NukeBERT: A Pre-trained language model for Low Resource Nuclear Domain

A. Jain*, Dr. Meenachi Ganesamoorty[†]

*BITS PILANI, Pilani, India, [†]IGCAR, Kalpakkam, India

*f20170093@pilani.bits-pilani.ac.in, [†]meenachi@igcar.gov.in

Abstract—Significant advances have been made in recent years on Natural Language Processing with machines surpassing human performance in many tasks, including but not limited to Question Answering. The majority of deep learning methods for Question Answering targets domains with large datasets and highly matured literature. The area of Nuclear and Atomic energy has largely remained unexplored in exploiting non-annotated data for driving industry viable applications. Due to lack of dataset, a new dataset was created from the 7000 research papers on nuclear domain. This paper contributes to research in understanding nuclear domain knowledge which is then evaluated on Nuclear Question Answering Dataset (NQuAD) created by nuclear domain experts as part of this research. NQuAD contains 612 questions developed on 181 paragraphs randomly selected from the IGCAR research paper corpus. In this paper, the Nuclear Bidirectional Encoder Representational Transformers (NukeBERT) is proposed, which incorporates a novel technique for building BERT vocabulary to make it suitable for tasks with less training data. The experiments evaluated on NQuAD revealed that NukeBERT was able to outperform BERT significantly, thus validating the adopted methodology. Training NukeBERT is computationally expensive and hence we will be open-sourcing the NukeBERT pretrained weights and NQuAD for fostering further research work in the nuclear domain.

Index Terms—Natural Language Processing, Question Answering, Bidirectional Representational Transformers, SQuAD, Nuclear, Pretraining, Fine-tuning.

I. INTRODUCTION

The nuclear industry presents a viable solution to end energy crisis. Advancements in machine and deep learning plays a significant role in the medical domain ([Deo, Rahul C., 2016; Shaker El-Sappagha, 2015; Christian Seebode et al., 2016; David McDonald, 2016 [1,2, 3]) and a lot of groundwork is already done in generation of various datasets and pretrained models, thus reducing the barrier for future research. Similar advancements needs to be done in the nuclear field. A survey indicated that limited effort has gone into the domains of power plants and atomic energy (N. Madurai Meenachi and M. Sai Baba, 2012 [4]). This research aims to explore the applicability of BERT (Bidirectional Encoder Representations from Transformers) (Devlin, Jacob, et al., 2018 [5]) in the nuclear field, which lacks high quality data.

On this ground, a model is developed in the nuclear field for machine learning which will be referred to as Nuclear Bidirectional Encoder Representations from Transformers for language understanding (NukeBERT) (Figure 1). NukeBERT is a contextualized nuclear word embedding, based on BERT, which can be fine-tuned further on eleven downstream tasks including Question Answering, Named-entity recognition and

Sentence segmentation. BERT is trained on wikipedia corpus, which is strikingly different and lacking in nuclear terminologies. Further, BERT requires a huge corpus for generating word embeddings, which is difficult to build in a low resource nuclear domain. Hence we developed a custom vocabulary, "NukeVocab", for training BERT. Thus formed model is then evaluated for its performance on Question Answering task. Due to lack of question-answering dataset in the nuclear domain, we developed a new dataset, called NQuAD (Nuclear Question Answering Dataset), prepared with the help of nuclear domain experts at IGCAR as part of this research.
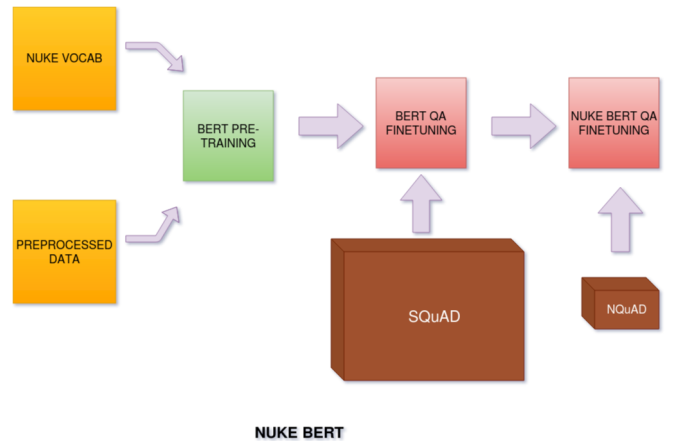

Fig. 1: NukeBERT: From raw data to the fine-tuned question-answering system.

With the same configuration as used by original BERT for fine-tuning, NukeBERT is fine-tuned on Stanford Question Answering Dataset (SQuAD) (Rajpurkar, Pranav, et al., 2016 [6]) for two epochs. It took around 8 hours to train on Google Cloud GPU. Then NukeBERT is further fine tuned on NQuAD train set for three epochs at a learning rate of 3e-6. On testing, the NukeBERT achieved F1 score of 93.87 and exact match score of 88.31. Hence the model was able to achieve 5.21 improvement on exact match criteria and 1.22 improvement on F1 score compared to BERT model.

For fostering future research and industrial applications, we will be open-sourcing NQuAD and pretrained weights.

II. RELATED WORK

Word2Vec pioneered the semi-supervised approach to read massive corpora and generate meaningful word embeddings.

However, vanilla Word2Vec suffers massively from polysemy and inability to effectively deal with out of vocabulary words (Iacobacci et al., 2015, Reisinger and Mooney, 2010 [7, 8])

The advent of ELMO (Embeddings from Language Models) (Peters, Matthew E., et al., 2018, [9]), GPT (Generative Pre-Training Model) (Radford et al., 2018, [10]) and BERT demonstrated the benefit of unsupervised training on large corpus for other sophisticated downstream tasks. BERT was trained on 3.3 Billion words dataset to generate BERT embeddings, which were then fine-tuned to various downstream tasks. With minimal architecture modification, BERT was able to achieve a state of the art in 11 natural language processing tasks.

SciBERT trained on 1.14 Million papers from a semantic scholar having 3.17 Billion tokens. It was able to improve the model accuracy on several scientific and biomedical datasets than BERT (Beltagy et al., 2019, [11]). BioBERT trained effectively on 18 Billion words dataset of medical corpus and 3 Billion words BERT dataset. BioBERT could significantly improve the state-of-the-art performance (Lee, Jinhyuk, et al., 2019, [12]).

In contrast to BERT, SciBERT in science and BioBERT in biology respectively, nuclear domain suffers from a significantly smaller dataset. After preprocessing and cleaning, 8 Million words dataset was generated, which is two magnitudes lower than dataset used by them. Hence a novel approach to optimize the formation of Nuclear Vocab (NukeVOCAB) and pre-training to achieve better results than BERT is developed. Unlike the medical domain, the nuclear domain doesn't have any gold training dataset to gauge the performance of this newly developed model. Hence a Nuclear Question Answering Dataset (NQuAD) is developed with the effort of Nuclear scientists. The domain experts have evaluated the system for its performance. The NukeBERT methodology is explained in the following section.

III. METHODOLOGY

A. Corpus Preparation

For the dataset preparation, 7000 internal reports, thesis and research papers from the Indira Gandhi Centre for Atomic Research (IGCAR) in the PDF format are taken for analysis. The sizes of the reports ranged from a couple of pages to a few thousand pages. Apart from this, a substantial portion of nuclear corpus consisted of very old reports, some of which were handwritten and hence stored as scanned copies. The reports primarily dealt with the nuclear domain, many of them explicitly dealing with FBR (Fast Breeder Reactor). The raw corpus needed cleaning and preprocessing to convert it into pre-training corpus suitable for BERT.

B. PREPROCESSING

The detailed pipeline for preprocessing is shown in Figure 2. After cleaning and investigating the dataset manually, it was decided to use OCR for extracting the text. Every page in each PDF is converted to an image using the pdf2image library. The corpus was a mixture of regular PDFs and the scanned
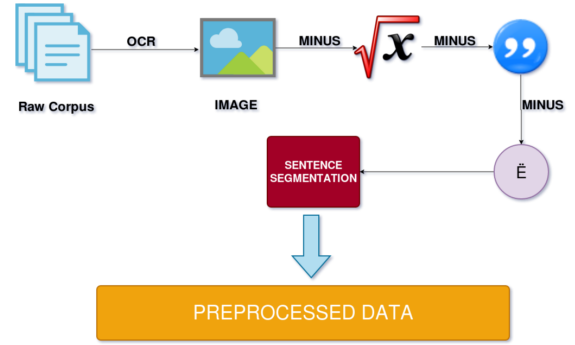

Fig. 2: Flowchart of steps for preparation of preprocessed data from raw corpus of 7000 research papers.

PDFs images. Regular PDF readers fail in reading scanned images. Hence all PDFs were converted into images to avoid manual separation of the PDFs. To avoid random Unicode errors in normally reading PDFs, which is unavoidable with the diversity of the PDFs in the corpus used.

Tesseract-Optical Character Recognition (OCR) (Smith et al, 2007, [13]) is used to read the text from the images. The OCR is not accurate and generates specific errors like reading 'o' as '0', 'I' as '1'. However, according to Namsyl et al. (2019) [14]. This can even be a useful data augmentation technique. Also, it is not a big issue because BERT tokenizer will be tokenizing the errored word by splitting it about the wrongly comprehended digit, hence would still be able to understand a meaningful representation of the word.

OCR is computationally expensive, and its accuracy and processing time highly depends upon the quality of the image. Our corpus has varied quality PDFs, with some being very old, handwritten and misaligned. It took approximately four days of continuous running on a single Google Cloud K80 Tesla GPU. The average time taken per page was between 20-25 sec.

There were many mathematical formulas in the research papers and thesis. We tried various regex patterns to remove them, but first of all, not all formulas were getting removed and secondly, there were a lot of stray numerals due to OCR reading diagrams and tables which were cluttering the data. Hence, we removed all the digits from the corpus.

There were a lot of references, both in-text and at the end. For removing them, we designed regex patterns for common styles of in-text citations used in the literature. We noticed that last 2-3 pages generally contain references, and hence we removed last two pages from each corpus.

Since the original BERT model is trained on general English corpus, having foreign text wouldn't be suitable for the use case. The corpus had some research papers and thesis in German and French (which will be referred to as foreign text). There are some libraries like NLTK for checking for foreign words. However, these won't be suitable for proposed use case as it uses its vocabulary and it might wrongly consider some nuclear jargon to be an unfamiliar word, thereby defeating
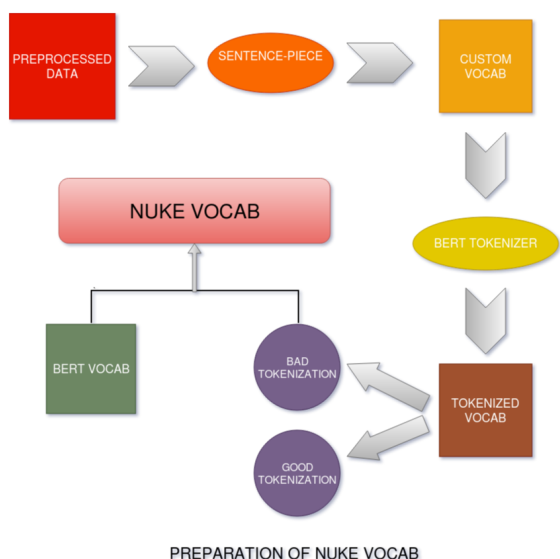
Fig. 3: Flowchart for preparation of NukeVocab. Notice we only concatenate "Bad Tokenization" with BERT Vocabulary.

original_word	tokenized_by_bert
coolant	
	['cool', '##ant']
irradiation	
	['ir', '##rad', '##iation']
reactivity	
	['react', '##ivity']
eee	
	['ee', '##e']
weld	
	['we', '##ld']

Custom Vocab

corrosion
function
should
level
density
because
per
concentration
distribution

(a) BERT tokenization of Custom-Vocab

(b) Custom Vocab generated using sentence piece

Fig. 4: Custom Vocabulary

the purpose. We noticed that some foreign texts typically contained some characters like ë, è, é. Since ASCII representation only contains numbers and English alphabets, and not these foreign characters which are provided in UTF-8 representation used by the text corpus, we ignored those lines which included these foreign characters and thus removed about 20000 foreign lines from the corpus. Finally, we performed a bit of manual cleaning by traversing the data manually.

The input for BERT pre-training is a text file with one sentence per line. An empty line delimits the documents. We used the gensim library for segmenting the text into sentences, which finally makes the input text file ready for pre-training and other tasks. After all the cleaning and preprocessing, we got the domain-specific corpus of 8 Million words.

C. NukeVOCAB

BERT makes use of WordPiece to build their vocabulary on their general words corpus. BERT uses a vocabulary of thirty two thousand words to build the word embeddings. Since the nuclear field contains jargons and technical words, it is essential to have a vocabulary tailored to the nuclear domain, which will be referred as NukeVocab (Nuclear Vocabulary). The procedure for constructing NukeVocab is summarized in Figure 3. Sentence Piece library by tensor2tensor, is used to generate a vocabulary similar to BERT which will be referred as Custom-Vocab (Figure 4). The preprocessed data was given as input to the library, which generated a vocabulary of essential words and sub-words in an unsupervised manner. We modified the output of sentence piece to make it similar to BERT vocabulary. We set the size of Custom Vocab to 32 thousand, identical to the BERT vocabulary.

Hugging Face's Pytorch implementation of BERT tokenizer is then used to tokenize the Custom Vocab . The BERT tokenizer breaks the words into tokens, with each token being a part of the BERT vocabulary. If a word is part of BERT vocabulary, it does not get broken into tokens. Otherwise, the words are tokenized, with each subword, except the first subword, being prefixed by #. The tokenization of Custom-Vocab is shown by Figure 4. All those words which were retained as complete words were removed while the rest were stored in a CSV file with two columns: one with the original word in Custom Vocab and second with the tokenized form by BERT. Around 17 thousand words were tokenized by BERT indicating about 47% overlap with BERT vocabulary. Intuitively, this process segregates those words which need special attention, as these words are a set of words which are considered important by Sentence Piece Library (and hence put in Custom-Vocab) and are not part of BERT Vocabulary.

With the help of domain experts, all the words were segregated into two groups: "good" and "bad". "Good" meant that the tokenization is either successfully extracting the root word or can break into words which are close to their actual meaning (Figure 5) and hence with pre-training on the domain-specific corpus, they will converge to their proper embedding. Bad meant that BERT is not even close to tokenizing these words and could create problems while performing downstream tasks (Figure 6)

By manually iterating over "bad words", those words were clubbed which were having same/similar root words. For example: 'lubricant,' 'lubrication,' lubricated' were replaced by a single word 'lubric.' This approach is vital as it effectively increases the probability of seeing more data and hence would help in learning a more meaningful representation. In this way, 429 words were selected from the "bad" category. Bert Vocabulary contains about 1000 "UNUSED" tokens. 429 "UNUSED" tokens were replaced with the selected 429 words from "bad" category, hence forming NukeVOCAB.

While performing the above operations, few notable observations were:

i) There were some non-English (Russian, German) words in the vocabulary, indicating that there were some portions of Russian and German texts in the corpus.

Words in "Good"	Split by BERT Tokenizer
electrochemical	['electro', '##chemical']
conductivity	['conduct', '##ivity']
neutrons	['neutron', '##s']
ultrasonic	['ultra', '##sonic']
shutdown	['shut', '##down']
exchanger	['exchange', '##r']

Fig. 5: List taken from "Good" Words.

Words in "Bad"	Split by BERT Tokenizer
lethargy	['let', '##har', '##gy']
lubricant	['lu', '##bri', '##can', '##t']
lubricated	['lu', '##bri', '##cated']
lubrication	['lu', '##bri', '##cation']
luminescence	['lu', '##mine', '##sc', '##ence']
machining	['mach', '##ining']

Fig. 6: List taken from "bad" words. Notice the several words with root "lubric".'

ii) Some words were combined without space like 'twodimensional'. These could be attributed to OCR errors, but as expected, BERT tokenizer was successfully able to split them, which makes BERT a suitable algorithm for domains like us having tough datasets.

IV. NQUAD: NUCLEAR QUESTION ANSWERING DATASET

Question-Answering is a crucial part of human conversation, and hence the NukeBERT was decided to be evaluated on Question Answering task. Unlike medical and general domain, the nuclear domain does not have any open source Question Answering Dataset. Such dataset is essential for the research community in this field to check the value of their models. Apart from checking the correctness, it is also useful for numerous applications. Hence it was decided to generate a new, high-quality Question Answering dataset.

For some time, we explored the idea of using the automatic question generation. We came across a few research papers [16, 17, 18], which takes a paragraph as input and generates general questions. There are some rule-based methods, which take a paragraph and create questions.

The important reasons for deciding against using automatic question generation are:

i. Our dataset comprising research papers related to the nuclear domain turns out to be too difficult for the existing question generation systems. Due to the complex language, jargons, and complex sentences, the models were not able to generate meaningful and useful sentences.

ii. The question generation systems which use Named Entity Recognition (NER) (Nadeau, David et al., 2007, [19]) to generate questions are generally ruled based which forms questions replacing for, e.g. nouns with a question having that noun as an answer. These types of questions doesn't require semantic understanding and is rather mechanical.

The question-answering dataset is built with a format similar to SQuAD dataset. For that, research papers were randomly selected out of the 7000 research papers corpus. From these research papers, around 200 paragraphs were randomly selected to form the questions on. Around 50 paragraphs were distributed to each domain expert, asking them to create questions on the paragraphs. They were encouraged to develop questions in their own words, the only restriction being that the answer must exactly lie within the paragraph as followed by SQuAD. The advised answer length was 8-10 words, though, in the final dataset, some accepted answers were longer than that also for generalization purposes. Experts found some paragraphs not suitable for forming questions, and hence, those were discarded. Finally, we were able to generate 612 questions on 181 paragraphs (Figure 7).

The paragraph-questions-answers were recorded in shared google docs, responses from which were merged into an excel file. The file was randomly divided into two sections: a train set of 155 paragraphs, 536 questions, and dev set of 26 paragraphs, 76 questions.

To fine-tune the question-answering platform after training on SQuAD dataset, the custom-made Paragraph-Question-Answer dataset was converted to JSON with the structure similar to SQuAD v1. For turning it into SQuAD format, an already available platform for question generation was adapted for generating the NQuAD dataset. It is built using Angular as frontend with a spring boot backend using MongoDB database.

Since the dev set requires three answers per question, domain experts were asked to answer each of the 100 questions individually. The generated excel was again was converted into JSON format using the question generation platform described above.

Fig. 7: Sample Paragraph-Questions-Answer triplets from NQuAD. Notice that the answers of the questions are strictly within the paragraphs similar to SQuAD.

(a) F1 scores   inin   (b) Exact Match

Fig. 8: Results of BERT and NukeBERT on NQuAD test-set

V. EXPERIMENT

A. BERT PRETRAINING

Pre-Training BERT is a computationally expensive job and requires a large amount of data to pretrain. Our corpus of 8 Million words is two orders of magnitude less than the 3 Billion words corpus used by original BERT. It would be infeasible to pretrain BERT from scratch on this small corpus. Also, as mentioned earlier, there was almost 47% overlap with the BERT vocab, which have already converged to proper representation in the pretrained weights. Pretraining from scratch would mean throwing away those learned embeddings. Pre-training from scratch becomes infeasible for domains like Nuclear, where the amount of data is pretty less. To tackle this issue, we used Nuclear Vocabulary (NukeVOCAB) in place of BERT vocabulary and pretrained it on the custom preprocessed corpus starting from BERT checkpoint. Due to Out of Memory issues, we used the batch size of 128 with maximum length set at 128. We pretrained it till the training loss more or less stopped decreasing, which happened after approximately 600000 steps. It took around 54 hours on a single Google Colab TPU for pre-training BERT. We will be referring this model as NukeBERT for further discussions.

B. BERT Fine-Tuning

For the base model, BERT was fine-tuned on SQuAD using the same configurations used by original BERT paper. On testing on NQuAD dev set, it scored 83.11 on exact match score and 92.66 on F1 score. The fine-tuning took around 8 hours on Google Cloud's Tesla k80 GPU (It is the version of GPU provided for free by google colab).

With the same configuration as used by original BERT for fine-tuning, NukeBERT was fine-tuned on SQuAD dataset for two epochs. It took around 8 hours to train on Google Cloud GPU.

Further NukeBERT was fine-tuned on the NQuAD train set for three epochs at a learning rate of 3e-6.

VI. RESULTS

On testing, the NukeBERT achieved F1 score of 93.87 and exact match score of 88.31. Hence the model was able to achieve 5.21 improvement on exact match criteria and 1.22 improvement on F1 score (Figure 8).

VII. CONCLUSION

BERT requires large data for pre-training and generating meaningful word embeddings. However, the availability of data becomes a bottleneck for domains like nuclear whose data is profoundly different from world language. Hence, for areas like Nuclear energy, it becomes essential to use methods which we used for NukeVOCAB generation to optimize the use of meagre data. The improvement in F1 and exact match score validates the NukeBERT embeddings, which can be used for other downstream tasks too like sentence classification, named entity recognition among many NLP tasks. NukeBERT can be further generalized to many downstream tasks like named entity recognition, sentence segmentation among many, which could be highly important for nuclear industry. Moving further, the ideas used in this paper can be utilised with new transformers that have come up in recent research. Data availability and creation in nuclear field, needs attention. This paper aims at providing momentum to research in nuclear domain.

REFERENCES

1 Deo, Rahul C. "Machine learning in medicine." Circulation 132.20 (2015): 1920-1930.

2 El-Sappagh, Shaker, Mohammed Elmogy, and A. M. Riad. "A fuzzy-ontology-oriented case-based reasoning framework for semantic diabetes diagnosis."Artificial intelligence in medicine 65.3 (2015): 179-208.

3 Seebode, Christian, et al. "Biobank Semantic Information Management With The Health Intelligence Platform." Diagnostic Pathology 1.8 (2016).

4 International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 4– No.2, September 2012 – www.ijais.org 46 A Survey on the usage of Ontology in Different Domains N. Madurai Meenachi & M. Sai Baba

5 Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

6 Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).

7 Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli. "Sensembed: Learning sense embeddings for word and relational similarity." Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015.

8 Reisinger, Joseph, and Raymond J. Mooney. "Multi-prototype vector-space models of word meaning." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010.

9 Peters, Matthew E., et al., "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

10 Radford, Alec, et al. "Improving language understanding by generative pre-training." URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018).

11 Beltagy, Iz, Arman Cohan, and Kyle Lo. "Scibert: Pretrained contextualized embeddings for scientific text." arXiv preprint arXiv:1903.10676 (2019).

12 Lee, Jinhyuk, et al. "Biobert: pre-trained biomedical language representation model for biomedical text mining." arXiv preprint arXiv:1901.08746 (2019).

13 Smith, Ray. "An overview of the Tesseract OCR engine." Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). Vol. 2. IEEE, 2007.

14 Namysl, Marcin, and Iuliu Konya. "Efficient, Lexicon-Free OCR using Deep Learning." arXiv preprint arXiv:1906.01969 (2019).

15 Machinery, Computing. "Computing machinery and intelligence-AM Turing." Mind 59.236 (1950): 433.

16 Heilman, Michael. "Automatic factual question generation from text." Language Technologies Institute School of Computer Science Carnegie Mellon University 195 (2011).

17 Aldabe, Itziar, et al. "Arikiturri: an automatic question generator based on corpora and NLP techniques." International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, 2006.

18 Rakangor, Sheetal, and Y. Ghodasara. "Literature review of automatic question generation systems." International Journal of Scientific and Research Publications 5.1 (2015)

19 Nadeau, David, and Satoshi Sekine. "A survey of named entity recognition and classification." Lingvisticae Investigationes 30.1 (2007): 3-26.