A Review on Neural Language Modeling

Anuar Maratkhan

School of Science and Technology
Nazarbayev University
anuar.maratkhan@nu.edu.kz

***Abstract*—This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.**

I. Introduction

Language Modeling (LM) is a central task in Natural Language Processing (NLP) and play main role in speech recognition, machine translation, optical character recognition, natural language understanding, question answering and many other tasks. Language modeling is all about sequential data. *Language model* is an algorithm for predicting next word in a text given preceeding ones (figure 1). It is also mostly known as *statistical language model* for its richness in statistical approaches in previous works. The state-of-the-art statistical language models turned up to be cache models and class-based models. In addition, according to [1], most of the statistical models require more data for better performance.

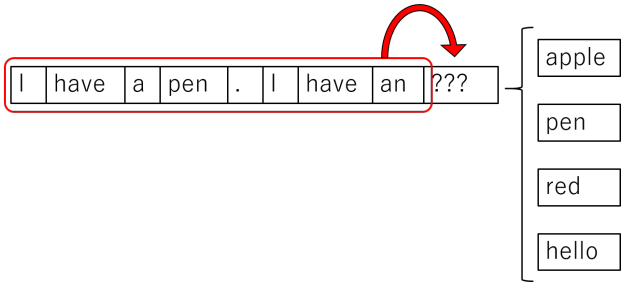

Fig. 1. Next word prediction in a text.

However, as soon as neural networks became popular, *neural language modeling* approaches were invented. Artificial neural networks perform well on supervised learning tasks (when the data is *labeled*) and unsupervised learning tasks (when the data is *unlabeled*). Quoc V. Le et al. in their study [2] explored and demonstrated how well neural networks operate on unsupervised learning task, where a model has to detect whether a given image contains a face or not. And as the study [2] notes, the work has been motivated by neuroscience hypothesis. So does the whole science about neural networks. Moreover, information retrieval tasks that are mainly composed of unlabeled data, like in search engines, can be approached by such unsupervised learning techniques. Neural language models, also known as continous-space language models, make use of these neural networks. Those are current state-of-the-art approaches in language modeling.

In the second section of this paper we start by reviewing Recurrent Neural Networks (RNN). Further, in the same section we review the first/previous and current state-of-the-art RNN based word-level language modeling approaches, starting from [1] and [3], proceeding by [4], and [5]. Next section presents different approaches on neural language modeling that include division of words into subwords presented in [6], reinforcement learning based techniques from [7].

II. Recurrent Neural Networks based Language Models

A. Recurrent Neural Network

Recurrent Neural Networks (RNN) are primarily used in sequential data analysis such as video processing, text processing, predicting stock prices and other. RNN architecture is an improvement of feed-forward networks that takes into account previous hidden layer results by storing them in memory. An invention of backpropogation caused an exciting use of RNNs [8].

Simple recurrent neural network used in [1] can be described by input at time $t - x(t)$, hidden layer at time $t - s(t)$, and the output layer at time $t - y(t)$, which are computed as follows:

$$x(t) = w(t) + s(t-1) \tag{1}$$

$$s_j(t) = f\left(\sum_i x_i(t) u_j i\right) \tag{2}$$

$$y_k(t) = g\left(\sum_j s_j(t) v_k j\right) \tag{3}$$

where (1) uses current word at time t and previous hidden layer at time $t-1$ as input, (2) and (3) have weights u and v used for computations, and both also have sigmoid activation function and softmax function, which are:

$$f(z) = \frac{1}{1+e^{-z}} \tag{4}$$

and

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \tag{5}$$

respectively.

The problem of simple RNN is that the backpropogated gradients vanish or explode [9]. That means that simple recurrent neural network model can not store long context but just several preceeding words. Therefore, the improvement in RNN architecture further was introduced with invention of Long-Short Term Memory (LSTM). The LSTM solved the vanishing

gradient problem by having special cells with activation gates that provided the model to store longer context. Moreover, Gated Recurrent Unit (GRU) improved the architecture of the LSTM by combining two gates that made the model simpler. The study [9] shows that RNNs perform better than complex statistical approaches, N-grams, on a real-world data.

B. Language Models

Neural language models have demonstrated relatively superior performance on language modeling and speech recognition tasks in comparison to the state-of-the-art statistical approaches, class-based models, in the last decades. Neural language models take their first steps from taking the advantage of feed-forward networks that are listed in [1], which motivated T. Mikolov et al. to investigate the performance of more advanced type of neural network, RNN. The study was first to evaluate RNN for modeling such sequential data.

In contrast to feed-forward networks that use fixed size of context to predict next word in a sequence, RNNs does not require to use limited number of preceeding words [1]. In other words, RNNs have dynamic size of context in memory. The basic structure of simple recurrent neural network used in the study is shown in figure 2 below.

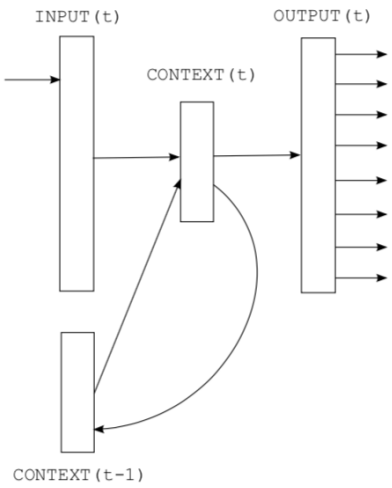

Fig. 2. Simple recurrent neural network from [1]

According to [1] results RNNs illustrated excellent performance relative to best backoff models even being trained on less data than the second ones. The performance in language modeling is measured by *perplexity* which illustrates how well the model can predict next words in a sequence. The lower the perplexity is, the higher the performance. On the other hand, the higher the perplexity, the lower the performance of a particular language model.

Furthermore, [3] work presents that RNN based LM can achieve higher efficiency on language modeling tasks by being trained with backpropogation through time (BPTT) optimization technique, which serves as an extension to standard backpropogation algorithm. This backpropogation algorithm is main optimization technique of neural network models. The backpropogation is basically updating neural network weights (coefficients) iteratively. The key concept of BPTT is that the error propogates back for several time steps through its recurrent connections, as it was stated in [3].

Later, Zaremba et al. were first to implement LSTM model with dropout regularization for language modeling task, and demonstrated another improvements in perplexity [4]. Moreover, subsequent studies listed in [5] show other improvements in perplexity with LSTMs and other techniques on word-level language modeling. Short overview on the work done there illustrated in table I[1].

III. Other Language Models

A. Subword-level Language Modeling

All of the studies discussed before involved word-level language modeling approaches. Word-level language models have particular size of vocabulary that indicates number of *unique* words learned in the process of training. Even though those models show relatively superior performance in terms of lower perplexity and number of parameters, when it comes to new words (during evaluation on the test set) that have not appeared throughout training, the word-level language models are unable to assign nonzero probability of occurence to those unknown Out-of-Vocabulary (OOV) words. As a consequence, the models incapable to use OOV words to predict next word in the sequence. If the OOV words are rare words that occur in the texts not so often, it is not that dangerous. But when the OOV words are not rare and occur often, OOV words become a significant problem to word-level language models. To eliminate this problem, statistical language models have been using *character-level* language modeling techniques, where model trains not on the words but on the individual characters in the words. From those statistical approaches to character-level modeling, neural language models that involved the same character-level modeling emerged.

According to [6] study, the different approach to language modeling tasks, character-level modeling, showed lower results in comparison to word-level language models that engaged simple feed-forward neural network, recurrent neural network, n-discounted n-gram, and maximum entropy n-gram on Penn Treebank and text8 datasets. Besides that, however, because new words are independent of vocabulary in the character-level models, the results have shown zero OOV. Moreover, the study [6] states that the character-level models are hardly trained in order to show high performance, and still even neural network architectures with 1000 neurons on hidden layer (comparatively large hidden layer) are not as good as word-level language models. Therefore, T.Mikolov et al. suggests to use different approach to language modeling, the subword-level model, that can be as efficient as word-level model and as general as character-level language models. The example of conversion of words into subwords is given in figure III-A below:

```
new company dreamworks interactive
new company dre+ am+ wo+ rks:  in+ te+ ra+ cti+ ve:
```

[1]It is out of scope of this paper to provide detailed description of those approaches.

TABLE I
SINGLE MODEL PERPLEXITY ON VALIDATION AND TEST SETS ON PENN TREEBANK [5].

Model	Param	Validation	Test
Mikolov & Zweig (2012) – RNN-LDA + KN-5 + cache	9M	-	92.0
Zaremba et al. (2014) – LSTM	20M	86.2	82.7
Gal & Ghahramani (2016) – Variational LSTM (MC)	20M	-	78.6
Kim et al. (2016) – CharCNN	19M	-	78.9
Merity et al. (2016) – Pointer Sentinel-LSTM	21M	75.7	70.9
Grave et al. (2016) – LSTM + continuous cache pointer	-	-	72.1
Inan et al. (2016) – Tied Variational LSTM + augmenter loss	24M	75.7	73.2
Zilly et al. (2016) – Variational RHN	23M	67.9	65.4
Zoph & Le (2016) – NAS Cell	25M	-	64.0
Melis et al. (2017) – 2-layer skip connection LSTM	24M	60.9	58.3
Merity et al. (2017) – AWD-LSTM w/o finetune	24M	60.7	58.8
Merity et al. (2017) – AWD-LSTM	24M	60.0	57.3
Salakhutdinov et al. [5] – AWD-LSTM-MoS w/o finetune	22M	58.08	55.97
Salakhutdinov et al. [5] – AWD-LSTM-MoS	22M	**56.54**	**54.4**
Merity et al. (2017) – AWD-LSTM + continous cache pointer	24M	53.9	52.8
Krause et al. (2017) – AWD-LSTM + dynamic evaluation	24M	51.6	51.1
Salakhutdinov et al. [5] – AWD-LSTM-MoS + dynamic evaluation	22M	**48.33**	**47.69**

TABLE II
SINGLE MODEL PERPLEXITY ON THE TEST SET OF THE PENN TREEBANK LANGUAGE MODELING TASK [7].

Model	Parameters	Test Perplexity
Neural Architecture Search with base 8	32M	67.9
Neural Architecture Search with base 8 and shared embeddings	25M	64.0
Neural Architecture Search with base 8 and shared embeddings	54M	62.4

Furthermore, the study shows the improvements in performance of language models using subword-level approach instead of character-level. In addition, subword-level language modeling reduces the number of parameters in neural network architecture, which is computed as follows:

$$\#ofparameters = (2 \times V + H) \times H \tag{6}$$

where V – the size of vocabulary, and H – the size of hidden layer.

Thus, the subword-level language modeling is worth considering for usage in language modeling tasks due to its combination of advantages of both word-level and character-level approaches.

B. Reinforcement Learning

Reinforcement learning is an area of deep learning with specific agent and environment where agent is rewarded according to its performance on some task. For instance, B.Zaph and Quoc V. Le in their study [7] present a novel reinforcement learning algorithm that searches best neural network architecture for performing language modeling tasks (refer to figure 3).

As it can be seen from figure 3, the agent gives reward signal to the controller that tries different neural network architectures based on the accuracy result of the neural network. As a result, the neural network architecture achieves higher accuracy performance further. For better understanding, one may imagine a controller parent recurrent neural network that produce child neural network architecture, and gets rewarded on the performance of the child. It is the same as if the hardware engineer (parent) develops a robot (child), and based on the performance of robot on a given task, the engineer gets rewarded.

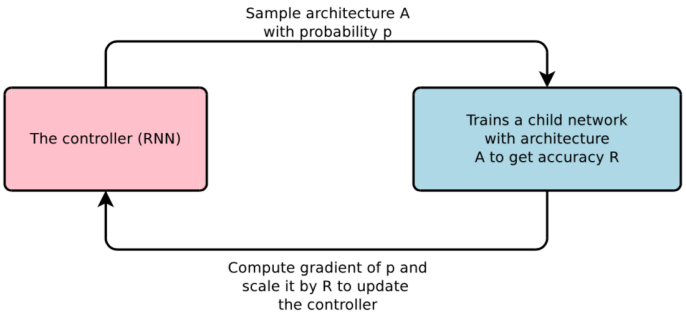

Fig. 3. An overview of Neural Architecture Search from [7]

The algorithm in [7] constructs a tree of architectures which RNN controller then uses. The search space for the controller includes 1) a combination method – *addition*, *multiplication*, and 2) an activation function – *identity*, *tanh*, *sigmoid*, *relu*. The authors of [7] evaluated the performance of their neural architecture searcher on a well-known dataset for language modeling, Penn Treebank. The table II demonstrates the obtained results of [7] work.

Hence, reinforcement learning techniques also can support language models with a guidance on how to achieve better accuracy results.

IV. CONCLUSION

Over the past few years, it was clearly seen that continous-space language models outperformed statistical modeling techniques significantly. The ease of feature learning without need

in extraction of those features simplified lives of scientists. The neural approach offers such opportunity to its users.

Recurrent networks are expected to improve natural language understanding in real-world applications.

To be continued...

REFERENCES

[1] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, vol. 2, 2010.

[2] Q. V. Le, M. A. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Deau, and A. Y. Ng., "Building high-level features using large scale unsupervised learning," in *Proceedings of the 29 th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

[3] T. Mikolov, S. Kombrink, L. Burget, J. Cernocký, and S. Khudanpur, "Extensions of recurrent neural network language model," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5528–5531, 2011.

[4] W. Zaremba, I. Sutskever, , and O. Vinyals, "Recurrent neural network regularization," 2014, arXiv preprint arXiv:1409.2329.

[5] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," 2017, under review as a conference paper at ICLR 2018.

[6] T. Mikolov, I. Sutskever, A. Deoras, H.-S. Le, S. Kombrink, and J. Cernocký, "Subword language modeling with neural networks," 2011.

[7] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2017, arXiv preprint arXiv:1611.01578.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Macmillan Publishers Limited*, vol. 521, pp. 436–444, 2015.

[9] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding reccurent networks," 2015, under review as a conference paper at ICLR 2016.