**A REPORT**

**ON**

# APPLICATION OF MACHINE LEARNING TECHNIQUES FOR NUCLEAR DATA ANALYSIS


BY

AYUSH JAIN

2017A7PS00093P

B.E. (HONS.) COMPUTER SCIENCE AND ENGINEERING

AT

**INDIRA GANDHI CENTRE FOR ATOMIC RESEARCH, KALPAKKAM**


A Practice School-I station of


**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**


**(July, 2019)**

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**PILANI (RAJASTHAN)**

**Practice School Division**

**Station: Indira Gandhi Centre for Atomic Research** **Centre: Kalpakkam**

**Duration: 52 days** **Start Date:21st May 2019**

**Date of submission: 12th June 2019**

**Title of the Project: Application of machine learning techniques for nuclear data analysis.**

**ID No:2017A7PS0093P** **Name: Ayush Jain** **Discipline: Computer Science**

**Name of the Expert: Mrs. R. Vijayalaxmi** **Designation: SO/D**

**Section: WMSS/RCCG/EIG**

**Name of the PS Faculty: Dr. Satyapaul Singh**

**Key Words: Natural language processing, word embeddings, semantics**

**Project Area: Machine Learning**

# ACKNOWLEDGEMENT

I want to thank BITS Practice School Division for this excellent opportunity to pursue my Practice School program at the Indira Gandhi Centre for Atomic Research.

I would like to express my heartfelt gratitude towards Dr. A.K. Bhaduri, Director of IGCAR, for allowing me to work in the prestigious university of IGCAR and presenting such great projects for the PS-1 programme.

Secondly, I would like to thank Dr. Madurai Meenachi, coordinator of my PS programme at IGCAR to encourage me at various stages of my project and providing help whenever I was in need.

I want to thank my guide and the in-charge for the project, R. Vijyalaxmi for guiding me with all the project work, encouraging me and providing the necessary aid with all the technologies and explanations of the work I did for the project.

I thank Dr. Satyapaul A. Singh, my PS faculty for providing necessary help during my stay for PS at Kalpakkam and making the PS such a great learning experience.

I want to acknowledge the role of my colleagues and friends who helped me during the project and my parents for their immense support throughout the project.

# ABSTRACT

Nuclear domain, despite having a huge amount of data, suffers from lack of technical research in extracting information. The main reason behind this is that data is largely unstructured and requires vast domain knowledge. Natural Language Processing(NLP), employing various machine learning techniques, can address this problem effectively. This report discusses the main issues with the already existing word representation models and seek to bank on latest researches, to make a machine learning model which can effectively understand nuclear data and thus can perform meaningful tasks.

**TABLE OF CONTENTS**

# 1. INTRODUCTION

Natural Language Processing is one of the fastest growing fields in the machine learning community. It aims to harness the enormous power of huge corpora of data to map it to useful information. There are many useful tasks that can be performed by it like machine translation[1], question-answering[2], etc. For our project, we wanted to explore how we can make the machine understand the word meanings in general and nuclear-specific meanings in particular. But the major bottleneck is that machine cannot comprehend words and meanings as we humans can do so easily. It understands only 0 and 1 or abstractly the numbers. Hence it is important to map the words to a suitable vector space. In technical terms, we need to map words to their suitable embeddings. There were many approaches which have been used in the past to accomplish this task. The most ground-breaking of these have been Word2Vec[3], followed by Glove[4] and most recently by Elmo[5] and Bert[6]. After getting the word embeddings, we define an architecture depending upon the task we want to achieve. On training on a suitable dataset, the machine learns how to do that task itself. This is the basic methodology we use when we want to perform any task in the domain of machine learning. This report aims to describe the study done till now, define the basic terminologies require to understand the problem, describes the major research questions and seek to answer those.

# 2. WORD EMBEDDINGS

Word embeddings refer to the mapping for the words to some dense matrix, so that it can capture the semantic correlation between different words. It is the most important and starting point for any machine learning task on textual data. There are many approaches in the domain of natural language processing to produce the word embeddings. I describe some of the important and relevant ones.

## 2.1. WORD2VEC

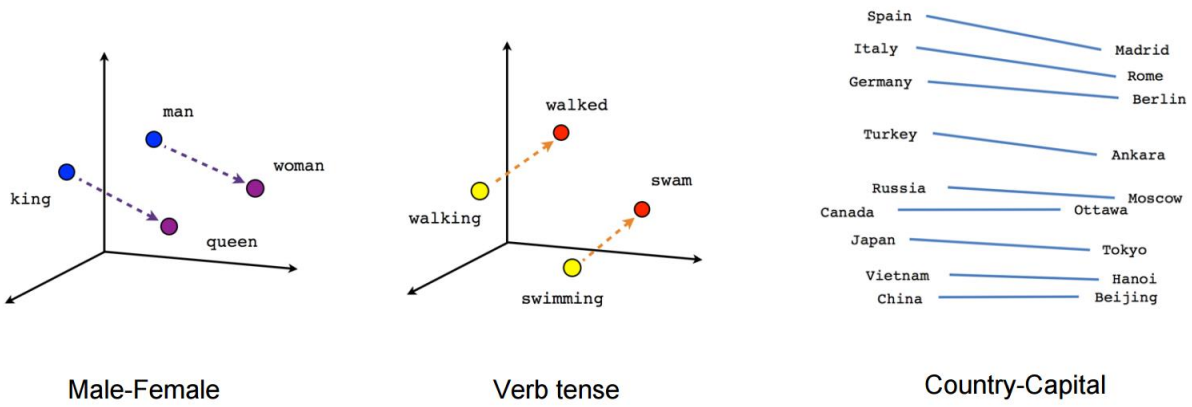


Figure-1: Word2Vec Embeddings

The word2vec[3] is unarguably is the most important breakthrough, which sparked a series of researches in the field of natural language processing. The step by step approach can be summarized as below:

1. **Pre-processing**: This is common to majorly anything we do in NLP. The text data is usually very noisy and hence needs to be pre-processed. The sentences are divided into individual word tokens, removing the punctuations, capitalization and other noises using various NLP techniques.

2. **Vocabulary** is formed by removing all the duplicates words.

3. A **one-hot-encoding** of all the words is formed based on the above-formed

vocabulary. These one-hot-encodings are grouped together to formed the input matrix to the architecture described in the word2vec paper[3].

For the brevity of this report, I won't go into the details of the architecture. More details can be referred to from the paper[3].

The output of the word2vec is a dense representation of the semantic meanings of the words. We can measure the similarity between different words using cosine similarity. Like ‘queen' and ‘woman' would be more similar than ‘queen' and ‘man'. Many seemingly magical tasks can be performed using them like ‘queen' – ‘woman' + ‘man' = ‘king'. These examples show that the word embeddings are not any random numbers but capture the relative meanings of the language.

## 3. WORD SENSE DISAMBIGUATION

Though they perform a very good task in capturing the meaning of the language, while analysing our task at hand, we realized that the major problem which our final product can face is "word sense disambiguation" due to polysemic nature of language.

**Polysemy** refers to the existence of multiple meanings of the same word. For example, a boiler can be ‘boiler in the nuclear industry', a boiler used in homes, etc. We, humans, are great in recognizing what a word means depending on the context it's used in. But the problem with the word2vec embeddings is that every word is represented by only one vector. And since the published pre-trained word2vec embeddings are trained on Wikipedia and general words dataset, it is not directly suitable for our case. Let's understand this with an example.

Let's consider the word ‘fuel'. Fuel can be associated with ‘automobile fuel', ‘nuclear fuel', ‘movie Fuel', ‘fuel computer language' etc. Now since it is trained on a general dataset, it is more similar to automobile fuel than nuclear fuel. But for our use case, we would like to have

a word embedding which is able to give more weight to our domain.

Hence, we thought of performing an experiment to adjust the word2vec vectors to shift them towards the nuclear words. There were many obstacles to this approach:

1. Do we have to manually select the words whose vectors we would like to shift? -The idea was that first for testing we would handpick some particular ambiguous words, again train the already trained embeddings on a custom-made dataset. The intuition was that the word vector would shift towards the nuclear domain. If it was successful, then we can retrain the word2vec embeddings on a bigger nuclear dataset, so that all the ambiguous words will shift towards nuclear semantic. It is then we encounter the next problem.

2. Word2vec does not handle out of vocabulary(OOV) words well. What it means is that word2vec is trained on a fixed vocabulary and all the words which are out of its vocabulary, it simply ignores it. This is very harmful to our case because nuclear data would have loads of words which would not have occurred in the dataset on which word2vec was trained.

3. Ignoring the OOV words also, it would require a significant amount of data to have appreciable deflection in the word2vec vectors because the hidden layers have a huge number of parameters. Such a huge amount of data would be very difficult to obtain.

## 4. CONTEXUALIZED WORD EMBEDDINGS: BERT

While meddling with these problems, we come across a very recent paper by Google ‘BERT'[6]. It has very elegantly handled the exact problem which we were facing. Instead of changing the word2vec model, it changed the entire architecture of how the machine learns the embeddings. The previous works in this domain like ELMO[5] and OpenAI GPT[7] tried to understand the context by reading from the left/right and predicting the next words using powerful architectures, the transformers, which outperforms a very popular architecture Long Short Term Memory(LSTM)[8] in retaining the context of the sentence. Though it

outperformed the then “state of the art accuracies”, they suffered from a major problem which BERT solved. The intuition of the problem lies in how humans comprehends the language. We consciously/unconsciously get the context of the word by understanding the words both on the left and right of that word. This is exactly what BERT tries to do. But a major problem with this approach is that if it sees both left and right, then it will be able to actually see the word in advance. Hence it might be able to cheat its way out! To mitigate this problem, they cleverly used masking, to mask some parts of the word. The detailed masking procedure is described in their excellent research paper[6]. Though the BERT model is also trained on general Wikipedia dataset, it gives the flexibility to transfer learn domain-specific embeddings. It banks on the already pre-trained word embeddings, which can be further trained on a domain-specific dataset. The idea of transfer learning[9], motivated by computer vision, enables to learn the embeddings with a smaller dataset.

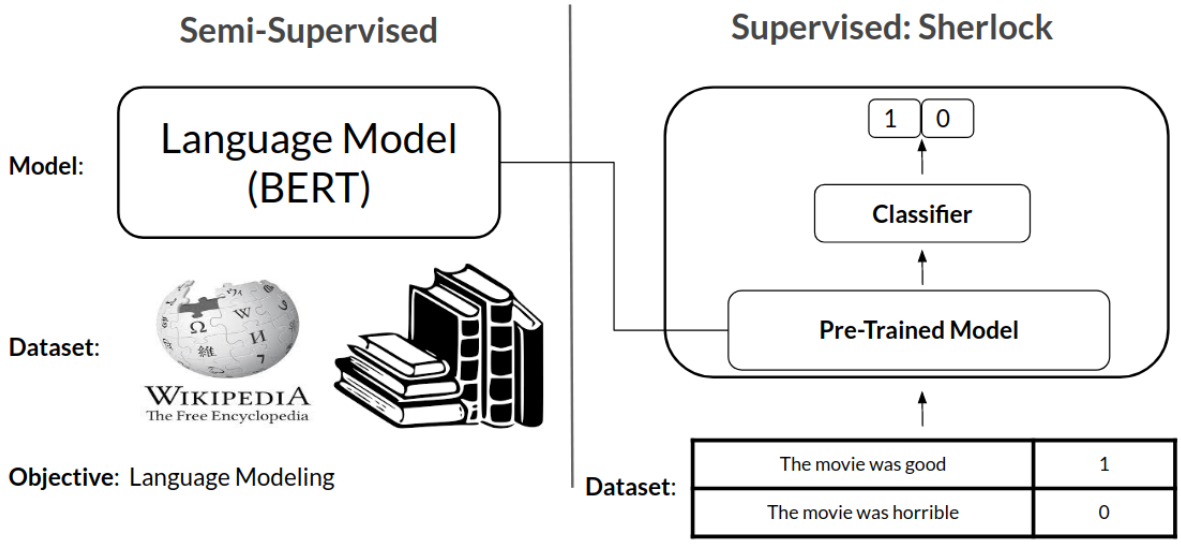


Figure-2: BERT: Pre-training and Fine-tuning

There are two important techniques introduced by BERT:

1. **Pre-training**: The researchers made a bidirectional encoder transformer architecture (which enables it to see both left and right context of a word) and trained it on a huge corpus of Wikipedia and other textual data. For our application we have two options: Either

we can keep the pretrained weights and continue training on our dataset to adjust it for our domain. The benefit of this approach is that the pre-trained weights already captures the knowledge of language representation and thus we would require a lesser amount of data to make the model learn the semantics of our domain. Also, this will be computationally less expensive. But the major drawback is that the vocabulary of the model can't be changed. Hence this approach is not suitable for those domains who have too many new words which are not included in the original vocabulary. Though BERT uses clever techniques like word-piece (dividing words into sub-words), but still it cannot match the scenario of having that word in its dictionary. The other option we have is to train the model from scratch. In this case the vocabulary would also be domain-specific and hence expected to perform better. But the bottle-neck is the data and computation resources. The main question here is: Can the model perform better while banking on its language understanding added with incomplete domain knowledge (case 1) or with a domain specific knowledge and vocabulary but not trained enough due to lack of data (case 2)? Also, if latter is better, then how much? Is it worth so much more computational investment (The latter will require nearly 20 times more compute)? We would have to experiment to reach a conclusion.

**2. Fine tuning**: This is demonstrated in the right part of the figure-2. After getting suitable word embeddings, we need to add one-two layers above the already trained architecture and train this new model on supervised(labelled) data. Since it is trained for a major portion, we can get good accuracies with less data. For example, for building a question-answering system, we would have to give the model a dataset with questions and answers. The model will see the questions and try to predict their answers hence learning how to answer questions.

Many research papers have come on BERT like SciBERT[10], BioBERT[11] which trained on the scientific research papers and biomedical datasets respectively and outperformed state of the art models on many tasks like question-answering, Name-entity-recognition, etc. It proves that using a context specific word embedding can go a long way in solving the polysemy problem and better understanding of the problem.

All these researches evaluated their models on the publicly published gold standard dataset. But the major problem in our domain is that there is no publicly available gold standard dataset. So, we decided to bank on other options like expert opinion for evaluating our results etc.

Nevertheless, we started working towards making the question answering platform. It is then, we encountered our next big problem. Agreed that we can have our customized word embeddings from BERT, but if we use these embeddings to make the question-answering platform but train the question-answering platform on the general datasets, then it won't make sense as it would try to answer according to the perspective of nuclear expert but the question wants answer in normal contexts. We need to train it on the nuclear dataset only. But there is no dataset available online. And making the dataset suitable for question answering would be a huge task because we would have to read a paragraph, generate a few questions, and then answer it.

## 5. AUTOMATIC QUESTION GENERATION

On a lot of searching the net, I came across a few research papers [12, 13, 14] which takes a paragraph as input and generates general questions. There are some rule-based methods too which take a paragraph and based on the grammar, generate questions. We would look into which method would work best for our case and experiment with that.

# CONCLUSIONS AND FUTURE WORK

Contextualized word embeddings are the building blocks of every specialized task that we aim to perform. It is crucial to find suitable methods, which can utilize unstructured data using unsupervised approaches, to derive semantically meaningful word embeddings.

We would be exploring BERT model, trying to devise experiments to understand the major pitfalls of using already available embeddings. We would experiment with pre-training and fine-tuning with different sizes of data to see how the amount of data is affecting the accuracy of the model. Finally we will be building a complex model using those embeddings like question-answering platform, and generate relevant insights from the data.

# REFERENCES

1. Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

2. Andrenucci, Andrea, and Eriks Sneiders. "Automated question answering: Review of the main approaches." Third International Conference on Information Technology and Applications (ICITA'05). Vol. 1. IEEE, 2005.

3. Goldberg, Yoav, and Omer Levy. "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method." arXiv preprint arXiv:1402.3722 (2014).

4. Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

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

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

7. Radford, Alec, et al. Improving language understanding with unsupervised learning. Technical report, OpenAI, 2018.

8. Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

9. Raina, Rajat, et al. "Self-taught learning: transfer learning from unlabeled data." Proceedings of the 24th international conference on Machine learning. ACM, 2007.

10. Beltagy, Iz, Arman Cohan, and Kyle Lo. "SciBERT: Pretrained Contextualized Embeddings for Scientific Text." arXiv preprint arXiv:1903.10676 (2019).

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

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

13. 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.

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