IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

WORD BASED PREDICTION USING LSTM NEURAL NETWORKS FOR LANGUAGE MODELING

1. ASWINI THOTA

PG Scholar, Department of Computer Science,

SVKP & Dr K S Raju Arts & Science College,

Penugonda, W.G.Dt., A.P, India

1. P. SRINIVASA REDDY

PG Scholar, Department of Computer Science,

SVKP & Dr K S Raju Arts & Science College,

Penugonda, W.G.Dt., A.P, India

ABSTRACT

Neural networks have become increasingly popular for the task of language modeling. Whereas feed-forward networks only exploit a fixed context length to predict the next word of a sequence, conceptually, standard recurrent neural networks can take into account all of the predecessor words. On the other hand, it is well known that recurrent networks are difficult to train and therefore are unlikely to show the full potential of recurrent models. These problems are addressed by the Long Short-Term Memory neural network architecture. In this work, we analyze this type of network on an English and a large French language modeling task. Experiments show improvements of about 8% relative in perplexity over standard recurrent neural network LMs. In addition, we gain considerable improvements in WER on top of a state-of-the-art speech recognition system. Index Terms: language modeling, recurrent neural networks.

INTRODUCTION

1.1 Introduction

A language model is the core component of modern Natural Language Processing (NLP). It's a statistical tool that analyzes the pattern of human language for the prediction of words. NLP-based applications use language models for a variety of tasks, such as audio to text conversion, speech recognition, sentiment analysis, summarization, spell correction, etc. Language modeling is central to many important natural language processing tasks. Recently, neural-network-based language models have demonstrated better performance than classical methods both standalone and as part of more challenging natural language processing tasks. Language modeling is the art of determining the probability of a sequence of words. This is

useful in a large variety of areas including speech recognition, optical character recognition, handwriting recognition, machine translation, and spelling correction. Developing better language models often results in models that perform better on their intended natural language processing task. This is the motivation for developing better and more accurate language. Language Models determine the probability of the next word by analyzing the text in data. These models interpret the data by feeding it through algorithms. The algorithms are responsible for creating rules for the context in natural language. The models are prepared for the prediction of words by learning the features and characteristics of a language. With this learning, the model prepares itself for understanding phrases and predicting the next words in sentences. For training a language model, a number of probabilistic approaches are used. These approaches vary on the basis of the purpose for which a language model is created. The amount of text data to be analysed and the math applied for analysis makes a difference in the approach followed for creating and training a language model. For example, a language model used for predicting the next word in a search query will be absolutely different from those used in predicting the next word in a long document (such as Google Docs). The approach followed to train the model would be unique in both cases. We apply the MDL principle to our problem to effectively manage an unknown number of clusters (i.e., an unknown number of templates).In our method, document clustering and template extraction are done together at once. The MDL cost is the number of bits required to describe data with a model and the model in our problem is the description of clusters represented by templates. Since a large number of web documents are massively crawled from the web, the scalability of template extraction algorithms is very important to be used practically. Thus, we extend MinHash technique to estimate the MDL cost quickly, so that a large number of documents can be processed. Experimental results with real life data sets up to15 GB confirmed the effectiveness and scalability of our algorithms. Our solution is much faster than related work and shows significantly better accuracy.

2. LITERATURE SURVEY

As discussed above, the advance of the internal structure unit can improve the performance of RNN in natural language word prediction. However, deeper network or increased number of hidden nodes often induces the network degradation and over-fitting issues. Li et al. [11] proposed a robust independent recurrent neural network (IndRNN) where neurons in the same layer are set independently of each other, and neurons in different layers are cross-connected. This method enabled the model to be robust at a relatively large network depth and effectively overcame the problem of network degradation. Zilly et al. [12] proposed Recurrent Highway Network (RHN), which is an extension of the LSTM, to allow multiple hidden state updates at each time step. In RHN, the degradation of the weight matrix was alleviated so that the vanishing gradient was resolved. The residual network [13] addressed the gradient dissipation problem in the backpropagation process by utilizing the nature of the identity connection.

The convolutional neural network (CNN) [20] consists of deep-stacked convolutional layers,

which has a strong processing ability for local information of data. CNN has been successfully adopted in many computer vision applications. In this paper, utilizing the ability of local information processing of CNN and the ability of the word sequences feature extracting of residual-connected MGU network, multi-window convolution and residual-connected MGU network (MCNN-ReMGU) is proposed. First, the convolution kernels extract the local feature relationships between the word sequences. Then the residual-connected MGU network fully learns the long dependency relationship between the word sequences. Thus, both global and local feature information of the word sequence is thoroughly used in word prediction. Also, L2-norm [21], [22] and batch normalization are employed to avoid the over-fitting of the network.

In order to reduce the over-fitting problem, two approaches: the dropout [14]–[16] operation and batch normalization [17]–[19] are widely used. In the training process, the dropout randomly prevented some neurons from participating in the training and weakened the joint adaptability between the neuron nodes to avoid over-fitting. The batch normalization normalized the input data to ensure that the data distribution remains unchanged and improved the generalization ability of the model.

3. OVERVIEW OF THE SYSTEM

3.1 Existing System

A goal of statistical language modeling is to learn the joint probability function of sequences of words. This is intrinsically difficult because of the curse of dimensionality. A distributed representation for each word (i.e. a similarity between words) along with the probability function for word sequences, expressed with these representations. Generalization is obtained because a sequence of words that has never been seen before gets high probability if it is made of words that are similar to words forming an already seen sentence.

3.2 Proposed System:

Curse of dimensionality and long sequence decencies are addressed using Long Short Term Memory neural networks, which are eventually memorize long sequencies from the data. This helps us to build efficient language models.

3.3 System Modules

Word Embeddings

Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation. They are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems.

A word embedding is a learned representation for text where words that have the same meaning have a similar representation. It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems.

Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural

network, and hence the technique is often lumped into the field of deep learning.

The distributed representation is learned based on the usage of words. This allows words that are used in similar ways to result in having similar representations, naturally capturing their meaning. This can be contrasted with the crisp but fragile representation in a bag-of-words model where, unless explicitly managed, different words have different representations, regardless of how they are used

Word Embedding Algorithms

Word embedding methods learn a real-valued vector representation for a predefined fixed sized vocabulary from a corpus of text. The learning process is either joint with the neural network model on some task, such as document classification, or is an unsupervised process, using document statistics. This section reviews three techniques that can be used to learn a word embedding from text data.

Embedding Layer

An embedding layer, for lack of a better name, is a word embedding that is learned jointly with a neural network model on a specific natural language processing task, such as language modeling or document classification. It requires that document text be cleaned and prepared such that each word is one hot encoded. The size of the vector space is specified as part of the model, such as 50, 100, or 300 dimensions. The vectors are initialized with small random numbers. The embedding layer is used on the front end of a neural network and is fit in a supervised way using the Backpropagation algorithm.

The one hot encoded words are mapped to the word vectors. If a Multilayer Perceptron model is used, then the word vectors are concatenated before being fed as input to the model. If a recurrent neural network is used, then each word may be taken as one input in a sequence. This approach of learning an embedding layer requires a lot of training data and can be slow, but will learn an embedding both targeted to the specific text data and the NLP task.

Word2Vec

Word2Vec is a statistical method for efficiently learning a standalone word embedding from a text corpus. It was developed by Tomas Mikolov, et al. at Google in 2013 as a response to make the neural-network-based training of the embedding more efficient and since then has become the de facto standard for developing pre-trained word embedding

- ˆ Continuous Bag-of-Words, or CBOW model.
- ˆ Continuous Skip-Gram Model.

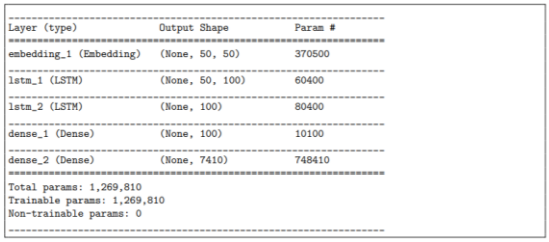
4. RESULTS

```
BOOK I.

I went down yesterday to the Piraeus with Glaucon the son of Ariston,
that I might offer up my prayers to the goddess (Bendis, the Thracian
Artemis.); and also because I wanted to see in what
```

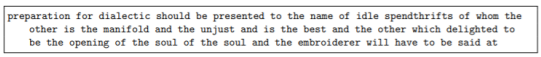
```
['book', 'i', 'i', 'went', 'down', 'yesterday', 'to', 'the', 'piraeus', 'with', 'glaucon',
 'the', 'son', 'of', 'ariston', 'that', 'i', 'might', 'offer', 'up', 'my', 'prayers',
 'to', 'the', 'goddess', 'bendis', 'the', 'thracian', 'artemis', 'and', 'also',
 'because', 'i', 'wanted', 'to', 'see', 'in', 'what', 'manner', 'they', 'would',
 'celebrate', 'the', 'festival', 'which', 'was', 'a', 'new', 'thing', 'i', 'was',
 'delighted', 'with', 'the', 'procession', 'of', 'the', 'inhabitants', 'but', 'that',
 'of', 'the', 'thracians', 'was', 'equally', 'if', 'not', 'more', 'beautiful', 'when',
 'we', 'had', 'finished', 'our', 'prayers', 'and', 'viewed', 'the', 'spectacle', 'we',
 'turned', 'in', 'the', 'direction', 'of', 'the', 'city', 'and', 'at', 'that',
 'instant', 'polemarchus', 'the', 'son', 'of', 'cephalus', 'chanced', 'to', 'catch',
 'sight', 'of', 'us', 'from', 'a', 'distance', 'as', 'we', 'were', 'starting', 'on',
 'our', 'way', 'home', 'and', 'told', 'his', 'servant', 'to', 'run', 'and', 'bid', 'us',
 'wait', 'for', 'him', 'the', 'servant', 'took', 'hold', 'of', 'me', 'by', 'the',
 'cloak', 'behind', 'and', 'said', 'polemarchus', 'desires', 'you', 'to', 'wait', 'i',
 'turned', 'round', 'and', 'asked', 'him', 'where', 'his', 'master', 'was', 'there',
 'he', 'is', 'said', 'the', 'youth', 'coming', 'after', 'you', 'if', 'you', 'will',
 'only', 'wait', 'certainly', 'we', 'will', 'said', 'glaucon', 'and', 'in', 'a', 'few',
 'minutes', 'polemarchus', 'appeared', 'and', 'with', 'him', 'adeimantus', 'glaucons',
```

Dataset



Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 50, 50)	370500
lstm_1 (LSTM)	(None, 50, 100)	60400
lstm_2 (LSTM)	(None, 100)	80400
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 7410)	748410

Total params: 1,269,810
Trainable params: 1,269,810
Non-trainable params: 0



preparation for dialectic should be presented to the name of idle spendthrifts of whom the other is the manifold and the unjust and is the best and the other which delighted to be the opening of the soul of the soul and the embroiderer will have to be said at

Result

5. CONCLUSION

Introduced Technology prediction of sequence of words is strategically both a defensive and offensive activity. It can assist in resource allocation and minimize the adverse impacts or maximize the favorable impacts of game-changing technology trends. In a general sense, it is wise to be circumspect by analyzing the state of trend-setting technologies, their future outlook, and their potential disruptive impact on industries, society, security, and the economy.

This summarizes and condenses key points from throughout the report, presented in the form of important system attributes and, second, steps to build a persistent prediction system for disruptive technologies.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

[2] Y. Goldberg, “A primer on neural network models for natural language processing,” CoRR, vol. abs/1510.00726, 2015.

[3] C. D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. Cambridge, MA, USA: MIT Press, 1999.

[4] P. Kłosowski, “Speech Processing Application Based on Phonetics and Phonology of the Polish Language,” in Computer Nerworks (Kwiecien, A and Gaj, P and Stera, P, ed.), vol. 79 of Communications in Computer and Information Science, (Germany), pp. 236–244, Springer- Verlag Berlin, 2010. 17th International Conference Computer Networks, Ustron, Poland, Jun 15-19, 2010.

[5] P. Kłosowski, “Improving speech processing based on phonetics and phonology of Polish language,” Przegla˛d Elektrotechniczny, vol. 89, no. 8/2013, pp. 303–307, 2013.

[6] J. Izydorczyk and P. Kłosowski, “Acoustic properties of Polish vowels,” Bulletin of the Polish Academy of Science - Technical Sciences, vol. 47, no. 1, pp. 29–37, 1999.

[7] J. Izydorczyk and P. Kłosowski, “Base acoustic properties of Polish speech,” in International Conference Programable Devices and Systems PDS2001 IFAC Workshop, Gliwice November 22nd - 23rd, 2001, pp. 61– 66, IFAC, 2001.

ABOUT AUTHORS:

ASWINI THOTA is currently pursuing MCA in SVKP & Dr K S Raju Arts & Science College, affiliated to Adikavi Nannaya University, Rajamahendravaram. His research interests include data Science, Machine Learning and Artificial Intelligence.

P. SRINIVASA REDDY is working as a associative professor and SVKP & Dr K S Raju Arts & Science College in Penugonda in AP. He received Master's degree Computer Applications from Andhra University. His research interests operational research, probability and Statistics, Design and Analysis of Algorithm, Bigdata Analytics.