DeepParse: Trainable Postal Address Parser

Nosheen Abid*, Adnan ul Hasan† and Faisal Shafait‡
*‡School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Islamabad, Pakistan
†‡ National Center of Artificial Intelligence (NCAI), Islamabad, Pakistan
Email: *nosheen.abid@seecs.edu.pk, †adnan.ulhasan@gmail.com, ‡faisal.shafait@seecs.edu.pk

***Abstract*—Postal applications are among the first beneficiaries of the advancements in document image processing techniques due to their economic significance. To automate the process of postal services it is necessary to integrate contributions from a majority of image processing domains, from image acquisition and preprocessing to interpretation through symbol, character and word recognition. Lately, machine learning approaches are deployed for postal address processing. Parsing problem has been explored using different techniques, like regular expressions, CRFs, HMMs, Decision Trees and SVMs. These traditional techniques are designed on the assumption that the data is free from OCR errors which decreases the adaptability of the architecture in real-world scenarios. Furthermore, their performance is affected in the presence of non-standardized addresses resulting in intermixing of similar classes. In this paper, we present the first trainable neural network based robust architecture- DeepParse- for postal address parsing that tackles these issues and can be applied to any Name Entity Recognition (NER) problem. The architecture takes the input at different granularity levels: characters, trigram characters and words to extract and learn the features and classify the addresses. The model was trained on a synthetically generated dataset and tested on real-world addresses. DeepParse has also been tested on the NER dataset i.e. CoNLL2003 and gave the result of 90.44% which is on par with the state-of-art technique.**

***Index Terms*—Document Analysis, Computer Vision, Deep Learning, NLP, NER**

I. Introduction

Every day, millions of letters and couriers are delivered worldwide. IPC Global Postal Industry Report 2017 [1] states that 38% of the global industrial revenue is generated by postal service providers. The rapid growth of the e-commerce has significantly strengthened the roots of the courier services. To automate the process of mailing requires reading the destination address and sorting the mailing items accordingly. To overcome the challenge of data acquisition from mailing items, document image processing techniques are being used from decades giving successful, integrated and beneficial results. The advancement in the document recognition techniques has introduced innovative methodologies used in postal address processing, showing a remarkable progress and raising new challenges in the field and encouraging research [2].

Document analysis uses computer vision and pattern recognition techniques to extract and process information from documents obtained from multiple sources. Image processing techniques are used to enhance the images to improve the performance of Optical Character Recognition (OCR). OCR, a sub-domain of computer vision, is used to recognize the text from document images. The digital text is analyzed and classified on the basis of a semantic structure through pattern recognition techniques of Natural Language Processing (NLP). Name Entity Recognition (NER) a subset of NLP, is used to classify nouns by analyzing the pattern of the text.

Automatic sorting of mailing items plays a crucial role in the postal service system. Currently, automated sorting systems in mainland China were designed based on recognizing the postcodes for address detection on the envelops [3] which may not be the sufficient information [4] as postal codes vary in size from place to place. In such a situation, an OCR module is required to recognise the postal address on mailing items. Ideally, the sorting system needs to precisely locate the mailing address in real time [5] [6] and send it to the OCR module. Otherwise, it may result in immediate rejection of the mail. The extracted portion of the image may contain unwanted variation and noise like signatures, stamps etc that create challenges for the OCR engine [7]. Many techniques have been proposed to overcome the hindrances in the OCR. Normalization is one of the important techniques [8] that removes the variation of the data and standardizes it for better results. Watanabe et al. [9] carried out extensive research to show that normalization greatly minimizes the error of recognition. Srihari [10] worked on the removal of underlined text by using "good continuity criterion" but the process was time-consuming and usually worked well only on thin images. Yu and Jain [11] used block adjacency graph restoration to propose a technique for character restoration and line removal. Blumenstein introduced horizontal black-line pixel runs to remove the underlines. Connected component [12] strategy was also proposed to overcome some of the errors that are fatal for OCR. The extensive work in the field of document analysis resulted in the development of efficient OCR APIs. Among these Tesseract OCR [13] is the leading OCR engine in terms of its accuracy.

Many techniques have been used to solve the problem of postal address parsing, like Hidden Markov Models (HMM) [14] [15], Support Vector Machines (SVM) [16], Condition Random Fields (CRF) [17], Naive Bayes Classifier [18] and regular grammar-based models [19]. Among these, rule-based and CRF techniques are widely used. Rule-based techniques are computationally efficient and provide promising results on standardized addresses but are limited in terms of classification and require intense domain knowledge [20]. Standardized addresses follow a set of conventions and peculiarities of a

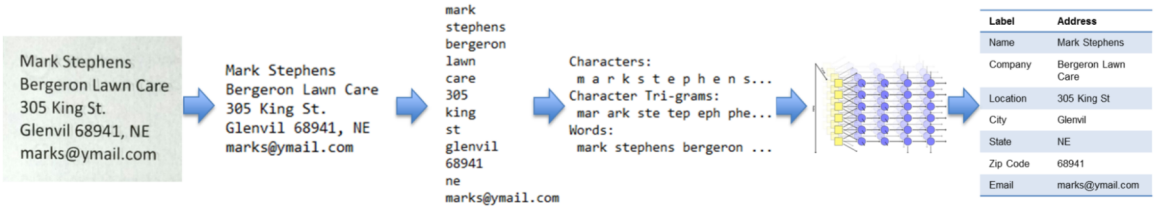

Fig. 1. The input image is fed to the text extractor to get the textual address that is tokenized. The tokenized address is divided into three type of inputs: words, character trigrams and characters which are injected into the classifier for the classification of the address.

specific region that may vary from country to country resulting in the need of a parser that is independent of standardization of address. Generative models like Bayesian Network and HMMs face more difficulties in dealing with rich and complex features than discriminative models like CRFs. CRF, a sequential classifier, has considerable significance in annotation tasks ranging from image segmentation [21] to entity extraction [22]. CRFs provide reasonable results on postal address parsing that requires data cleansing and feature extraction module before CRF classifier. Neural Networks are emerging readily because of its remarkable performance. Sharma et al. [23] have used a fully connected neural network with expensive pre and post-processing modules for postal address parsing. Most of these techniques are designed on the assumption that the data is clean and does not contain any OCR errors which decrease the adaptability of the architecture in real-world scenarios. Furthermore, their performance is affected in the presence of non-standardized addresses resulting in intermixing of similar classes.

A professional tool for postal address parsing, libpostal [24], is designed and trained on Open-StreetMap data and requires address normalization before being fed to the engine. It employs the use of average perceptrons as a classifier popularized by Collins [25]. Nguyen and Guo [26] have done a survey of different structured learning techniques including HMMs, CRFs, Averaged Perceptron (AP), Structured SVMs, Max Margin Markov Networks (M3N), and an integration of search and learning algorithm (SEARN) and concluded that Structured SVMs are better in performance, comparatively. They also introduced their own Structured Learning Ensemble (SLE) by combining state-of-the-art structured learning algorithms. SLE outperformed state-of-the-art structured learning algorithms.

The performance of the traditional techniques gets heavily affected by non-standardized addresses containing OCR errors. In this work, a robust deep learning based architecture has been introduced that takes input at different granularity levels to consider the semantic and structural information of and among the words and tackles these issues of postal address parsing providing effective results.

The main contribution of this paper is a novel trainable neural network-based architecture for non-standardized postal address parsing.

DeepParse architecture is explained in section II. Section III contains the details of the dataset and experiments. Section IV describes the experiments and implementation details. Session V gives the critical analysis of the performance and results of DeepParse. Section VI comprises conclusion and future work.

II. DeepParse – System Description

The proposed architecture, DeepParse, is a neural network based architecture designed to parse the non-standard postal addresses in the presence of OCR errors. Non-standardization destroys the standard sequence of the address. OCR intermixes the characters that look similar to some extent. Further OCR may ignore the bounding characters of the token word or may add the special character instead. The traditional techniques fail to perform well on non-standardized addresses in the presence of OCR errors. RNN has the capability to overcome the stated challenges. The system works in four basic steps:

1) Text extraction is applied to get the textual addresses.
2) Tokenization of textual addresses is done.
3) N-grammar is applied to characters and tokens.
4) Bi-directional LSTMs are trained to classify each field.

Figure 1 describes the pipeline followed by DeepParse. DeepPares can take the document image or string as an input. It is efficient enough to deal with OCR errors. If the input is an image, it will follow the complete pipeline otherwise it will ignore the first step. The textual address is divided into a token of words. Each word is then formatted into three different types of input i.e. lists of characters, trigram characters and words. Each character, trigram character and word is converted to a vector representation using the process of embedding. Lastly, the vectors are injected into the RNN model that will learn the features and classify each category of the address.

Text Extraction Engine: Text extractor divides the input address image into lines, words and the characters to classify each character and generates a digital copy of the text in the image. We have used Tesseract OCR to extract the digital address from the envelope images. Tesseract includes line, baseline and poportional word findings. Line finding involves the recognition of skewed page, baseline fitting involves handling the pages with curved baselines usually encountered in scanning, and proportional word finding involves word recognition by splitting it into characters and analyzing the spaces between them.

Tokenizer: Tokenization is the process of dividing the string into tokens of words on the basis of some pattern. Tokenization is a part of formatting the string to make it suitable as an input to the model. No explicit cleansing of data is required anymore as the designed architecture is efficient enough to deal with noise. After getting the digital postal address, it is divided into tokens on the basis of blank spaces.

Fig. 2. Flow of division of address into tokens and tokens in to character trigrams.

N-grammar: The N-grammar uses the output of tokenizer to form N-grams. Grams can be created with the words or characters. N represents the number of words/characters we are going to using to create a gram. DeepParse's N-grammar unit generates the character trigrams from the addresses. Each formed character trigram is separately treated as a token. In trigram characters, a combination of three characters is chosen in sequence, declaring as a gram. An example in Figure 2 explains the concept of character trigram generated for parsing of postal addresses. These character trigrams are then converted to the vectorized form for further processing.

A. Embeddings

Embedding is a Neural Network that converts the character string to vectorized representation. We used GloVe developed by Pennington et al. [27] an unsupervised learning algorithm for generating vector representations of the words on the basis of the count. DeepParse uses 100 dimension GloVe representations initially. The vectorized representation is used as an input to the classifier.

B. DeepParse

Parsing of digital postal addresses is a sequential problem and has a memory dependency since it has to cater the forward and backward sequence. Hochreiter and Schmidhuber [28] invented LSTMs in 1997. LSTM, a type of RNN, is designed to handle information that needs to be remembered over time. LSTM takes into account the immediately previous state that helps in learning the accurate pattern. A vanilla LSTM preserves the information of the previous state only i.e. information learned at time $t-1$. It is suggested to use bi-directional LSTM (BLSTM) for NER systems because it considers the sequence information in the forward and backward fashion. BLSTM consists of two LSTMs functioning in opposite directions i.e. positive and negative time direction. The output of the forward and backward LSTM is concatenated to form the output for BLSTM.

DeepParse is designed using BLSTM model. The architecture is composed of four BLSTM layers, four dropout layers, two concatenation layers and an activation function. The architecture of DeepParse takes the input in three different formats. It considers a token and formats it for the character, trigram character, and word. Each of these three is passed through the embedding module to generate vector representation. Now it is required to learn the features from different formats of input. The vectors of characters and tri-characters are injected into BLSTM separately to learn their representation. A dropout layer reduces the complexity by ignoring the minute outputs of the nodes. Dropout layer is applied to each BLSTM layer of character and trigram character giving us the most relevant information only. The output of these two layers is concatenated and passed through another BLSTM layer i.e. Projection Layer which projects the output to make it suitable for the next layer with respect to dimensionality. The projected output is concatenated with the embedding of the word and is injected into the final layers that learn the features at word level and performs its classification. Learning of features at different granularity levels of the input helps the models to learn the pattern efficiently and properly classify them. For the graphical representation of detailed DeepParse architecture, see Figure 3.

The positive impact of CRF in literature review has encouraged us to design an architecture that contains deep neural network-based architecture for feature engineering and CRF for classification. All the experiments were conducted on CRF based architecture along with the DeepParse to analyze the performance of both architectures.

The architectures are designed to cater any NER problem on a variable number of classes. The evaluation measures used in this paper are Precision, Recall and F1 Score.

C. Algorithm

The algorithm, given below, explains the working of DeepParse.

III. Dataset

The postal address is confidential information that no courier service provides. So, the best approach in this scenario is to generate synthetic postal address data to train the model. We have modelled the addresses as close as possible to the real world postal addresses. The data is randomized to ensure the non-standardization. A few examples of synthetically generated postal addresses are given in Figure 4.

US postal addresses were simulated. In synthetic data, thirteen different classes of postal addresses were catered i.e. City, Company, Country, Designation, Email, Location, Name , Phone, State, Title, Website, Zipcode, Other. Data scrapping was performed to acquire text files for each class comprising of the corresponding information. The total dataset consists of 18,000 non-standard postal addresses. The training data contains 15,000 addresses, while the validation and test corpus contain 1500 addresses. Table I explains the detailed overview of each corpus: test, validation and train, indicating a number of tokens against each class. The last class is termed as "OTHER"; this prevents the misclassification of categories not present in the current model. The OCR errors were added into the test corpus to analyze the performance of DeepParse in the presence of 22-25% of OCR errors. A few of the OCR errors added in the test corpus are stated below:

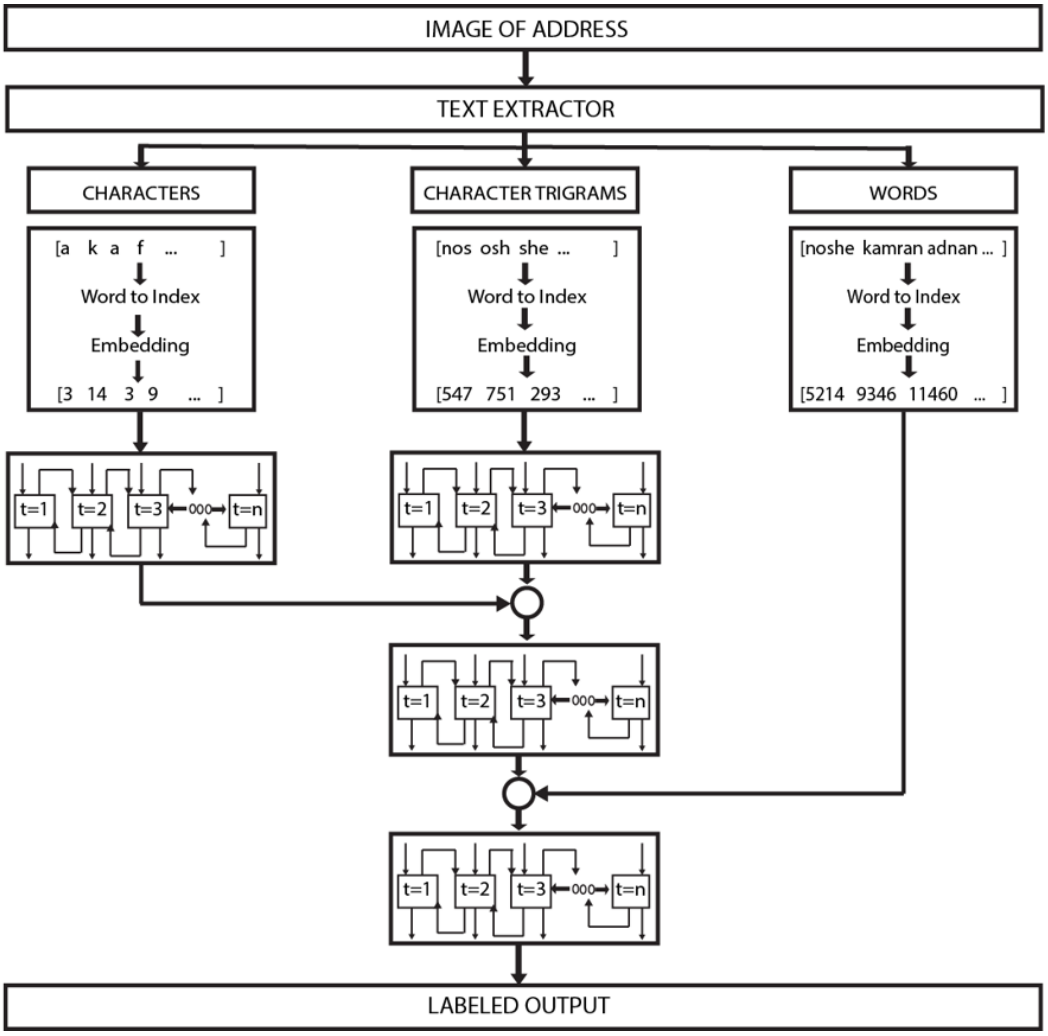

Fig. 3. The textual address is extracted from the input images and divided into words, character trigrams and characters. Indices are generated for each of them and embeddings are formed to get the vectorized form. Vectorized characters and character trigrams are injected into BLSTM separately and their output is concatenated that is further passed through another BLSTM to learn the sequential pattern. Its output is concatenated with vectorized words and fed to another BLSTM layer to learn the necessary representations.

```
Glenvil
Jernigan Turkey Farm
936 Floyd Stevenson, NE
Zip Code 68941-7003
glenvil@gmail.com
```
(a) Standardized

```
Jernigan Turkey Farm
Glenvil
Zip Code 68941-7003
glenvil@gmail.com
936 Floyd Stevenson, NE
```
(b) Non-Standardized

```
Jcrnlgan Turkcy Faarm
Glcnui1
Zip Code 68941-7OO3
glenv!l@gmall.com
936 FI0yd Stcvens0n, NE
```
(c) OCR Errors

Fig. 4. In a few countries the postal addresses are standardized, however, there are countries where a standard address pattern is not followed. For the robustness of DeepParse, the addresses were randomized to ensure non-standardization.

1) I to l or 1 or !
2) i to l or 1 or !
3) O to 0 or C and vice versa
4) o to a
5) e to c and vice versa
6) N to 11 or ll or !! or ii
7) V to U and vice versa
8) B to E or 8 and vice versa
9) S to 2 and vice versa
10) Inserting extra characters, missing few characters, and inserting special characters or replacing characters with special characters, etc.

The DeepParse and CRF model were trained on the synthetically randomized data and tested on three separate datasets. The first test dataset composed of synthetically generated addresses. In the second test dataset, OCR errors were added to the corpus. The third test dataset consisted of approximately 400 real-world addresses, captured from the envelope, that were provided by a courier service provider which is confidential.

DeepParse was further tested on another locally available Name Entity Recognition (NER) dataset, called Conll2003 [29]. This dataset was used to analyze the generalization of the model on other NER problems. The CoNLL2003 dataset has 4 classes, Location, Person, Organization and Miscellaneous,

Algorithm 1 DeepParse's Algorithm

Input: Address Image
Output: Classified Address

```
x ← addressImage
ocrAd ← OCR(x)
tokAddress ← tokenizer(ocrAd)
for address in tokAddress do
    for tok in address do
        tokInd ← genIndex(tok)
        for char in tok do
            charInd ← genIndex(char)
        end for
        for trichar in ngram(tok, 3) do
            tInd ← genIndex(trichar)
        end for
    end for
end for
charEmb ← genEmb(charInd)
triEmb ← genEmb(tInd)
tokEmb ← genEmb(tokInd)
charLSTM ← BLSTM (charEmb)
triCharLSTM ← BLSTM(triEmb)
cIn ← cat(charLSTM, triCharLSTM)
cOut ← BLSTM(cIn)
tokLSTM ← cat(cOut, tokEmb)
deepParseOut ← BLSTM(tokLSTM)
```

TABLE I
WORD LABELS ACROSS THE TRAIN, VALIDATION AND TEST CORPUS OF SIMULATED ADDRESS.

Sr. No	Classes	Train	Validate	Test
1	City	18614	1833	1223
2	Company	12045	1272	790
3	Country	643	95	68
4	Designation	4121	446	268
5	Email	1260	134	85
6	Location	53343	5319	3534
7	Name	27788	2772	1829
8	Phone	2362	233	170
9	State	15078	1519	1004
10	Title	389	34	27
11	Website	955	83	66
12	Zipcode	34625	3441	2270
13	Other	2169	257	131

see Table II. This dataset was designed in 2003 for a competition of ACL that is used as one of the benchmarks for the evaluation of NER architectures since 2003. DeepParse was trained and tested on Conll2003.

TABLE II
WORD LABELS ACROSS THE TRAIN, VALIDATION AND TEST CORPUS OF CONLL2003.

Sr. No	Classes	Train	Validate	Test
1	LOC	7263	1893	1925
2	MICS	3339	898	918
3	ORG	6464	1370	2773
4	PER	6745	1887	8112

A. Data Formatting

The data was annotated in BRAT format [30]. BRAT is a tool used for annotating the dataset that can be later processed and interpreted by the computer easily. It is a structured form of annotation. The format has its defined set of rule and constraints that make it applicable for any text annotation task.

IV. EXPERIMENTS

Experiments were performed on DeepParse and CRF based architecture. It took around 600 seconds per epoch for DeepParse and around 185 seconds per epoch for CRF on a quardcore system. DeepParse converged at 4th epoch whereas CRF architecture converged at 26th epoch on average. Since the architecture of the models was very simple, they were easily trainable on CPUs. Training was performed till last ten epochs until no further improvement in the validation corpus was seen. DeepParse and CRF models were also evaluated on the CoNLL2003 dataset. The training time per epoch for the CoNLL2003 dataset is around 185 seconds.

DeepParse and CRF based architecture was trained and tested on the generated dataset. The trained models were further tested on the generated dataset containing OCR errors and real-world addresses giving it a touch of semi-supervised learning. Furthermore, the models were trained and tested on the ConLL2003 dataset, one of the benchmarks for the evaluation of NER systems. The detailed description of the results and analysis is mentioned in section V.

A. Implementation

Hyper-Parameters of DeepParse were tuned to yield the optimum results on the synthetically generated postal address dataset. The extracted address was segregated into characters, character trigrams and tokens. The dimensions of the corresponding embeddings formed were (x, 25), (x, 25), and (x, 100) respectively, where x represents the number of input tokens. The learning rate was set to 0.0005. The maximum number of epochs was set to 100. The training was stopped when the last ten epochs showed no considerable improvement on the validation corpus. Furthermore, DeepParse uses Adaptive Moment Estimation (Adam) optimization function [31]. Performance evaluation measures were calculated for each class of the postal address to have a detailed analysis of the results.

V. RESULTS

The real world addresses are in the form of captured images of envelopes. To extract the digital address from the images OCR engines is needed. The maturity of data acquisition techniques has provided the facility of OCR APIs. We deployed three different OCR APIs. 1) Tesseract OCR engine [13] was proposed and developed over a span of ten years and provided shining results in terms of accuracy. 2) Contextual LSTMs [32] is an extension of Recurrent Neural Networks (RNNs) designed to segment the text from the images. 3) OCRopy [33] is a collection of programs for document analysis that uses LSTM architecture based on recurrent neural networks. Tafti et

al. [34] have evaluated and did the comparison of state-of-the-art OCR service engines. Tesseract OCR 4.0.0 was performing better than the rest on our postal address images, see Table III. Hence, Tesseract OCR was used for test extraction from the envelopes images.

TABLE III
RESULTS OF DIFFERENT OCR ENGINES.

Sr. No	OCR API	Accuracy %
1	Tesseract	86.00
2	CLSTM	78.00
3	OCRopy	81.00

Since postal addresses contain sensitive information, there is no publicly available dataset of addresses. Because of the unavailability of the dataset, the comparison of the architecture with the proposed techniques become challenging. Even though we have implemented and analyzed the performance of state-of-the-art classifier, CRFs, it is still insufficient to claim the performance of DeepParse. To overcome this drawback, we have used ConLL2003, a freely available dataset for Named-Entity Recognition, being used as one of the benchmarks for the evaluation of NER architectures.

DeepParse was tested on generic NER problem to compare its performance with NER models. Many researchers have worked to solve the classification of the CoNLL2003 dataset. DeepParse performed highly well on the CoNLL2003 dataset giving the F1-Score of 90.44, see Table IV.

DeepPare performance has been compared with multiple methods as stated in table V. People are working on the classification of CoNLL2003 since 2003 till now. It can be seen that DeepParse gave the accuracy of 90.44% that is quite close to state-of-the-art solution.

TABLE IV
DEEPPARSE RESULTS ON CONLL2003 DATASET

Classes	*Precision*	*Recall*	*F1-Score*
LOC	89.52	91.84	90.67
MISC (790)	77.94	80.07	78.99
ORG(68)	87.50	88.58	88.04
PER(268)	90.08	96.36	96.24
AVGl(85)	**89.83**	**91.06**	**90.44**

State-of-the-Art model for the task of postal address parsing is conditional random fields (CRF). Thus, we implemented CRF based architecture for analyzing the performance of CRF along with DeepParse. The evaluated results of DeepParse and CRF based architecture are reported in Table VI and Table VII, respectively.

DeepParse was first evaluated on the synthetically generated dataset. It performed exceptionally well by achieving 100% F1-Score on several classes i.e. Country, Email, Websites, and States. The model performed reasonably well for the rest of the classes too by giving more than 97% of F1-Score. However, CRF architecture was able to perform close to DeepPare but it was expensive in terms of training time. DeepParse took around 30 minutes for training whereas CRF took 75 minutes.

TABLE V
F1-SCORES ON THE TEST SET COMPARING DEEPPARSE WITH OTHER PUBLISHED METHODS FOR CONLL 2003.

Models	CoNLL2003
Curran and Clark (2003) [35]	84.89
Collobert et al. (2011) [36]	88.67
Florian et al. (2003) [37]	88.76
Ando and Zhang (2005) [38]	89.31
Collobert et al. (2011) [36]	89.59
Huang et al. (2015) [39]	90.10
Dernoncourt et al. (2017) [40]	90.50
Turian et al. (2010) [41]	90.80
Passos et al. (2014) [42]	90.90
Luo et al. (2015) [43]	**91.20**
This Work (DeepParse)	90.44

The trained models (DeepParse and CRF based Model) were further tested on the generated dataset containing OCR errors and real-world addresses. Synthetically generated OCR errors did not decrease the performance of the architectures but with a small fraction of less than 0.7%, see Table VI and VII.

The models were then evaluated on real-world addresses containing real-time OCR errors. The performance of the models is affected because the training dataset does not contain OCR errors. Without learning OCR errors the architectures were still able to give F1-Score close to 90%. The performance can be considerably improved if the model is trained for OCR errors.

A. Error Analysis

The results of DeepParse shows that most of the field values are perfectly classified for generated datasets. However, Name class and Company class are intermixing the values to some extent. 59 company names were classified as names and 21 names were classified as company names. On digging down the error, it was seen that most of the company names were on the human names, resulted in the intermixing of the classes. The model has learnt really well if a person name if frequently appeared in the company it will give it a considerable weight for company class. Most of the company names were properly classified but a few company names which were appearing in the names, as well as company names with the same frequency, were intermixed by the model. For example, James Thurmond, Tony Dretch, Kent T Sanders, we as a human as well could not distinguish either it's a company or a person name. When their true label was seen, they were marked as company names. Similarly, words like, Story and Honstein were marked as company names and their true value named. Further, 14 city names were marked as location. On exploring the values in the corpus it was seen that multiple location names were containing the city names as well, meaning city name was part of the location name. Due to which a very small chunk of city names was labelled as location name by the model. Looking at the results of real-time addresses, DeepParse performance was affected on Phone class. It was because it intermixed the Phone class with Postal Code as both classes share similar characteristics. Further, there was

TABLE VI
RESULTS OF DEEPPARSE ARCHITECTURE

Dataset Classes	**Dataset Results**								
	Synthetic Dataset			OCR Error Dataset			Real World Addresses		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
City(1223)	99.42	98.94	99.18	99.19	97.77	98.48	94.62	93.06	93.83
Company (790)	95.62	91.14	93.32	91.53	89.47	90.48	82.41	75.54	78.82
Country(68)	100.00	100.00	100.00	98.51	98.51	98.51	82.41	75.54	78.82
Designation(268)	97.05	98.13	97.59	93.67	96.23	94.93	55.49	80.16	65.58
Email(85)	100.00	100.00	100.00	100.00	95.35	97.62	25.00	100.00	40.00
Location(3534)	99.77	99.83	99.80	99.66	99.69	99.67	95.97	95.53	95.75
Name(1829)	96.79	98.91	97.84	96.63	98.23	97.43	53.41	65.69	58.92
Phone(170)	99.41	98.82	99.12	98.71	96.23	97.45	60.00	8.11	14.29
State(1004)	100.00	100.00	100.00	99.90	99.90	99.90	95.90	98.32	97.10
Title (27)	96.43	100.00	98.18	85.71	75.00	80.00	80.00	80.00	80.00
Website(66)	100.00	100.00	100.00	98.33	100.00	99.16	1.00	1.00	1.00
Zipcode(2270)	99.87	99.78	99.82	99.61	99.83	99.72	95.15	99.39	97.22
micro-avg (11334)	**98.93**	**98.93**	**98.93**	**98.32**	**98.31**	**98.31**	**89.66**	**89.87**	**89.36**

TABLE VII
RESULTS OF CRF BASED ARCHITECTURE

Dataset Classes	**Dataset Results**								
	Synthetic Dataset			OCR Error Dataset			Real World Addresses		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
City(1223)	99.51	98.94	99.22	99.10	99.26	99.18	95.67	91.24	93.40
Company (790)	95.17	89.87	92.45	94.56	90.25	92.36	81.78	72.38	76.79
Country(68)	100.00	100.00	100.00	100.00	100.00	100.00	42.86	100.00	60.00
Designation(268)	98.82	93.66	96.17	98.42	92.91	95.59	80.77	83.33	82.00
Email(85)	100.00	100.00	100.00	100.00	100.00	100.00	33.33	100.00	50.00
Location(3534)	99.72	99.92	99.82	99.72	99.94	99.83	93.26	97.00	95.09
Name(1829)	96.13	99.18	97.63	96.74	98.91	97.81	48.83	68.25	56.93
Phone(170)	100.00	100.00	100.00	100.00	100.00	100.00	88.89	10.81	19.28
State(1004)	100.00	100.00	100.00	100.00	100.00	100.00	97.49	97.90	97.69
Title (27)	96.43	100.00	98.41	96.43	100.00	98.18	66.67	80.00	72.73
Website(66)	100.00	100.00	100.00	100.00	100.00	100.00	1.00	1.00	1.00
Zipcode(2270)	99.96	99.91	99.93	99.96	99.91	99.93	96.53	99.56	98.02
micor-avg (11334)	**98.85**	**98.85**	**98.84**	**98.85**	**98.86**	**98.85**	**89.97**	**89.91**	**89.34**

only one email in the test corpus that was perfectly classified but two company names were classified as email as one of them was containing '@' symbol. For the CoNLL2003 dataset, DeepParse misclassified 'White' as a person and 'House' as an organization. Similarly, 'Shannon', being a location name, was misclassified as a person. Furthermore, 'Africa', 'Dubai' and 'Iran', being miscellaneous in the dataset, were classified as a location by the model. Another example is of 'Honda', categorized as miscellaneous in the dataset, being classified as an organization by the model.

VI. CONCLUSION AND OUTLOOK

Postal address parsing faces the challenge of non-standardization and the intermixing of classes. Previously, standalone machine learning algorithms have been implemented to solve the problem of postal address parsing. Almost all of them either require feature engineering and/or clean data as input. We presented trainable deep learning based approach to postal address parsing that requires no separate feature engineering module and classifies the results. The results show that DeepParse has good generalization on non-standard data and solves the problem of intermixing of classes. DeepParse is also evaluated on CoNLL2003 and the results are on par with the current state-of-the-art techniques. Our results suggest that deep neural networks work well on the problem of postal address parsing and NER and paves the path for further explorations.

Our major goal was to introduce a trainable deep learning based approach to the task of processing of postal addresses. DeepParse is an optimal architecture that takes the images and returns the classified output of the postal address. We deployed the OCR engine to acquire the data from the image and implemented character trigrams along with character and token level embeddings. However, the application of n-gram token embeddings on the parsing of postal addresses can be explored. Furthermore, the stacking of BLSTM layers could also be explored, but we hypothesize that it would affect the overall time complexity of the model.

REFERENCES

[1] International Post Corporation (2017, December). IPC Global Postal Industry Report-Key Findings. Brussels, Belgium. https://www.ipc.be

[2] M. Gilloux. (2014). Document Analysis in Postal Applications and Check Processing. In Handbook of Document Image Processing and Recognition (pp. 705-747). Springer, London.

[3] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri and D. K. Basu (2010). A novel framework for automatic sorting of postal documents with multi-script address blocks. Pattern Recognition, 43(10), 3507-3521.

[4] J. Xue, X. Ding, C. Liu, R. Zhang and W. Qian. (2001). Location and interpretation of destination addresses on handwritten Chinese envelopes. Pattern Recognition Letters, 22(6-7), 639-656.

[5] A. K. Jain and S. K. Bhattacharjee. (1992). Address block location on envelopes using Gabor filters. Pattern recognition, 25(12), 1459-1477.

[6] X. Dong, J. Dong, H. Zhou, J.Sun and D. Tao. (2018). Automatic Chinese Postal Address Block Location Using Proximity Descriptors and Cooperative Profit Random Forests. IEEE Transactions on Industrial Electronics, 65(5), 4401-4412.

[7] A. Rehman, D. Mohammad, G. Sulongand T. Saba. (2009, November). Simple and effective techniques for core-region detection and slant correction in offline script recognition. In Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on (pp. 15-20). IEEE.

[8] A. S. Wanchoo, P. Yadav and A. Anuse. (2016). A Survey on Devanagari Character Recognition for Indian Postal System Automation. International Journal of Applied Engineering Research, 11(6), 4529-4536..

[9] M. Watanabe, Y. Hamamoto, T. Yasuda and S. Tomita. (1997, August). Normalization techniques of handwritten numerals for Gabor filters. In Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on (Vol. 1, pp. 303-307). IEEE.

[10] [5] V. Govindaraju and S. N. Srihari (1992). Separating handwritten text from interfering strokes. From Pixels to Features III: Frontiers in Handwriting Recognition, 17-28.

[11] [6] B. Yu and A. K. Jain. (1996). A generic system for form dropout. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(11), 1127-1134.

[12] Z. Bai and Q. Huo. (2004). Underline detection and removal in a document image using multiple strategies.

[13] R. Smith. (2007, September). An overview of the Tesseract OCR engine. In Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (Vol. 2, pp. 629-633). IEEE.

[14] X. Li, H. Kardes, X. Wang and A. Sun. (2014, November). HMM-based address parsing: efficiently parsing billions of addresses on MapReduce. In Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 433-436). ACM.

[15] R. A. Abbasi. (2005, December). Information Extraction Techniques for Postal Address Standardization. In 9th International Multitopic Conference, IEEE INMIC 2005 (pp. 1-6). IEEE.

[16] C. H. Chang and S. Y. Li. (2010, August). MapMarker: Extraction of postal addresses and associated information for general web pages. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on (Vol. 1, pp. 105-111). IEEE.

[17] C. H. Chang, H. M. Chuang, C. Y. Huang, Y. S. Su and S. Y. Li. (2016). Enhancing POI search on maps via online address extraction and associated information segmentation. Applied Intelligence, 44(3), 539-556.

[18] Y. Zhou, M. Wang, V. Haberland, J. Howroyd, S. Danicic and J. M. Bishop. (2017). Improving record linkage accuracy with hierarchical feature level information and parsed data. New Generation Computing, 35(1), 87-104.

[19] M. Wang, V. Haberland, A. Yeo, A. Martin, J. Howroyd and J. M. Bishop. (2016, December). A probabilistic address parser using conditional random fields and stochastic regular grammar. In Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on (pp. 225-232). IEEE.

[20] T. Churches, P. Christen, K. Lim and J. X. Zhu. (2002). Preparation of name and address data for record linkage using hidden Markov models. BMC Medical Informatics and Decision Making, 2(1), 9.

[21] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang and P. H. Torr. (2015). Conditional random fields as recurrent neural networks. In Proceedings of the IEEE international conference on computer vision (pp. 1529-1537).

[22] F. Peng and A. McCallum. (2006). Information extraction from research papers using conditional random fields. Information processing & management, 42(4), 963-979.

[23] S. Sharma, R. Ratti, I. Arora, A. Solanki and G. Bhatt. (2018, January). Automated Parsing of Geographical Addresses: A Multilayer Feedforward Neural Network Based Approach. In Semantic Computing (ICSC), 2018 IEEE 12th International Conference on (pp. 123-130). IEEE.

[24] O. (2018, May 24). Openvenues/libpostal. Retrieved from https://github.com/openvenues/libpostal

[25] M. Collins. (2002, July). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 1-8). Association for Computational Linguistics.

[26] N. Nguyen and Y. Guo (2007, June). Comparisons of sequence labeling algorithms and extensions. In Proceedings of the 24th international conference on Machine learning (pp. 681-688). ACM.

[27] J. Pennington, R. Socher and C. Manning. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)(pp. 1532-1543).

[28] S. Hochreiter and J. Schmidhuber. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[29] E. F. Tjong Kim Sang and F. De Meulder,(2003, May). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 142-147). Association for Computational Linguistics.

[30] P. Stenetorp, S. Pyysalo, G. Topić,T. Ohta, S. Ananiadou and J. I. Tsujii. (2012, April). BRAT: a web-based tool for NLP-assisted text annotation. In Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (pp. 102-107). Association for Computational Linguistics.

[31] D. P. Kingma and J. Ba. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[32] T. (2017, December 08). Tmbdev/clstm. Retrieved from https://github.com/tmbdev/clstm

[33] T. (n.d.). Tmbdev/ocropy. Retrieved from https://github.com/tmbdev/ocropy/wiki/Jupyter-(IPython)-Notebooks

[34] A. P. Tafti, A. Baghaie, M. Assefi, H. R. Arabnia, Z. Yu and P. Peissig. (2016, December). OCR as a Service: An Experimental Evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym. In International Symposium on Visual Computing (pp. 735-746). Springer, Cham.

[35] J. R. Curran and S. Clark. (2003, May). Language independent NER using a maximum entropy tagger. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 164-167). Association for Computational Linguistics.

[36] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.

[37] R. Florian, A. Ittycheriah, H. Jing and T. Zhang. (2003, May). Named entity recognition through classifier combination. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 168-171). Association for Computational Linguistics.

[38] R. K. Ando and T. Zhang. (2005, June). A high-performance semi-supervised learning method for text chunking. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 1-9). Association for Computational Linguistics.

[39] Z. Huang, W. Xu and K. Yu. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

[40] F. Dernoncourt, J. Y. Lee and P. Szolovits. (2017). NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. arXiv preprint arXiv:1705.05487.

[41] J. Turian, L. Ratinov and Y. Bengio. (2010, July). Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 384-394). Association for Computational Linguistics.

[42] A. Passos, V. Kumar and A. McCallum, (2014). Lexicon infused phrase embeddings for named entity resolution. arXiv preprint arXiv:1404.5367.

[43] G. Luo, X. Huang, C. Y. Lin and Z. Nie. (2015). Joint entity recognition and disambiguation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 879-888).