Literature Survey of statistical, deep and reinforcement learning in Natural Language Processing

Pranav Kaushik (*Author*)
Information Technology Dept.
Amity School of Engineering and Technology
Delhi, India

Akanksha Rai Sharma (*Author*)
Electronics and Communication Dept.
Maharaja Surajmal Institute of Techology
Delhi, India

Abstract-This paper underlines the necessity to incorporate Deep learning and Neural networking in language models under scrutiny for Natural Language Processing. The paper describes various statistical models proposed and the limitations incurred in the same due to limited intelligence of a machine. We have discussed different neural networks highlighting the importance of Convolutional Neural Networking. We have discussed about open source software TensorFlow that works on Deep learning and the edge it has over the conventional models. Also we have recommended Reinforcement learning as an extension to neural networking which is widely used in gaming for the purpose of Natural Language Processing. We can utilize the reward-driven algorithm for better results.

Keywords- Natural Language Processing; Information Extraction; Statistical models; Neural Networks; Deep Learning; TensorFlow; Reinforcement Learning

I. INTRODUCTION

Natural Language Processing (NLP) is one of the dominant fields in data mining. With the increasing importance of Big Data Analytics today, NLP plays a major role in acquiring relevant information of importance to business and intelligence. Millions of items are uploaded on Web everyday, with relevant as well as irrelevant data. Information retrieval and extraction from reviews, comments, social media etc by customers is a complex task since most of the information is in semi-structured and unstructured form. Ambiguity of large corpora on Web underlines the need for decent and efficient data mining techniques.

The branch of NLP predominantly works to analyze, summarize or retrieve pertinent information from the large pool of data available. Exploration in this field dates back to 1950 when Turing's article on 'Computing Machinery and Intelligence' was published [1] and Message Understanding Conferences in '90s. NLP requires combination of linguistics and computational knowledge. It can be done for various languages. For English, various problems incurred during information extraction include- paraphrasing, idioms, rhetoric, metaphors etc. [2]

Disambiguation of the language is the key to parsing and information extraction in NLP. Sentence parsing, identification of words at morphological level (using lexicon, NER etc) and relation extraction are sequential functions to be performed for NLP. Various models [3] such as Wrappers in Web corpus, Named Entity Recognition for identification of proper noun by type, Part of Speech tagging etc has been proposed. Information Extraction can be used to extract data for predetermined slots in a particular frame [4].Generative models for NLP using joint probability of a word and its label/tag, extending to Hidden Markov Model(HMM) and Trigram HMM has been described in [5]. It also states use of pseudo-random words and their mapping for NLP of unknown words in small corpus. In [6] authors have discussed about models to elucidate Natural Language for

- Unordered arrangement of adjectives - direct evidence, transitive closure and back-off bigram model
- Isolation of compound nouns – adjency and dependency model
- Interpolation of web and corpus counts to increase flexibility.

Relation extraction is prominent for Information Extraction as it is a transformational step from unstructured pool to partially structures form. In [3] author has compared Feature-based and kernel-based relation extraction for large corpus. Complexity of linguistics because of metaphors and idioms hinder relation extraction by aforementioned approach, hence making them unsuitable.

For example:

Maria was on Cloud 9 on hearing the news.

Even though Cloud 9 refers to the feeling of extreme happiness or euphoria, the machine translator will misinterpret the sentence and program on extracting relation between Maria and Cloud 9. Various leads have been taken by Linguamatics to design efficient NLP platforms [7].

Rhetorical Structure Theory is crucial to identify coherence of text and relation between texts using nuclei-satellites relation. rstWeb[8] is a tool for browser based annotation that works on creation of multinuclear and multi-mode workflow. The 2 major classifications of approaches for NLP [3] are-

1. Statistical learning approach
2. Rule-based approach

In this paper we will be discussing Statistical Approach in detail, describing various models in Section 2. Section 3 of the paper describes working of Skip Gram model and CBOW. In Section 4 we discuss the different Neural Networks and Tensor flow, open source software used for NLP. Section 5 will be a descriptive analysis Reinforcement learning and its future scope and analysis of language. We finally conclude in Section 6.

II. STATISTICAL LEARNING APPROACH

In statistical learning, we represent every observation as a feature vector and infer a relationship between the word and its corresponding vector. Different models for this approach are

- Hidden Markov Model
- Trigram Hidden Markov Model
- Maximum Entropy Markov Model
- Conditional Random Field

A. Hidden Markov Model

Hidden Markov Model is a double stochastic method with 3 major elements [3]

1. Input ($x_1 \ldots\ldots x_n$)
2. Label/Tag/Hidden state ($y_1 \ldots\ldots y_n$)
3. transition probabilities, p(y|x) or p(x,y)

HMM randomly assigns transition probabilities and observational probabilities; therefore there is a scope of improvement as proposed by authors in [9].Since it is controlled by transmission matrix, it is not possible to determine which states produce a given output. For an input x , the output of HMM is given as [5]

$$f(x)= \arg\max p(y|x) \quad (1)$$

For a noisy channel

$$f(x)= \arg\max p(x|y) \quad (2)$$

B. Trigram Hidden Markov Model

A Trigram Hidden Markov Model is given as [5]

$$P(x_1..x_n,y_1..y_{n+1})= \Pi f(y_i|y_{i-2,i-1}) \;\; \Pi g(x_i|y_i) \quad (3)$$

For i € [1,n+1] & y_{n+1} = STOP.

Where f (y_i | $y_{i-2,}y_{i-1}$) is the probability of bigram tags $y_{i-2,}$ y_{i-1} before y_i and g(x_i|y_i) is the probability of the input tag y.

C. Maximum Entropy Markov Model

In Maximum Entropy Markov Model (MEMM), transition and observation functions are given by a single function that gives the current observation on the basis of previous input [10].It is a discriminative model for Named Entity Recognition.

D. Conditional Random Field

Conditional Random Field (CRF) makes present observation on the basis of previous and future tags/labels. It is an extension to MEMM. It solves the label bias problem (incurred in MEMM) and is trained by maximum likelihood or MAP estimation [11].

Limitation of CRF includes inadequate extraction of long-distance dependencies in different parts of document [4] and slow convergence of training algorithm.

III. CONTINUOUS BAG OF WORDS V/S CONTINUOUS SKIP GRAM MODEL

Continuous Bag of words (CBOW) model predicts current word on the basis of equal number of words before and after the word of reference (current word).Similar to feedforward Neural Network Language Model, weight matrix between input and projection is shared without dependence on other words in input [12].

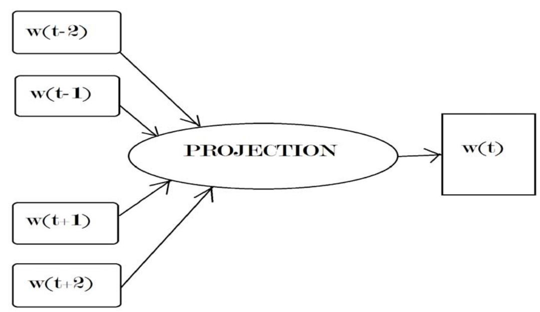

Fig. 1. Block diagram of CBOW where w(t) is the current word and w(t-2) w(t-1) w(t+1) w(t+2) are words before and after w(t).

Continuous Skip Gram Model is an exact mirror image to CBOW as it predicts the context on the basis of current word. Though both CBOW and Continuous Skip Gram feed log linear classifier [13] to input, the latter may result in computational complexity.

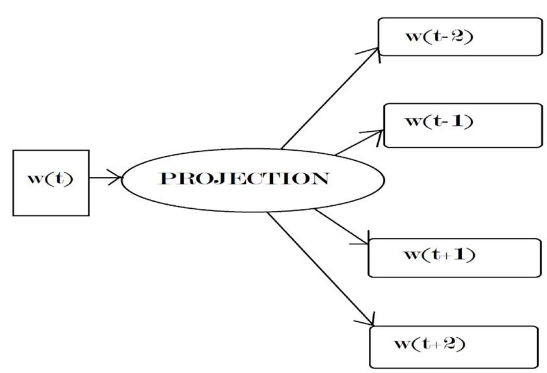

Fig. 2. Block diagram of Skip Gram Model where w(t) is the current word and w(t-2) w(t-1) w(t+1) w(t+2) are words before and after w(t).

word2vec models incorporate CBOW and SG models with the help of word embedding and perception algorithm for NLP in neural networks [14].

IV. NEURAL NETWORKS IN NLP

Deep Learning and Neural networks are gaining importance in the field of NLP with hidden states between the input and output and extensive networking to provide best results [15].

In Recursive Neural Network, semantics are isolated via tree structures. Since textual tree construction can be time consuming for long sentences, it is inefficient. Recurrent Neural Network can extract contextual information by utilizing stored previous text in the form of fixed sized hidden layer. The problem with the same is its bias towards end of the document. Hence keyword in the other parts of the document will be ignored.

One of the best alternatives in neural networks is Convolutional Neural Network is an unbiased model that uses convolutional kernels as a part of its deep learning architecture. 3 layers of CNN are [16]-

1. Convolution layer
2. Pooling layer
3. Activation layer (fully-connected)

Deep learning in CNN is achieved with convolving filters of variable widths and feature map. Pooling is responsible for downsampling of the matrix from filters whereas Fully-connected layer computes class score [17]. Deep neural networks open source software Tensor Flow has been proposed for application in [Paul] Youtube Recommendation with the help of matrix factorization approach in minimizing cross entropy loss.

Google's continuous research in the field of AI and machine learning is making a big break through future NLP applications. Likewise GoogleBrain project is one of the commendable projects in this field.

In November 2015, Google open sourced TensorFlow, which is one of the projects under Googlebrain. TensorFlow is an open source software library for machine learning which is used by google for many of Google product, such as speech recognition, Gmail, Google Photos etc. TensorFlow is now being widely used for research purpose, creating a number of useful applications. It runs on multiple CPUs and GPUs (with optional CUDA extensions for general-purpose computing on graphics processing units).It can work on different platforms like Linux, Windows and Mac OSX .It also works on Android and Apple's iOS [18]. To understand how it works we must first understand what "Tensor" is. So, first, we recall matrix multiplication, which is given as {

v[x]→vector is a simple array of one dimension

m[x][y][z]→matrix (is a 2 0r 3 dimensional)

t[x][y][z][?][?]..→ tensor (is arbitrary large number of dimension)

}

TensorFlow [19] is based on Deep Learning of Neural networks such that the input is given as a tensor and then that tensor flows through nodes in the neural network adding some weight to it and the softmax function in the final layer of the neural networks. TensorFlow [20,21] library can easily be downloaded and installed in your system and coding in tensor flow is done in python .So TensorFlow works with the python API (compatible with python or python3).It is loaded up with many different packages like speech recognition and image recognition etc.

V. REINFORCEMENT LEARNING IN NLP

Reinforcement Learning as well as deep learning is promising is the field of resolving control optimization problems to select best action from the states traversed in a system. It is primarily applied in systems with large number of states having complex stochastic structures [22].

Q-Learning is a subtype to RL and is used to learn optimal actions in every traversed state by the system via trial and error as represented in Figure 3. It is based on incremental neural networks such that Q-factor is updated at every next state and hence the new value of Q-factor overwrites on the old value saving significant memory space.

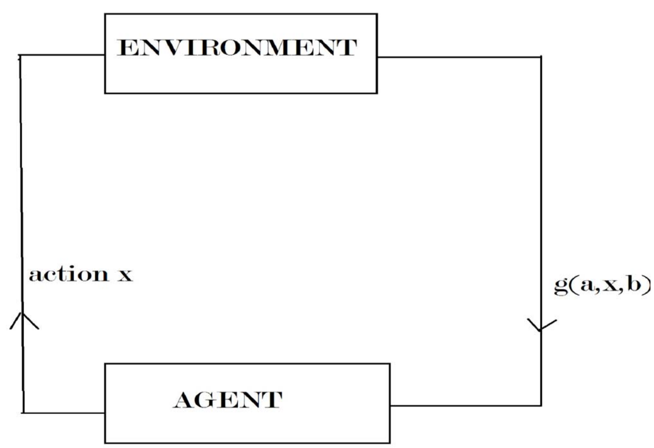

Fig 3. Reinforcement Learning mechanism provides feedback for every current state to produce optimal function.

For current state a and next state b,

$$feedback = g(a,x,b) + \lambda_{max}\, Q(b,y) \qquad (4)$$

Where x is the action performed on the current state and λ is the discounting factor. Q(b,y) gives the Q - function/function approximation of next state. Therefore updating equation in Q-Learning mechanism is given by

$$Q(a,x) \leftarrow [1-\gamma]Q(a,x) + \gamma\, [feedback] \qquad (5)$$

From (1) & (2) we get

$$Q(a,x) \leftarrow [1-\gamma]\, Q(a,x) + \gamma\, [g(a,x,b) + \lambda_{max}\, Q(b,y)] \quad (6)$$

RL uses 2 approaches for function approximation, namely vale function approach and policy gradient approach. Value function is an 'avarice' based approach to maximize reward time in RL. Though limitations in this approach can be listed as [23]

1. It does not provide stochastic policies, which are considered optimal

2. Unpredictable changes do not assure convergence with respect to algorithms.

Hence policy-gradient method is gaining importance recently. It is stochastic process wherein policy parameters are updated for every differential gradient of a policy.

For performance of corresponding policy ρ, policy gradient is given by

$$\Delta\theta \approx \alpha\, \partial\rho/\partial\theta \quad (7)$$

where α is the positive definite step size and θ is the vector of policy parameters.

Other function approximators include temporal differences, back propagation etc. Various conditions of optimality have been proposed in Markov Decision Process and Bellman Equations [24].Neuroevolution utilizes genetic algorithms for artificial evolution of neural networking is forming basis for increased efficiency in RL methods. In [25] NeuroEvolution of Augmenting Topologies minimizes dimensionality of search space of connection weights by utilizing structures in effective way. Also in [26], authors propose Deep Reinforcement Relevance Network processing language though RL procedures, supplemented by Deep Learning.

VI. CONCLUSION

We conclude that neural networks and deep learning resolves most of the problems incurred in NLP. The hidden states between input word and output vector form intensive network for thorough and efficient learning. This technology can be used as the backbone of Artificial Intelligence. Future works to be done in this field include Cross Language IR and machine-human dialog. Also Reinforcement learning is making reliable progress in gaming and spreading its application in various fields including Natural Language Processing. It focuses on higher long term reward such as solution to pole balancing problem. The reward driven software is an integral part of TensorFlow as well.

REFERENCES

[1]Jona, "Natural Language Processing", https://en.wikipedia.org/wiki/Natural_language_processing.

[2] Paul Shoebotttom, "Syntax- English sentence structure", esl.fis.edu/learners/advice/syntax.htm.

[3]Jing Jiang, "Information Extraction from Text", Springer Science+Business Media,DOI 10.1007/978-1-4614-3223-4_2,pp 11-41,2012.

[4] Raymond J. Mooney and Razwan Bunescu, "Mining Knowledge from Text Using Information Extraction", SIGKDD Explorations, Vol.7 Issue. 1 pp 3-10.

[5] Michael Collins, Chapter 2: Tagging Problems, and Hidden Markov Models, Columbia University.

[6] , Mirella Lapata and Frank Keller, "Web-based models for Natural Language Processing"; ACM Transactions on speech and Language Processing , Vol 2 No.1 , pp 1-30, Feb 2005.

[7] Linguamatics ,"Linguamatics", https://www.linguamatics.com/

[8]Amir Zeldes, "rstWeb-A Browser based-Annotation Interface for Rhetorical Structure Theory and Discourse Relations", In Proceedings of NAACL-HLT 2016(Demonstrations),pp 1-5,June 2016.

[9]Afroza Sulatana,, Abdelwahab Hamou-Lhadi and Mario Couture, "An Improved Hidden Markov Model for Anomaly Detection Using Frequent Common Patterns" ,IEEE ICC Communication and Information Systems Security Symposium, IEEE, Ottawa, ON, 10-15 June, 2012, pp 1113 - 1117.

[10] Andrew McCallum ,Dayne Freitag and Fernando Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation" ,Proceedings of 17th International Conference on Machine Learnig,pp591-598.

[11] John Lafferty, Andrew McCallum and Fernando Pereira, "Conditional Random Fields: Probabilistic Models for Segmentation and Labeling Sequence Data" ,Proceedings of the 18th International Conference on Machine Learning, pages 282-289,June2001.

[12]Tomas Mikolov ,Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representation in Vector Space", arXiv:1301.3781v3, pp 1-12, Sept 2013.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality" , unpublished, pp 1-9.

[14]Xin Rong, "word2vec Parameter Learning Explained" , unpublished , pp 1-21.

[15] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao, " Recurrent Convolutional Neural Networks for Text Classification, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence", pp 2267-2273,2015

[16] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting

Liu, Xingxing Wang, Gang Wang, "Recent Advances in Convolutional Neural Networks" , arXiv:1512-07108v5 ,pp 1-37,2017.

[17] Karpathy, "Convolutional Neural Networks (CNNs / ConvNets)" , css\n.github.io/convolutional-networks/

[18]Ancheta Wis , "TensorFlow" , https://en.wikipedia.org/wiki/TensorFlow#TensorFlow

[19]Wolfram MathWorld, "Tensor", http://mathworld.wolfram.com/Tensor.html

[20] Mart´ın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", Preliminary White Paper, pp 1-19, 2015.

[21]TensorFlow,"An open source software library for Machine Intelligence", https://www.tensorflow.org/.

[22] Abhijit Gosavi, "Neural Networks and Reinforcement Learning", web.mst.edu/~gosavia/neral_networks_RL.pdf

[23] ,Richard S. Sutton, David McAllester, Satinder Singh, Yishay Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation", Advances in Neural Information Processing Systems 12, pp. 1057–1063, 2000.

[24] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning:

“An introduction, https://webdocs.cs.ualberta.ca/~sutton/book/bookdraft2016sep.pdf, 2016

[25]Kenneth O.Stanley and Rist Mikkulainen, Efficient Reinforcement through Evolving Neural Network Topologies”, In Proceedings of Genetic and Evolutionary Computational Conference 2002

[26]Deep Reinforcement Learning with Natural Language Action Space,Ji He,Jianshu Chen,Xiaoding He,Jianfeng Gao,Lihong Li,Li Deng,Mari Ostendorf, Proceedings of the 54th Annual Meetings of the Association for Computational Linguistics ,pp 1621-1630,August 2016