Comparative Analysis of Next-Word Prediction Technologies Using Machine Learning

Rajiv Ranjan
School of Computing Science and Engineering
Galgotias University
Greater Noida, India
Codmw2382@gmail.com

Harsh Jaiswal
School of Computing Science and Engineering
Galgotias University
Greater Noida, India
harshjaikumar1@gmail.com

Prof. R Sathiya Priya
Asst. Professor
Galgotias Univesrity
Greater Noida, India
sathiya.priya@galgotiasuniversity.edu.in

***Abstract*— This paper abstracts from and analyzes ML models of next-word prediction in the context of NLP. The explored key models consist of RNNs, LSTM network, and Transformer network architectures like BERT and GPT etc. By comparing these model's architecture, training methods and performance metrics, the study finds this to be useful models in next word prediction tasks and understand trade-offs between accuracy, computational cost, and adaptive nature. It is found that although Transformer models are more accurate with the richer contextual understanding, LSTM models offer computational efficiency in the environment with limited resources. Finally, the study provides future directions that will help pushing next word prediction towards hybrid, and optimized models.**

Keywords— Next-word prediction, Recurrent Neural

Networks, LSTM, Transformer, BERT, GPT, Natural

Language Processing, Machine Learning

I. INTRODUCTION

Some subfield of Natural Language Processing (NLP) called next word prediction, has started to take the lead by leaps and bounds with the significance given by mobile computing and conversational AI. The objective of this task is to predict the next word in a text sequence based on previous words, a skill used in current virtual keyboards, search engines, chatbots, and language translation systems. Predicting the next word of the sequence accurately improves user engagement and reduces time consumed in text based interaction by reducing cognitive load which makes communication tool more effective and intuitive. The development of such next word prediction technologies hinges on early probabilistic models which compute the likelihood of a sequence of words followed by a given word based on some observations over large text corpora. Foundationally, n-gram models were based outside of a fixed window of preceding words and had very low contextual awareness. The result was the development of statistical language models and smoothing techniques in cases with sparse data as researched by Chen and Goodman [1]. Yet, they had difficulty handling long-term dependencies, which are required for high quality next word prediction in present day usage.

ML introduced RNNs, which allowed memory to be retained in the form of feedback loops so as to permit sequential dependencies in language modeling. However, RNNs did not effectively solve the problem of retaining information across long sequences, due to the vanishing gradient problems that plagued them [3], and thus were outperformed by traditional statistical methods [1]. Hochreiter and Schmidhuber (1997) developed Long Short Term Memory (LSTM) networks that addressed this problem by a gating mechanism of selective information retention and information forgetting of the whole architecture [4]. Subsequently, further advancements led to the development of Transformer architectures and self attention mechanisms (Vaswani et al., 2017) as presented in "Attention Is All You Need" [5]. Compared with LSTM, a Transformer would have been able to process all sequences at once and was very efficient and effective for complex language tasks. This enabled the model to create a dynamic interpretation of each word within the sequence based on the importance it should have. BERT and GPT were followed by transformer models that marked a great performance in NLP tasks, for instance, next word prediction [6] [7]. One important aspect of this study deals with how the various models perform in predicting the next word. As GPUs and TPUs grow more powerful these models have become available to be trained on ever larger datasets, and with pre trained models common these days broadening to specific tasks. Researchers seek to combine LSTMs' efficiency with Transformers' contextual sophistication in order to deploy efficient next-word prediction on resource restricted devices, such as phones [8], and a trend towards the hybrid architectures which combine those 'efficiency' and 'contextual sophistication', is also observed. This study compares the architectures and capabilities of the traditional n-gram models, RNNs, LSTMs and Transformers and tries to put the current state of the next word prediction technologies. Finally, the study also discusses the computational tradeoffs of each model, especially with the consideration of real world applications where we need to maintain the accuracy at high level and also low latency.

II. LITERATURE SURVEY

Next word prediction is a core of Natural Language Processing (NLP) and has advanced greatly over the period of time. This has given many approaches to improve the accuracy as well as the efficiency. In this literature survey I take a look at the main aspects of this domain, by putting different model architectures through, and what are their restrictions and benefits.

A. Traditional Approaches

Next word prediction with early attempts was based upon statistical methods — particularly ngram models. As shown by Chen and Goodman [1], smoothing still remains crucial to improve robustness of language models, and they demonstrated how to use smoothing techniques to learn from unseen n-grams. However, these methods were unable to handle long range dependencies since there is a preset window that is fixed.

B. *Neural Network*

RNNs came into the limelight with the introduction of deep learning as they were able to process sequential data. Following the successful application of RNNs for language modeling, this, Mikolov et al. [2] proposed to use RNNs for language modeling, to greatly improve the prediction accuracy due to contextual information. Nevertheless, RNNs have been faced with difficulties of capturing long range dependencies due to the vanishing gradient problem; thus, LSTM networks of Hochreiter and Schmidhuber [3] were introduced. Information spanning the longer sequences was successfully retained by LSTMs improving the performance of the next word prediction tasks.

C. *Attention Mechanisms*

Attention was a key part of a change in paradigm for NLP. Transformer architecture [4] has been introduced, and self attention is used to process input sequences in parallel. By allowing the model to focus on different parts of the input sequence, this model has achieved state of the art results on various NLP tasks including next word prediction, because the model was not restricted to the order of data processing.

D. *Pre-Trained Language Models*

Recently, most of the work has been centered around pre trained language models that are then fine tuned to a specific task. Continuous BERT [4] extends BERT, which proposed by Devlin et al. [5], by the addition of continuous token tags and proceeds with bidirectional training to capture context from both directions, resulting in improved masked language tasks. In the same way, GPT models produce text with great coherence and contextual relevance in the next words by using an autoregressive scheme [6].

E. *Hybrid Approaches*

Recently, the hybrid models have been used integrating various architectures for enhanced performance. Yang et al. [7] first introduce XLNet language model, in which the autoregressive pretraining is generalized to cope with bidirectional context without having to be masked. Additionally, Wu et al. [8] used knowledge based methods to integrate with deep learning with the purpose of improving the contextual understanding in model.

F. *Current Trends*

Models with ever increasing sizes of 175 billion parameters are developed which show few shot learning capabilities [9]. However, these models do set new benchmarks in next-word prediction tasks, but at the same time it suggests that they raise issues regarding computational efficiency and related ethical issues of AI deployment in real world applications.

III. METHODOLOGY

In general, the development and evaluation of next-word prediction systems has several phases such as data collection and preprocessing, model training and evaluation. And this section details that through the architecture and optimization strategy, the architecture and optimization strategy can achieve high accuracy and efficiency for this.

A. *Data Collection and Preprocessing*

1. Dataset Selection

   The quality and size of the training data is crucial, in fact, as to the accuracy of next word prediction. Commonly used datasets for training language models include Wikipedia text dumps, Common Crawl datasets, BookCorpus, OpenWebText, as well as use cases proprietary app logs that actually capture real user input. In this way, these datasets encompass an abundant variety of language, such as the formal text to informal user generated content. Since I had a combined corpus across sources to capture diverse linguistic patterns and improve the model's capability to generalize, I used a corpus for this study.

2. Data processing and Tokenization

   First of all, the noise including HTML tags, special char, and unnecessary symbols are removed from the raw data. After that, the tokenization phase decays the text into single words or subword units. Subword tokenization techniques such as BPE or WordPiece do not require the model to recover out-of-vocabulary or unique words as a sequence of common subwords, improving efficiency and coverage.

3. Data Segmentation

   It performs data segmentation (dividing the text in sequences of fixed number of tokens) as each model take a fix number of tokens when they are used as input. For instance, in BERT and GPT models, sequences are used like 128 or 512 tokens. Consistency of the sequence length across the entire dataset is maintained by padding and truncation.

B. *Model Architecture*

1. N-Gram Models

   n-gram models would be the simplest form of the language model, where it canspecifies the next word based on the probability of the group of n-words in the dataset. Unfortunately, the n-gram model is computationally light weight, but it is not able to deal with long range dependencies as it only considers a fixed window size. However, low frequency word combinations are smoothed using techniques such as Kneser-Ney smoothing in order to improve the model's ability to predict infrequent words [1].

2. Recurrent Neural Networks

   Sequential processing is introduced by RNNs, where each output depends not only on the current input but also on previous states, and hence, an RNN can learn contextual

dependencies. This feedback loop feature allows RNN to learn how to capture patterns over time; but has a problem of stability issue due to long sequences. Processed by processing sequence of input tokens one at a time and predicting the next word based on accumulated context till the current word, the next word prediction task is performed.

3. Long-Short-Term Memory

   RNNs had limitation and its limitation are overcome by developed LSTMs. As it passes, information is governed by the gates to be proceeded, and long term dependencies are retained. The forget gate rules how old information is disregarded, the input gate decides which new information is incorporated into cell state and the output gate dictates an output based on present input and memory. Then this gating mechanism allows the LSTM to have the ability to remember for extended sequences for next word prediction with improved accuracy for longer text [4].

4. Transformer Model

   As first introduced by Vaswani et al. [5], Transformers do so by handling dependencies between all tokens in a sequence as a whole using selfish mechanisms, allowing all the training to be highly parallel. Such attention mechanism assigns a weight to each token of the group when compared to other tokens, so that the model can pay attention to specific parts of the text, depending on the case. Later transformer based models such as BERT, GPT and XLNet work on perfect results in the next word prediction due to its process text with multiple attention layers.

5. GPT

   GPT is the unidirectional autoregressive model, it generates the next word in the left context. In contrast to BERT, GPT was built for text generation, and hence it is more suitable for next word prediction. On inference, GPT will predict a word in a rolling manner, i.e., predicts a word and uses this word as part of the input to predict the next word [6].

C. Training Procedure

1. Objective Function and Optimization

   Generally speaking, the cross entropy loss is used for next word prediction as we penalize the incorrect prediction and reward correct one by minimizing the difference between the given probabilities and actual labels. It is often that the Adam optimizer is used, since it is a combination of momentum and adaptive learning rates, to make the convergence very fast [11].

2. Training Strategies

   To take the high computational demands into consideration, the models are trained on a distributed computing environment by means of data parallelism or model parallelism. In order to improve model robustness and to mitigate overfitting, data augmentation techniques such as random masking, shuffling, data subsampeling are used.

3. Hyperparameter Training

   Models are fine-tuned with the hyperparameters such as learning rate, batch size, sequence length, etc. based on a joint of grid search and random search strategy in order to maximize the model performance. To avoid overfitting such as used in the large models, such as Transformers that tend to memorize the training data, regularization techniques like dropout and weight decay are employed.

D. Evaluation Metrices

1. Perplexity

   The perplexity is a metric to evaluate language models, as it records the uncertain of the model for making the next predicted word / word bag in the stream. The lower is the perplexity score, the more confident the model predicts it is and hence at a better performance.

2. Accuracy and BLEU Score

   As you can guess, accuracy is the percentage of correct next_word prediction, which indirectly measures the model's performance. To evaluate files generated text, BLEU score is used to compare the generated sequence to a reference sequence and hence it can be used to evaluate the fluency and contextual accuracy part in next prediction and the score depending on how close generated sequence is to reference sequence.

3. Latency and Efficiency

   Inference latency is important for actual time use. Model used to predict the next word is measured for the average time used for the prediction and computational resource usage. Especially in resource limited environments such as mobile devices, lightweight models or quantized versions are evaluated to meet accuracy vs. efficiency balance.

IV. TECHNOLOGY USED

The hardware and software technologies for the implementation of next word prediction are based on multiple layers, which involve training, deploying, and real time running of language models. As specialized software libraries, computational frameworks, and hardware accelerators have been created for these NLP tasks and the sizes of corresponding datasets have been growing in size, hence.

A. Software Libraries and Framework

1. **TensorFlow:** It's a framework for numerical computationbased on data flowgraphs and is used for training and deploying of deep learning models

mostly. TensorFlow integrates the Keras API making a model building the process easy for researchers to build complex RNN, LSTM, and Transformer models with modular layers [8].

2. **PyTorch:** Thanks to its dynamic computation graph, PyTorch is quite flexible and is very suitable for research. One thing that really makes PyTorch especially friendly for it is the fact that it integrates with existing deep learning libraries, like Hugging Face Transformers for example, so that is something that is already there for NLP research or applications like next word prediction. This support for fine-tuning and adaptation to new tasks is possible with the framework's support for Transformer models. [9]
3. **Hugging Face Transformers:** However, to support NLP applications, this library had become an essential library to use, providing various pre trained models like BERT, GPT and DistilBERT which can be directly plugged in to solve a wide range of problems including next word prediction. Since the Hugging Face model repository and its transformers library allow easy manipulation, transfer learning, and fine tuning of state of the art models, one can easily deploy models intended for specific applications [10].

B. *Hardware Accelerators*

Deploying and training modern language models requires significant computing resources — thus we make use of the hardware accelerators and exploit the performance of high-performance architectures, especially for the architectures based on the Transformer.

1. **GPUs:** Parallel computation is essential for matrix multiplications in deep learning models, and GPUs have been essential in the advancement of NLP for this reason. Together with GPU optimized TensorFlow and PyTorch libraries, the NVIDIA CUDA framework turned GPUs into the normal way of training deep neural networks, such as LSTMs and Transformers, for next word prediction.
2. **TPUs:** TPUs are specially designed to run fast ML workloads, and are optimized by Google for matrix operations. Transformers are compute heavy activities, well suited for TPU. Researchers can use TPU for training large language models such as BERT, GPT but much faster than would be possible on GPUs. As a result, the pre training of large models on large datasets was enabled with these hardware.

C. *Cloud and Distributed Computing*

1. **Google Colab and AWS Sagemaker:** You can model on the cloud platforms such as Google Colab and AWS Sagemaker. Google colab offers free GPU and TPU resources which serves well for the academic research since they use it widely while AWS Sagemaker provides the ability to scale the training and deployment on dedicated infrastructure, which leads to a more robust implementation of next word prediction models.
2. **Distributed Computing in Model Training:** The datasets that modern NLP models rely on are often very large, and such models need to be trained on them, which requires a distributed computing setup. In preprocessing, techniques like data and model parallelism remove large parts of compute away from GPUs or TPUs toward a bottleneck share of them, speeding up training times. Due to the distributed computing environment, each node in the cluster contributes to different parts of the model's training to improve the model's performance and efficient processing of large datasets [12].

D. *Model deployment and Real-Time Inference*

When they are trained, next word prediction models have to be optimized for deployment in real-time applications, having low latency is important.

1. Techniques in Model: Model quantization and Pruning techniques are normally used to decrease model size and increase inference speed. Both quantization and pruning reduce the accuracy of the model's weights and thus provide a similar but more lightweight model for mobile and embedded devices to run efficiently [13].
2. Edge computing: Very much off device, edge computing helps deal with application that need dealing with on device, which diminishes latency and network utilization. Hardware advance such as Google's Edge TPU makes it possible to deploy next word prediction models in mobile devices, no need of cloud resources, to allow for real time interaction.

V. EXPERIMENTAL RESULT

Furthermore, the evaluation of next-word prediction model relies on a set of experiments to benchmark their performance across various measures. This section gives a summary of the experimental results regarding different models, and presents some key results as well as the performance comparison.

A. *Experimental Setup*

A combined dataset of text of Wikipedia, Common Crawl as well as other publicly available corpora was used for training models. To reduce the variance caused by the differences in model input, a standardized preprocessing pipeline was applied to tokenization and segmenting of the data. Accuracy, perplexity, and inference latent were used to evaluate the models.

B. *Performance Metrics*

1. Lower values of perplexity equate to high model performance. With a perplexity score of 54.7, the LSTM model is superior, but the Transformer claims to have a perplexity of only 35.2, indicating the ability for the latter to capture language structure.

2. Next word prediction accuracy: The next word prediction accuracy was evaluated on a test set. When BERT was fine tuned to predict the next word its accuracy was found to be 88.4 while the GPT-2 model reported 89.7. The goodness of fit of the autoregressive models was slightly better than that of masked models [10].
3. Measured were the average time it took for models to infer the next word. Out of these two models, LSTM was found to be 32ms and Transformer was 25ms on average. This however was still not fast enough to be latency feasible in real time application (but was for more than 150ms average latency) [11]

VI. CONCLUSION

So the future of next word prediction models is completely revolutionized in Natural Language Processing — it goes away from the traditional statistical ways to the complicated architectures of the deep learning. This research has described the evolution through a number of models from ones that are less accurate, less contextual and not able to generate but all the way up to models that are seen as very accurate, very contextual, and able to generate. We provide results that confirm that traditional approaches and earlier neural models are no match for Transformer based models that have been pre trained and use attention mechanisms. Although, models such as GPT-3 are showed impressive performance, latency and the challenge of resource requirements needs to be tackled for deploying GPT-3 in real time applications.

VII. FUTURE WORK

This is promising and many things to explore in the future of next word prediction:

1. Model Optimization: Such techniques as knowledge distillation, quantization and pruning are needed to further increase the efficiency of the models. One of these approaches can lower the computational footprint of large models without incurring much performance cost.
2. Multimodal Data: Naturally, it is possible to incorporate multimodal data in our models, for instance, images or audio which can help with a richer context of what a model is predicting and accommodate for a much more nuance, contextually aware prediction.
3. Opportunity: Personalizing user preferences and linguistic styles via giving models that adapt to the individual user preference and linguistic style presents a great opportunity to improve user experience when using applications such as chatbots and predictive text inputs.
4. Ethical aspects: The bigger the language models, the more it will be necessary to take care of ethical problems (bias, misinformation, privacy). Future deployments will require research into mechanisms for the responsible AI usage.
5. Future Works: Future works should explore cross linguistic capabilities when predicting the next word in different languages, particularly for languages that are not common in NLP. There would be more inclusive AI applications from models that can generalize well across languages.

In the end, next word prediction landscape is still romantic, and there is much potential for new breakthrough that may fundamentally change how we interact with language technology using our daily language.

VIII. REFERENCES

[1] C. Chen and E. Goodman, “An Empirical Study of Smoothing Techniques for Language Modeling,” in Proceedings of the 1st International Conference on Natural Language Processing, 1996.

[2] T. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems, vol. 33, 2020.

[3] M. Johnson et al., “Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation,” Transactions of the Association for Computational Linguistics, vol. 5, pp. 339-351, 2017.

[4] Y. LeCun et al., “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[5] Z. Cheng et al., “Quantized Convolutional Neural Networks for Mobile Devices,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.

[6] T. Mikolov et al., “Recurrent Neural Network Based Language Model,” in Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010.

[7] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[8] A. Vaswani et al., “Attention is All You Need,” in Advances in Neural Information Processing Systems, vol. 30, 2017.

[9] J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv preprint arXiv:1810.04805, 2018.

[10] A. Gulli and S. Pal, Deep Learning with Keras, Packt Publishing Ltd., 2017. 11. A. Radford et al., “Language Models are Unsupervised Multitask Learners,” OpenAI, 2019.