with pre-trained GloVe word embeddings. The proposed technique is a part of a deployed Resume Information Extraction system where we have to cater to a demand of very fast extraction time for each resume. We found BiLSTM-based sentence encoder to be much better in terms of speed and hardware constraints (our system is deployed on only CPU-based servers). As our joint model needs to keep an entire resume in memory during training and inference, we would need GPUs with large memory and these are not feasible within our cost constraints. Moreover, the language used in resumes is quite different from the text on which BERT is pre-trained. Hence, we would not have obtained much benefit from such transfer learning. We observed this fact when we tried BERT-based entity extraction algorithm for entity extraction which gave us comparable accuracy with respect to existing GloVe-based BiLSTM-CRF entity extraction. Therefore, we preferred to use GloVe-based LSTM sequence autoencoder to specifically learn sentence representations for resume sentences.

6 Ethical Considerations

We understand that the most important ethical consideration for us is data privacy, as the resumes contain sensitive personal, career-related, educational and other information. All the resumes in our training as well as evaluation datasets were annotated by only the employees of our organization, who have signed the necessary Non-disclosure Agreement (NDA) to ensure the privacy of the data mentioned in the resumes. We also did not use any kind of crowd-sourcing for these annotation efforts. Further, the system does not extract any information like gender, religion or mother tongue which may lead to bias of any kind.

7 Conclusions and Future Work

In this paper, we proposed to model education and career details mentioned in resumes in the form of well-defined semantic relations. These relations are complex in nature in the sense that they can be N-ary, cross-sentence and may allow partial relation mentions with empty entity arguments. We proposed a joint neural model based technique for extracting mentions (tuples) of such complex semantic relations, along with entity mentions. Our technique is embedded in a larger resume information extraction system of our organization which is currently in use by several customers. In future, we wish extend our technique for extraction of similar complex semantic relations for domains other than resumes.

References

[Celik and Elci, 2012] D. Celik and A. Elci. An ontology-based information extraction approach for resumes. In *Proc. 7th International Pervasive Computing and the Networked World (ICPCA/SWS'12)*, pages 165–179, 2012.

[Chen *et al.*, 2015] J. Chen, Z. Niu, , and H. Fu. A novel knowledge extraction framework for resumes based on text classifier. In *Proc. 16th International Conference on Web-Age Information Management (LNCS 9098)*, pages 540–543, 2015.

[Chen *et al.*, 2018] Jie Chen, Chunxia Zhang, and Zhendong Niu. A two-step resume information extraction algorithm. *Mathematical Problems in Engineering*, 2018.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Honnibal and Montani, 2017] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *https://spacy.io/*, 2017.

[Huang *et al.*, 2015] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[Jia *et al.*, 2019] Robin Jia, Cliff Wong, and Hoifung Poon. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3693–3704, 2019.

[Kumaran and Sankar, 2013] V. S. Kumaran and A. Sankar. Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT). *International Journal of Metadata, Semantics and Ontologies*, 8(1):56–64, 2013.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[Li *et al.*, 2019] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, 2019.

[Li *et al.*, 2020] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, page 1, 2020.

[Maheshwari *et al.*, 2010] S. Maheshwari, A. Sainani, and P. K. Reddy. An approach to extract special skills to improve the performance of resume selection. In *Databases in Networked Information Systems (LNCS 5999)*, pages 256–273, 2010.

[Mandya *et al.*, 2018] Angrosh Mandya, Danushka Bollegala, Frans Coenen, and Katie Atkinson. Combining long short term memory and convolutional neural network for cross-sentence n-ary relation extraction. *arXiv preprint arXiv:1811.00845*, 2018.

[Palshikar *et al.*, 2018] G.K. Palshikar, R.Srivastava, M. Shah, and S. Pawar. Automatic shortlisting of candidates in recruitment. In *Proc. First Workshop on Professional Search (ProfS2018)*, 2018.

[Palshikar, 2012] G. K. Palshikar. Techniques for named entity recognition: A survey. In S. Bruggemann and C. D'Amato, editors, *Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resources*, pages 191–217. IGI Global, 2012.

[Pawar *et al.*, 2017] Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*, 2017.

[Peng *et al.*, 2017] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[Ruder *et al.*, 2016] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005, 2016.

[Settles and Craven, 2008] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, 2008.

[Singh *et al.*, 2010] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla. Prospect: A system for screening candidates for recruitment. In *Proc. 19th ACM international conference on Information and knowledge management (CIKM'10)*, pages 659–668, 2010.

[Yu *et al.*, 2005] Kun Yu, Gang Guan, and Ming Zhou. Resume information extraction with cascaded hybrid model. In *Proc. 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 499–506, June 2005.