A CNN based approach to Phrase-Labelling through classification of N-Grams

Chinmay Choudhary
National University of Ireland
Galway, Ireland
c.choudhary1@nuigalway.ie

Colm O'Riordan
National University of Ireland
Galway, Ireland
colm.oriordan@nuigalway.ie

ABSTRACT

Modern approaches address the task of *Phrase-labelling* within any input sentence (eg: NER, Chunking etc.) as a variant of word-tagging problem. These approaches extract the desired phrases as word-sequences which are mapped to specific tag-sequences (with Begin, Intermediate and End tag-types).
However we argue that basic nature of *Phrase-labelling* is not temporal but spatial in nature. Thus we propose and test the hypothesis that a CNN based model that directly extracts labelled n-grams from the input sentence would outperform standard RNN based model.

KEYWORDS

Phrase-labelling, Convolutional Neural Networks, NLP, Information Retrieval

ACM Reference Format:
Chinmay Choudhary and Colm O'Riordan. 2019. A CNN based approach to Phrase-Labelling through classification of N-Grams. In *Forum for Information Retrieval Evaluation (FIRE '19), December 12–15, 2019, Kolkata, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3368567.3368571

1 INTRODUCTION

Some basic NLP/IR tasks such as named entity recognition (NER), chunking, identifying symptoms within a medical report etc. can be referred to as *Phrase Labelling* tasks. These tasks involve the identification of phrases (of variable length) within an input sentence that refer to a single unit belonging to a specific category, and subsequently tagging these phrases with appropriate category labels (for example, the NER task involves identification of phrases within input sentences that refer to individual named entities).
The basic nature of the *Phrase Labelling* problem is slightly distinct from the nature of standard *Word-tagging* NLP problems such as part-of-speech (POS) tagging, semantic role labelling (SRL) etc., as the latter requires labelling strictly at the word-level whereas the former requires labelling at the phrase-level. Further, as these phrases can be of variable length and be located anywhere within sentence space, *Phrase-labeling* task is not fundamentally a Sequence-to-Sequence mapping task (unlike *Word-tagging task*).
Yet modern approaches [23][25][21][26][3] attempt to solve these *Phrase Labelling* problems by reformulating them as *Word-tagging* problems. These approaches extract a desired phrase from the input sentence, as a sequence of words which is mapped to a specific sequence of tags. This specific tag-sequence comprises of *Begin*, *End* and *Intermediate* tag-types (Usually Indexed as B- ,E-, I-)[17][19] indicating first, last and intermediate words of the desired phrase. Modern approaches mostly use various standard Sequence-labelling algorithms such as Hidden Markov Model (HMM) [16], Maximum Entropy Markov Model (MEMM)[1], Conditional Random Field (CRF)[11], RNN [7]/LSTM [9] based models etc. to address a *Phrase Labelling* task.
However we propose following hypothesis about the nature of *Phrase-labeling* problem.

Hypothesis: *Unlike standard Word-tagging problems, the Phrase Labelling problem is not temporal but rather spatial in nature as desired phrases can be of variable length and could be located at any random location within the whole sentence.*

In other words, the nature of *Phrase-labelling* problem is similar to the problem of *Object-recognition* within *Image-processing* domain (identifying specific regions within entire pixel-space).
If the above hypothesis is true, then a *Convolution Neural Network* (CNN) [10] should perform the task of phrase-labelling better than a *Recurrent Neural Network* (RNN) [7].
Thus to test the proposed hypothesis, we propose (as section 2) a simple *CNN based model* that takes in the whole input sentence and directly outputs labelled-phrases. Performance of this simple *CNN based model* is then compared with that of a simple *RNN based* benchmark model through preliminary experiments described in section 6. Section 7 outlines the results obtained.
There are several approaches such as [6][20][5][24] that use CNNs [10] to extract features from input sentence to be fed into sequence-tagger. The work of [22] even attempted to use CNNs to compute a probability distribution of a specific-tag sequence over given word-sequence.
Yet all these approaches eventually address the *Phrase-Labelling* task as sequence-tagging problem. As far as we are aware, this is the first paper that addresses it as a spatial problem. Thus by proving our proposed hypothesis to be true, we would provide a new perspective to the *Phrase-labeling* problem.

2 MODEL

This section describes high-level overview of our proposed CNN based model for *Phrase Labelling*. It is a simplistic model that classifies each n-gram within input-sentence as one of the given categories or not belonging to any of the categories (belonging to

FIRE '19, Dec 12–15, 2019, Kolkata, IN

ACM ISBN 978-1-4503-7750-8/19/12…$15.00
https://doi.org/10.1145/3368567.3368571

category *NONE*).
Let C be the set of category-labels to be assigned to appropriate phrases as the respective *Phrase Labelling* task (for example if task is NER then C=[PER, LOC, ORG, MISC]) and let $s = \{w_1, w_2, ..., w_T\}$ be the input sentence of length T.
The proposed approach would implement following sequential steps on input sentence s.

(1) Extract a set G of all possible n-grams from s with value of n ranging from 1 to T.
(2) For each n-gram $g \in G$ apply following steps.
   (a) Compute the tensor representation X_g of n-gram g. The mechanism for computing such tensor representations (and the various features of g that are encoded within tensor X_g) are described in sections 3 and 4.
   (b) Input tensor Xg to a CNN which outputs the vector $v_g \in R^{|C|+1}$ such that i^{th} value v_{g_i} is associated with i^{th} category $c_i \in C$ and value $v_{g(|C|+1)}$ is associated with the category label *NONE*. The architecture and training procedure of this CNN is described in detail in section 5.
   (c) Apply the *Softmax function* over vector v_g to compute probability vector $p_g \in R^{|C|+1}$ such that p_{g_i} is the probability of g having category label c_i and $p_{g(|C|+1)}$ is the probability that g does not belong to any $c \in C$ (g has category-label *NONE*).
   (d) Assign the category label with the maximum probability to n-gram g.

Thus proposed model will directly extract desired phrases from input sentence s belonging to each category $c \in C$. Figure 1 depicts the overall procedure.

3 FEATURES

This section describes the features of any n-gram g encoded within its tensor representation X_g.
Let $g = w^t, w^{t+1}, \ldots, w^{t+n-1}$ be the n-gram extracted from the sentence $s = w^1, w^2, \ldots, w^t, w^{t+1}, \ldots, w^{t+n-1}, \ldots w^T$ of length T.
The features used to compute the tensor-encoding of g i.e. $X_g \in R^{(n-1+4)\times d\times 3}$ can be classified into two types namely, *N-Gram features* which are derived from the actual n-gram, and *Context features* which are derived from its context within the whole sentence.

3.1 N-Gram features

This section outlines all the features extracted from the actual N-gram g. All these features are embedded as a tensor $NG_g \in R^{(n-1)\times d\times 3}$ which is computed according to equation 1.

$$NG_g = [Word_g; POS_g; SRL_g] \tag{1}$$

Here $Word_g$, POS_g and SRL_g are matrices of dimension $(n-1)\times d$. Concatenating these matrices we create the tensor $NG_g \in R^{(n-1)\times d\times 3}$. The process of computing $Word_g$, POS_g and SRL_g are described in sections 3.1.1, 3.1.2 and 3.1.3 respectively.

3.1.1 Word-Embedding Matrix. This feature matrix is computed by concatenating Word-embeddings of all surface words forming the respective n-gram. Thus, the Word-Embedding feature matrix $Word_g$ of n-gram g is given by equation 2.

$$Word_g = [v^t; v^{t+1}; ...; v^{t+n-1}] \tag{2}$$

Here $v^i \in R^d$ is the Word-embedding vector of word w^i in s, such the $t \le i \le (t+n-1)$. Thus, $Word_g \in R^{(n-1)\times d}$.
For preliminary experimentation, we use Skip-gram Word2Vec embedding [15] but we will experiment with various other Word-embeddings in future work.

3.1.2 Part-Of-Speech Tag Matrix. This feature matrix encodes POS tags of all the words forming the respective n-gram. We perform the POS tagging of the entire sentence s using an nltk tagger.
The POS-Tag feature matrix POS_g of n-gram g is computed by concatenating the encodings of POS-tags achieved by all the words forming g. Thus, POS_g is given by equation 3.

$$POS_g = [tp^t; tp^{t+1}; ...; tp^{t+n-1}] \tag{3}$$

Here $tp^i \in R^d$ is the encoding vector of tag t^i achieved by the word w^i within s, such the $t \le i \le (t+n-1)$. This encoding (as a d-dimensional vector) is computed from one-hot vector indicating POS-tag t_i using a simple auto-encoder. Thus $POS_g \in R^{(n-1)*d}$.

3.1.3 Semantic Role Label Matrix. Similar to POS_g, this feature matrix encodes SRL tags of all the words forming the respective n-gram. We perform the SRL tagging of the entire sentence s using a built-in nltk SRL tagger.
The SRL feature matrix SRL_g of the n-gram g is computed by concatenating the encodings of SRL-tags achieved by all the words forming g. Thus SRL_g is given by equation 4.

$$SRL_g = [ts^t; ts^{t+1}; ...; ts^{t+n-1}] \tag{4}$$

Here $ts^i \in R^d$ is the encoding vector of tag tt^i achieved by word w^i within s, such the $t \le i \le (t+n-1)$. This encoding (as d-dimensional vector) is computed from one-hot vector of tag tt_i using simple auto-encoder. Thus $SRL_g \in R^{(n-1)*d}$.

3.2 Context Features

This section outlines features extracted from the context of N-gram g, within the sentence s.

3.2.1 Adjacent Word Matrix. These features include word-embedding, POS-tag embedding and SRL embedding of words adjacent to n-gram g within the whole sentence s on both sides (i.e. words w_{t-1} and w_{t+n}).
We provide knowledge about adjacent words as exclusive features because adjacent words can drastically influence the classification of n-gram g. For example, if the input sentence consists of the phrase *New York City* then our model should be able to classify the n-gram *New York* as *NONE* (mostly its to be classified as *LOC* though) while the n-gram *New York City* as LOC.
Adjacent word matrices $C_{t-1} \in R^{d\times 3}$ and $C_{t+n} \in R^{d\times 3}$ for words w_{t-1} and w_{t+n} are given by equations 5 and 6 respectively.

$$C_{t-1} = [v^{t-1}; tp^{t-1}; ts^{t-1}] \tag{5}$$

$$C_{t+n} = [v^{t+n}; tp^{t+n}; ts^{t+n}] \tag{6}$$

Here v, tp and ts are word-embeddings, POS-tag encoders and SRL-tag encoders as d-dimensional vectors computed by same processes as for n-gram words.

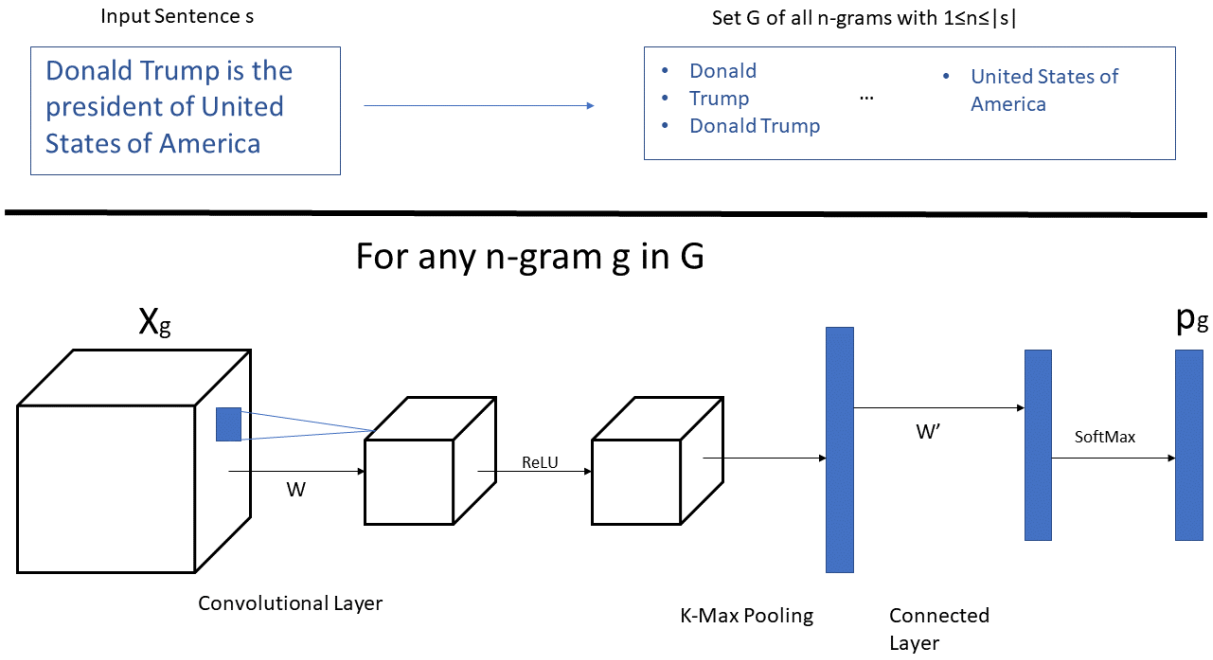

Figure 1: Architecture of our proposed *Phrase-Labelling* model

3.2.2 Context Encoding Matrix. These features include encodings of the entire Word-sequence, its corresponding POS-tag sequence and SRL sequence for both sides of context of n-gram g within whole sentence s excluding adjacent words.
Thus there are two context matrices as $C_{1\to(t-2)}$ and $C_{(t+n+1)\to T}$ representing left and right contexts of g within s. These matrices are given by equations 7 and 8.

$$C_{1\to(t-2)} = [ve^{1\to(t-2)}; tpe^{1\to(t-2)}; tse^{1\to(t-2)}] \quad (7)$$

$$C_{(t+n+1)\to T} = [ve^{(t+n+1)\to T}; tpe^{(t+n+1)\to T}; tse^{(t+n+1)\to T}] \quad (8)$$

Here ve, tpe and tse are Word-Sequence, POS-tag sequence and SRL sequence encodings, encoded using a simple RNN based Language Model, POS tagger and SRL tagger respectively.

4 TENSOR REPRESENTATION

The overall tensor representation of any n-gram g within an input-sentence s i.e. $X_g \in R^{(n-1+4)\times d\times 3}$, encoding all the features discussed in section 3 is computed with equation 9.

$$X_g = [C_{1\to(t-2)}; C_{t-1}; NG_g; C_{t+n}; C_{(t+n+1)\to T}] \quad (9)$$

Figure 2 depicts the feature distribution within a tensor representation X_g.

5 N-GRAM CLASSIFIER

As evident in Figure 1, our model is simplistic CNN based classifier with following layers.

(1) *Convulational Layer*: This includes D filters of dimensions $H \times W \times 2$ convolving over tensor representation X of any n-gram, thus outputting a $H \times W \times D$ dimensional tensor. H,W,D are hyper-parameters to be learnt on held-out dataset. Depending upon the dimensions of filters we may apply zero-padding to X if required.
(2) *Relu Layer*: This layer induces non-linearity to output of Convukational Layer.
(3) *K-Max Pooling* : This layer extracts k maximum values in each of the D matrices (output of each filter) and presents them as a single dense k.D dimensional vector. k is a hyper-parameter to be learnt.
(4) *Fully Connected Layer*: This layer takes in the k.D dimensional vector from previous layer as input and outputs a $|C| + 1$ dimensional vector where C is the set of category-labels.
(5) *Softmax Layer*: Finally, this layer applies Softmax function to the $|C| + 1$ dimensional output from previous layer to compute the desired $|C| + 1$ dimensional probability vector P_g. Here ${P^c}_g$ is the probability of n-gram g having label $c \in C$, whereas ${P^{|C|+1}}_g$ is the probability of n-gram g having label as *NONE* (not belonging to any category).

The model is trained by optimizing simple Cross-entropy function [12].

6 PRELIMINARY EXPERIMENTATION

For preliminary experimentation, we tested the proposed model (for *Phrase-labeling*) by performing Named Entity Recognition (NER) task on CoNLL 2000 [18] dataset, and compared the obtained results with benchmark *basic RNN* approach.
However, in future we do intend to test this approach for various

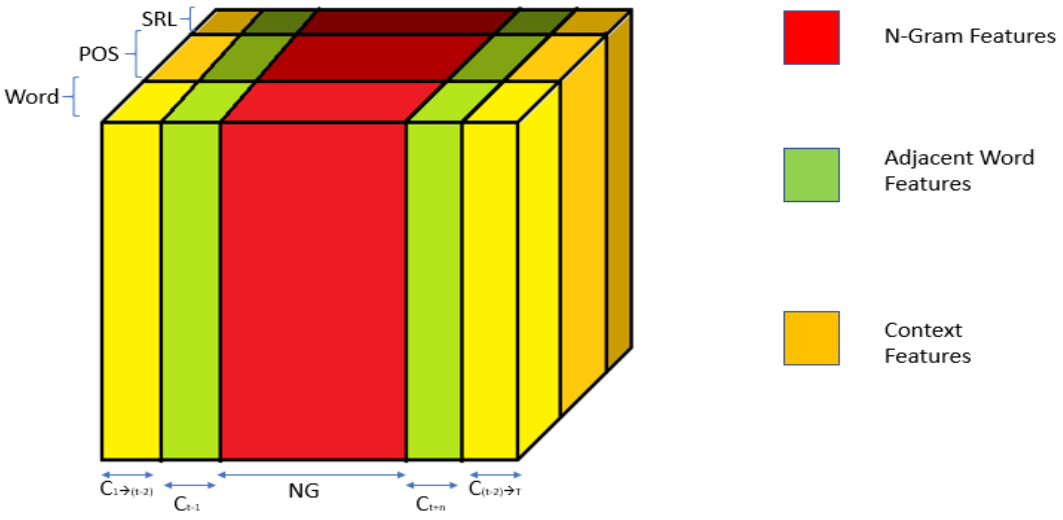

Figure 2: Distribution of features within Tensor representation of any n-gram.

other standard *Phrase Labelling* tasks such as Chunking, Biomedical Entity Recognition etc.

Overall experimental procedure involved training a set of neural-network models listed as follows. All the models listed below are trained on *Brown Corpus* [13].

(1) A Word2Vec model trained using Skip-Gram [8] algorithm.
(2) A simple linear auto-encoder to encode any $|T_{pos}|$ dimensional one-hot vector representing a POS-tag into a $|d|$ dimensional vector (T_{pos} is the POS tag-set).
(3) A simple linear auto-encoder to encode any $|T_{srl}|$ dimensional one-hot vector representing a SRL-tag into a $|d|$ dimensional vector (T_{srl} is the SRL tag-set).
(4) A simple RNN based Language-Model (LM) [14] trained through Stochastic Gradient Descent. This LM is used to encode a sequence of words as d-dimensional vector.
(5) A simple RNN based POS-tagger trained through Stochastic Gradient Descent to encode a sequence of POS-tags (mapped to specific word-sequence) as a d-dimensional vector.
(6) A simple RNN based SRL-tagger trained through Stochastic Gradient Descent to encode a sequence of SRL-tags (mapped to specific word-sequence) as a d-dimensional vector.

6.1 Training Data Normalization

This section highlights a significant issue that appears during training. It is logically evident that more than 99% of all the N-grams extracted from training sentences (raw N-Gram set) would belong to *NONE* category. Thus, training the CNN based model on this raw N-Gram set would most likely result in biased classifier.

To address this issue we normalize the training-dataset by retaining only $Count_{None}$ N-Grams belonging to *None* category from raw N-Gram dataset. $Count_{None}$ is given by equation 10.

$$Count_{None} = nf.(\Sigma_{c \in C} Count_c) \tag{10}$$

Here C is the set of *Phrase-Labels* (PER, LOC, ORG and MISC), $Count_c$ is the count of n-gram having category label c and nf is the normalization-factor. nf is a hyper-parameter to be trained on held-out set.

We normalized the raw N-Gram set (NG_{Raw}) by applying equation 11.

$$NG_{Normal} = NG_{T \neq NONE} \cup NG_{T=NONE} \tag{11}$$

Here $NG_{T \neq NONE}$ is computed by applying equation 11.

$$NG_{T \neq NONE} = \{g | g \in NG_{Raw}; T_g \neq NONE\} \tag{12}$$

$NG_{T=NONE}$ is a randomly sampled subset of set T' of size $Count_{None}$. T' is computed by equation 13.

$$T' = \{g | g \in NG_{Raw}; T_g = NONE; min \leq l_g \leq max\} \tag{13}$$

Here l_g and T_g are the length and phrase-label of n-gram g respectively. *max* and *min* are computed by equations 14 and 15.

$$max = argmax_{g \in NG_{T \neq NONE}}(l_g) \tag{14}$$

$$min = argmin_{g \in NG_{T \neq NONE}}(l_g) \tag{15}$$

In other words, we randomly sampled $nf.(\Sigma_{c \in C} Count_c)$ n-grams from a set of all n-grams within *raw N-gram set* that belong to *NONE* category and have the length l ranging from *min* to *max*, where *min* and *max* are minimum and maximum lengths of n-grams within *raw N-gram set* that do-not possess category label *NONE* ($NG_{T \neq NONE}$).

6.2 Benchmark Approach

We compared the performance of our proposed model with the performance of simple RNN based Named Entity Recogniser.

The input $X^t \in R^{3d}$ to this benchmark simple RNN based model at any given time-step t is the concatenation of d-dimensional feature-vectors $v^t \in R^d$, $tp^t \in R^d$ and $ts^t \in R^d$ representing Word-embedding, POS-tag encoding and SRL-tag encoding at the t^{th} time-step (Equation 16).

$$X^t = [v^t; tp^t; ts^t] \tag{16}$$

	CNN based approach				Simple RNN approach			
	PER	LOC	ORG	MISC	PER	LOC	ORG	MISC
Precision	86.4	81.85	81.8	72.3	75.04	55.8	68.06	69.48
Recall	87.1	81.2	78.76	65.09	75.69	57.2	68.31	70.02
F-Score	86.75	81.5	80.25	68.5	75.36	56.49	68.18	69.74

Table 1: Preliminary Results

It is important to note that our intention within this paper is to test the proposed hypothesis, rather than to improve the existing benchmark performance. Thus we compare our simple CNN model with only single benchmark basic RNN model. However our future research would involve evaluation of more sophisticated architectures of CNN based NER models that could outperform state-of-the-art RNN/LSTM architectures.

7 PRELIMINARY RESULTS AND INFERENCES

Figure 3 indicates the plot of Tensor-representations of 500 N-Grams randomly sampled from training-dataset in a 2D space. The $(n-1+4) \times d \times 2$ dimensional tensor representations of all 500 n-grams are projected in 2D space using Principle Component Analysis (PCA) [4] approach.

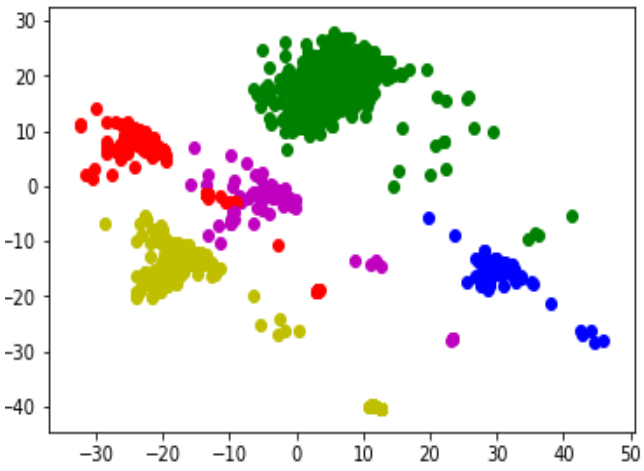

Figure 3: Plots of Tensor representation of 500 N-Grams randomly sampled from test-dataset. Here Red:PER; Blue:LOC; Yellow:ORG; Purple:MISC; Green:NONE

It is evident from figure 3, that the selected features do result in segregation of N-Grams having common Phrase-labels, for all four label types (i.e. PER, ORG, LOC, MISC). Thus a good classification model would generalize well on the training dataset. Hence we can conclude that the Tensor representation of an N-gram described in section 4 is relevant for the CNN based phrase-labelling task.
Table 1 outlines the results obtained on the test dataset. As CNN based approach predicts *Phrase-labels* for N-grams, the Precision and Recall values w.r.t. each label class is computed directly. Whereas for benchmark approach, these values are computed by implementing CoNLL eval [2] script.

The results outlined in table 1 indicate that proposed approach outperforms the benchmark *basic RNN* based approach.

8 CONCLUSION AND FUTURE WORK

Within this paper we outlined the basic hypothesis that the task of Phrase-labelling is not temporal but rather spatial in nature. Thus we argued that CNN based architectures would be more efficient for Phrase-labeling than RNN based architectures.
The results outlined within section 7, provide initial evidence for this proposed hypothesis.
Future research-work would include more exhaustive testing of the hypothesis for numerous Phrase-labeling tasks within NLP/IR domain such as Chunking, Biomedical NLP tasks etc. on multiple datasets.
Our proposed model is a very basic CNN based architecture which outperforms the basic RNN model. As future research-work we would attempt to design sophisticated CNN based Phrase-tagger architectures (with numerous Convolution, Pooling, Capsule and fully connected layers etc.) and compare the performances with other sophisticated RNN based models such as Bi-LSTM-CRF, Bi-LSTM-CNN-CRF etc. Further we would also experiment with different features to be considered by proposed model including both generic as well as task specific features.

9 ACKNOWLEDGEMENT

We thank the anonymous reviewers for their valuable feedback and suggestions. This research is funded by **Hardiman Research Scholarship** at *National University of Ireland, Galway*

REFERENCES

[1] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics* 22, 1 (1996), 39–71.
[2] Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning*. Association for Computational Linguistics, 149–164.
[3] Soravit Changpinyo, Hexiang Hu, and Fei Sha. 2018. Multi-Task Learning for Sequence Tagging: An Empirical Study. *arXiv preprint arXiv:1808.04151* (2018).
[4] Songcan Chen and Yulian Zhu. 2004. Subpattern-based principle component analysis. *Pattern Recognition* 37, 5 (2004), 1081–1083.
[5] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4 (2016), 357–370.
[6] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research* 12, Aug (2011), 2493–2537.
[7] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.
[8] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[11] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).
[12] Chun Hung Li and CK Lee. 1993. Minimum cross entropy thresholding. *Pattern recognition* 26, 4 (1993), 617–625.
[13] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. (1993).
[14] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.
[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[16] Lawrence R Rabiner and Biing-Hwang Juang. 1986. An introduction to hidden Markov models. *ieee assp magazine* 3, 1 (1986), 4–16.
[17] Erik F Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. *arXiv preprint cs/0009008* (2000).
[18] Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050* (2003).
[19] Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050* (2003).
[20] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928* (2017).
[21] Sifatullah Siddiqi and Aditi Sharan. 2015. Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications* 109, 2 (2015).
[22] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098* (2017).
[23] Xu Wang, Chen Yang, and Renchu Guan. 2018. A comparative study for biomedical named entity recognition. *International Journal of Machine Learning and Cybernetics* 9, 3 (2018), 373–382.
[24] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601* (2018).
[25] Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2145–2158.
[26] Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. *arXiv preprint arXiv:1806.04470* (2018).