How to Cite:
Awachat, S., Raipure, S., & Kalambe, K. (2022). A technical review on knowledge intensive NLP for pre-trained language development. *International Journal of Health Sciences*, *6*(S2), 9591–9602. https://doi.org/10.53730/ijhs.v6nS2.7510

A technical review on knowledge intensive NLP for pre-trained language development

Snehal Awachat
Department of CSE RCOEM, Nagpur, India
Corresponding author email: awachatsr@rknec.edu

Shwetal Raipure
Department of CSE RCOEM, Nagpur, India
Email: raipuress@rknec.edu

Kavita Kalambe
Department of CSE RCOEM, Nagpur, India
Email: kalambekb@rknec.edu

***Abstract*---**In today's world where data plays the very important role, we have various sources of pre-data like online books, equation analysis, encyclopedia, common-sense reasoning, common-sense knowledge, etc. The increasing capacity of pre-training language models have given knowledge intensive natural language processing (KI-NLP) a new boost for advanced functionalities for establishing a stable, flexible, robust and efficient model. Though pre-trained models have its own drawback for handling the KI-NLP tasks, we are here to discuss the challenges faced in this field. A wide variety of pre-trained language models enhanced with external knowledge sources have been proposed and are in rapid development to meet this difficulty. In this research we have also discusses the challenges in NLP in terms of generation of knowledge intensive models. We have also defined some mathematical model and its framework dependability for pre-training different language in NLP. Finally, we have also discussed about variety of literature reviews based on we intend to describe the present progress of pre-trained language model-based knowledge-enhanced models (PLMKEs) in this work by deconstructing their three key elements: information sources, knowledge-intensive NLP tasks, and knowledge fusion methods.

***Keywords*---**pre-trained NLP, NLP, knowledge-based-NLP.

Manuscript submitted: 27 March 2022, Manuscript revised: 18 April 2022, Accepted for publication: 9 May 2022

Introduction

The BERT [1] and GPT [2] families of large pre-trained transformer-based language models (PLMs) have swept the area of Natural Language Processing (NLP), achieving state-of-the-art performance on a number of applications. As a result of these large PLMs, NLP has undergone a paradigm shift. Consider the classification problem p(y|x) (converting textual input x to a label y): traditional statistical NLP techniques employ hand-crafted features to describe x, and then a machine learning model is used to learn the classification function (e.g., SVM, logistic regression). Deep learning models use a deep neural network to learn the latent feature representation in addition to the classification function [3].

The task of resolving numerous named entities mentions in an input-text sequence to their correct references in a knowledge graph is known as named entity disambiguation (NED). Author addresses the NED problem by combining two novel pre-training objectives and proposing a new pre-training NED model. Experiments are carried out using the CoNLL and TAC datasets, as well as other datasets offered via the GERBIL platform. The experimental results show that the suggested model outperforms earlier models by a significant margin [4]. Denoising autoencoding-based pretraining, such as BERT, outperforms pretraining systems based on autoregressive language modelling because it can simulate bidirectional contexts [5]. On numerous subtasks of natural language processing, neural networks for language modelling have been shown to be effective. Deep language model training, on the other hand, is time-consuming and computationally intensive. Using multi-class text classification results reveal that the FinBERT model does not improve performance when compared to generic BERT models [6].
Textual factual knowledge is difficult to capture with pre-trained language representation models (PLMs). Knowledge embedding (KE) methods, on the other hand, can successfully describe relational facts in knowledge graphs (KGs) with useful entity embeddings, but traditional KE models are unable to fully exploit the plentiful textual information. Author have proposed a unified approach for Knowledge Embedding and Pre-trained Language Representation (KEPLER), which may provide effective text-enhanced KE with strong PLMs while also better integrating factual knowledge into PLMs [7].

In several NLP disciplines, fine-tuning pre-trained language models has had a lot of success. However, it is remarkably vulnerable to adversarial cases, such as word substitution assaults that use only synonyms to trick a BERT-based sentiment analysis machine [8]. Text mining has been utilised to extract knowledge from free texts throughout the previous few decades. Over the years, applying neural networks and deep learning to natural language processing (NLP) tasks has resulted in numerous achievements for real-world language difficulties [9]. These models have been shown to capture numerous aspects of language such as hierarchical relations, long-term dependency, and sentiment, and are considered the NLP counterparts of Image Net [10].

Model

The BERT-Cap Model

For user intent classification, a BERT-Cap hybrid model with focal loss based on pretrained BERT and capsule network has been proposed[1,6]. Input embedding, sequence encoding, feature extraction, and intent categorization are the four modules that make up the BERT-Cap model. Figure 1 depicts the architecture of our model. The input embedding module represents a sentence to a sequence of embeddings by maintaining token information, position information, and segment information when given a sentence as input. The sequence encoding module loads the pre-trained language model produced through transfer learning and performs sentence encoding using the transformer encoder.

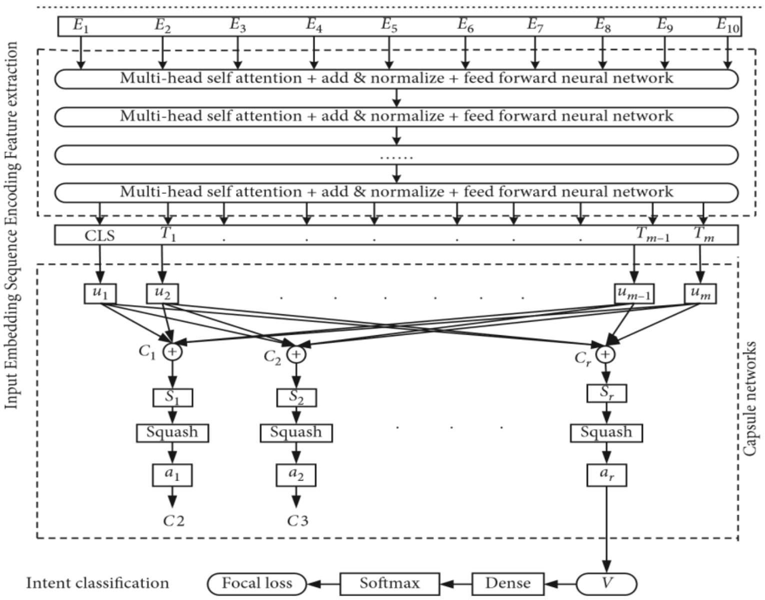

Fig 1: BERT architecture model

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. If you are using A4-sized paper, please close this file and download the file “MSW_A4_format”.

Token-based Pre-trained Models

With the emergence of all the above token-based pre-trained models, word embeddings have been commonly used as text representation in NLP tasks. Although these models are simple and effective, they are only suited to attain fixed representations rather than capturing polysemy. That is also why we call this type of model static pre-trained models.

Context-based Pre-trained Models

To address the problem of polysemy, pre-trained models need to distinguish the semantics of words and dynamically generate word embeddings in different contexts. Given a text $x_1, x_2, \ldots, x_T$where each token x_t is a word or sub-word, the contextual representation of x_t depends on the whole text.

$$\left[h_1, h_2, \ldots, h_T\right] = f_{enc}(x_1, x_2, \ldots, x_T) \ , \qquad (1)$$

where $f_{enc}(\cdot)$ is neural encoder and h_t is contextual embedding.

Constructs of Language Models

It is essential to understand the construct of the language model that effectively applies to solve the problems associated with software security issues. Problem related to software security can be solved by leveraging the natural language data that is available across companies and industries. Effective language modelling capabilities can help derive the information hidden in these data. Further from this, security-related information can be leveraged by the software development team as and when they need it. Language models are the basis of the models that used in this research by author. Language models find their roots in the N-gram modelling approaches. N-gram modelling uses the though process of assessing the probability of a given the word using its history [31]. For example, the probability of the next word in the phrase "Jack and Jill went up the" to be "hill."

P (hill|Jack and Jill went up the) (1)

One of the approaches used to compute this probability is relative frequency. By taking the corpus of language as a base, how often the word "hill" follows the phrase can be calculated as follows:

P(hill|Jack and Jill went up the) = $\frac{C(JackandJillwentupthehill)}{C(JackandJillwentupthe)}$ (2)

where C represents the count of occurrence of the phrase.
The chain rule of probability is applied to words to obtain the following expression:

$$P\left(W_{1:n}\right) = P\left(w_1\right) P\left(w_2 \middle| w_1\right) P\left(w_3 \middle| w_{1:2}\right) \ldots P(w_n | w_{1:n-1})$$
$$= \prod_{k=1}^{n} P\left(w_k \middle| w_{1:k-1}\right), \qquad (3)$$

where w denotes the word, n represents a word count, and k represents the length of the sequence. To simplify the complexity of dependencies on the word, Markov it will be enough to look at n − 1 previous words. Maximum likelihood estimation is used for calculating the probabilities of N-grams as follows:

$$P\left(w_n \backslash wn - 1\right) = \frac{c(w_{n-1} w_n)}{w_{n-1}}$$

KEPLER

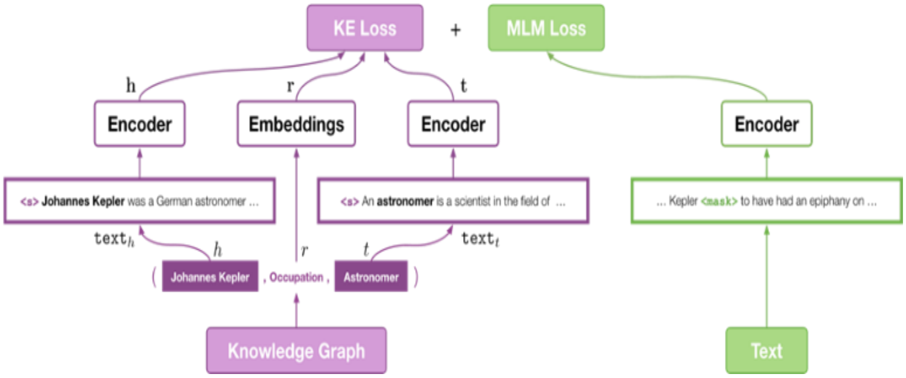

Fig 2: The KEPLER framework

We encode entity descriptions as entity embeddings and jointly train the knowledge embedding (KE) and masked language modelling (MLM) objectives on the same PLM [32].

Knowledge Embedding in KEPLER

We use the knowledge embedding (KE) aim in our pre-training to integrate factual knowledge into KEPLER. KE uses distribute representations to encapsulate entities and relations in knowledge graphs (KGs), which helps with a variety of downstream tasks like link prediction and relation extraction.

Entity Descriptions as Embeddings For a relational triplet (h, r, t), we have:

$h = E_{<s>}\left(text_h\right),$

$$t = E_{<s>}\left(text_t\right),$$
$$r = T_r,$$

Where $text_h$ and $text_t$ are the descriptions for h and t, with a special token <s> at the beginning. $T \epsilon \mathbb{R}^{|R| \times d}$ is the relation embeddings and h, t, r are the embeddings for h, t, and r.

Methodology

Knowledge Embedding in KEPLER

NLP systems that use 10,000 words and have a strong coverage of the domain in question are uncommon. A few writers have achieved this objective using prototypes that take advantage of existing linguistic knowledge corpora. If we don't want to recreate the wheel, this is the best way to go. Existing sources must thus be reused. Consider the many sources that are available one by one. They will be evaluated based on the type of linguistic expertise they can offer, in order to emphasise their uniqueness.

For appropriate handling of free text utterances, many kinds of linguistic knowledge are required.' The emphasis shifts from one category to another depending on the treatment and methods used to emphasise it. The most complex system, on the other hand, is based on all categories. The following classifications have been established:
vocabulary and its syntax,
concepts and their typology,
semantic co-occurrences,
conceptual schemata and frames.

This broad classification is useful in this context, but it should be refined further when examining real implementations. NLP technologies also require a slew of other rules and linguistic data.

Natural Language Processing is built on the foundation of vocabulary and syntactic information. Without huge lexicons, nothing can be accomplished. However, as soon as we want a multilingual vocabulary, a problem arises: how can we connect together the same terms in various languages? The most simple solution is to connect them using an abstract object that conveys the meaning of these words, which we refer to as an idea. After a couple of thousand thoughts have been collected, the following stage is to organise them. This is commonly referred to as a concept typology or ontology. Single inheritance hierarchy approaches are used in implementations aimed at naming and categorization. Other solutions are based on a domain compositional model and are better able to offer the semantic information that NLP tools require. There are just a few ongoing domain representation projects.

The UMLS Source

The UMLS knowledge sources make a vast quantity of linguistic data easily accessible to the medical community. However, the form and organisation of this knowledge is somewhat of a compromise between several methods.

The GALEN Source

The GALEN group has been working together since 1992 and has created a generic model of medicine with almost 6,000 ideas using the GRAIL representation language. Because this technique is compositional, it provides a high level of complexity for concepts and a fine granularity of knowledge representation. This paradigm is compatible with conceptual graphs, and the perspective it provides on a given notion makes it comparable to a frame system. The GALEN technique has the disadvantage of being difficult to scale and achieve ready-to-use coverage of the medical sector.

The MED Source

The Medical Entity Dictionary (MED) is a frame-like medical domain model with the goal of providing a restricted vocabulary for medical applications. It's built on top of the UMLS semantic network, and it's being gradually expanded to satisfy the demands of supplementary healthcare systems.

The SNOMED Source

SNOMED International publishes a bilingual version with thorough coverage of medical terminology and topics. It is in a machine-readable format. Because it is a systematic nomenclature, it has a wide range of applications. SNOMED is a structured repository of information with the potential to benefit NLP systems due to its multiaxial approach

Knowledge Types	Knowledge Sources	Knowledge Domains
Encyclopaedic Knowledge	Wikipedia /Wikidata [31]	open domain
	DBpedia	
	Freebase	
	UMLS Bodenreider	biomedicine
	AMiner	science
Commonsense Knowledge	Concept Net	Open domain
	TransOMCS	
	CSKG	
	ATOMIC	Human interaction
	ATOMIC	
	ASER	eventuality

Table 1: Common knowledge sources used in PLMKEs

Knowledge-Intensive NLP

Knowledge-intensive jobs (e.g., open Question Answering (QA), fact checking) are intended to obtain evidence passages relating to an input query from a huge collection of passages, such as Wikipedia. The most effective approaches for these tasks include retrieval-augmented generation [33], which is taught to create the final output given recovered passages from a separate-trained retriever. As demonstrated by recent work (Xu), this training process disregards the evidentiality of passages whether the passages accurately support the output or not, resulting in the use of misleading cues or the generation of hallucinations. For the QA test, for example, a response may be wrongly extracted from a paragraph having a high lexical overlap with the question. This might occur as a result of memorising of out-of-date knowledge, particularly as many portions in training aren't evidence-based. For some QA jobs, using heuristics to train the generator with passages containing the target strings might help address the problem. Even yet, there's a chance that these texts don't provide enough evidence to resolve questions. Furthermore, such heuristics are ineffective for applications requiring open-ended creation or categorization.

Base Generator $\mathcal{G}$

As our basic generator model G, we adopt FiD , a state-of-the-art retrieval-augmented generation model. For clarity, we provide a high-level overview of the model, directing the reader to Izacard and Grave (2021b) for further information.

- Encoder:

- Using a pre-trained T5 encoder, we first encode an input query and passages. Each passage has the input question x prepended to it, and the encoder encodes all N passages individually. We convert passage pi into pi R(Lh), where L is the length of the input text and h is the concealed size.
- Answer generator:
  $\widehat{P}$ is a concatenation of the encoded sections that forms a summary representation of the input. P is input into the response generator, which generates the final answer autoregressively. It produces the sequence probability for y in the following format:

$$P\,(y/x\,,\widehat{P}) = \prod_{j=1}^{T} p(y_i|y_{<j},x,\widehat{P}).$$

where y_i denotes the j th token of the generated output y and T is the length of the final output. The generator is based on the T5 architecture and uses cross attentions to model the interactions between retrieved passages.

Mining Silver Evidentiality E^{silver}

The majority of datasets and tasks only offer query-answer (x, y) annotations for passages and do not include evidentiality labels E. Some gold evidence annotation datasets, such as Natural Questions or HotpotQA [34], cover subsets of gold passages from certain Wikipedia articles, whereas P may include unlabeled gold passages from a different page. False-positive annotations can be created by labelling paragraphs that include the response string as evidentiality positive passages. For example, p2 in contains the answer string "seven," but it has no bearing on the input inquiry. Importantly, even this noisy heuristic is not accessible for tasks like knowledge-enhanced discourse and fact verification that need open-ended generation or answer categorization.

- Leave-one-out evidentiality mining:

Gold evidence annotations are missing from the majority of knowledge-intensive datasets. We provide a new method for mining evidentiality data that involves determining whether passages give enough information for a trained model to create.

Task	Datasets	Data Sources
	NATURAL QUESTIONS	Wikipedia
	HOTPOTQA [35]	Wikipedia
Fact Verification	FEVER	Wikipedia
	BaolQ	Wikipedia
Entity Linking	ACE2004 [36]	news
	AIDA CoNLL-YAGO	DBPedia & YAGO
	WNWI	Wikipedia
	WNCW	Clueweb

Table 2: Detailed information about representative encyclopedic knowledge-intensive tasks and datasets.

Knowledge-Intensive NLP

Entities (points), relations, and qualities are the main components of knowledge (edges). We first implement matching and linking of entities based on relations and characteristics in the process of investigating open network knowledge fusion, and then associate and merge the corresponding relations and attributes with the original information in the knowledge base based on the link findings. The goal of entity matching and linking, also known as the entity linking technique, is to dynamically link entities found on the Internet to the knowledge base after they have been matched.

i. Named Entity Recognition
   Named entity recognition (NER) is used to extract the provided entity item from the text, and the entity in the text is acquired together with its category. Rule-based techniques, statistical-based methods, and neural network-based methods are the three types of named entity recognition methods.The rule-based approach [48], which included pattern matching, finite automata, and other techniques, was the dominant method employed in the early days. In both Chinese and English, such strategies have had positive outcomes. For example, Chinese organisation names, such as Dalian Wanda Group and China Southern Power Grid Company, have certain structural features, such as place names + other components + characteristic words (such as company, group, etc.); named entities in English often contain lexical features, such as vocabulary abbreviations, initial capitalization, and so on. These techniques, on the other hand, are overly difficult in terms of establishing rules, overly reliant on empirical information, and lack portability.
ii. Candidate Entity Generation:
   To address the many-to-many, link between the referential item and the entity [49], the candidate entity is created. While maintaining the recall rate, the comparison range should be narrowed to increase the efficiency and quality of subsequent computations. The candidate entity collection should have the fewest unrelated and most Equivalent items.
iii. Disambiguation Sorting:
   For the following phase of fusion, disambiguation sorting is used to find the related entity of the entity referential item. In most cases, the context background information and the graph's structural knowledge are employed in this stage. Probabilistic based techniques, graph based methods, deep learning based methods, and representation learning based methods are the four types of disambiguation sorting systems.

Task	Dataset	Models	Fusion Types	Fused Knowledge	
	NATURALS QUESTIONS [34]		UnitedQA	Post-fusion	Wikipedia
	WEBQUESTIONS		EMDR	Post-fusion	Wikipedia
Open-domain QA	TEIVIAQA		EMDR	Post-fusion	Wikipedia

		REALM	Hybrid-fusion	Wikipedia
Other Representative model		Hybrid-fusion	Wikipedia	

Conclusions

The enhanced functionality for developing a stable, flexible, resilient, and efficient model in knowledge intensive natural language processing has been fueled by the growing capability of pre-training language models (KI-NLP). While pre-trained models have their own set of drawbacks when it comes to addressing KI-NLP duties, we're here to look into the challenges that this field has to offer. A variety of pre-trained language models coupled with external knowledge sources have been established and are in rapid development to solve this difficulty. This paper also looks at the issues that come with developing knowledge-intensive models using NLP. In addition, we developed mathematical models for pre-training numerous languages in natural language processing, as well as their framework dependability. Finally, we discussed a variety of literature reviews in order to describe the current state of pre-trained language model-based knowledge-enhanced models (PLMKEs) in this work by breaking down their three critical components: information sources, knowledge-intensive natural language processing tasks, and knowledge fusion methods. Future NLP models will benefit from the pre-trained knowledge-based model's increased efficiency, accuracy, and precision, as well as its wide range of applications.

References

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
2. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. OpenAI blog, page 12.
3. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. nature, 521(7553):436–444.
4. ZIZHENG JI 1, LIN DAI 1, JIN PANG 2, AND TINGTING SHEN," Leveraging Concept-Enhanced Pre-Training Model and Masked-Entity Language Model for Named Entity Disambiguation", Received April 7, 2020, accepted April 24, 2020, date of publication May 12, 2020, date of current version June 9, 2020.Digital Object Identifier 10.1109/ACCESS.2020.2994247.
5. Zhilin Yang 1, Zihang Dai 12, Yiming Yang1, Jaime Carbonell1, Ruslan Salakhutdinov1, Quoc V. Le," XLNet: Generalized Autoregressive Pretraining for Language Understanding", 19 Jun 2019
6. Yusuf Arslan, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F. Bissyandé, Jacques Klein, and Anne Goujon. 2021. A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain. In Companion Proceedings of the Web Conference 2021 (WWW '21

Companion), April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3442442.3451375.

7. Wang, Xiaozhi & Gao, Tianyu & Zhu, Zhaocheng & Liu, Zhiyuan & Li, Juanzi & Tang, Jian. (2019). KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation.
8. Hai Liu,1,2 Yuanxia Liu,1 Leung-Pun Wong,3 Lap-Kei Lee ,3 and Tianyong Hao,” A Hybrid Neural Network BERT-Cap Based on Pre-Trained Language Model and Capsule Network for User Intent Classification”, Volume 2020, Article ID 8858852, 11 pages https://doi.org/10.1155/2020/8858852
9. Schomacker, T.; Tropmann-Frick, M. Language Representation Models: An Overview. Entropy 2021, 23, 1422. https:// doi.org/10.3390/e23111422
10. Munazza Zaib, Quan Z. Sheng, and Wei Emma Zhang. 2020. A Short Survey of Pre-trained Language Models for Conversational AI-A New Age in NLP. In Proceedings of the Australasian Computer Science Week Multiconference (ACSW 2020), February 4–6, 2020, Melbourne, VIC, Australia. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3373017.3373028.
11. Bonan Min , Hayley Ross , Elior Sulem , Amir Pouran Ben Veyseh ,Thien Huu Nguyen, Oscar Sainz , Eneko Agirre , Ilana Heinz , and Dan Roth, “Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey”.
12. Da Yin1∗ , Li Dong2 , Hao Cheng2 , Xiaodong Liu2 , Kai-Wei Chang1 , Furu Wei2 , Jianfeng Gao2, “A Survey of Knowledge-Intensive NLP with Pre-Trained Language Models”.
13. Hai Liu,1,2 Yuanxia Liu,1 Leung-Pun Wong,3 Lap-Kei Lee ,3 and Tianyong Hao, “A Hybrid Neural Network BERT-Cap Based on Pre-Trained Language Model and Capsule Network for User Intent Classification”, Volume 2020, Article ID 8858852, 11 pages https://doi.org/10.1155/2020/8858852
14. Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, Jinghui Peng, “A Survey of Knowledge Enhanced Pre-trained Models”.
15. Junyi Li1,3† , Tianyi Tang2† , Wayne Xin Zhao1,3∗ and Ji-Rong Wen1,2,3 1 Gaoling, “Pretrained Language Models for Text Generation: A Survey”,2021.
16. Raghavendra Rao Althar , Debabrata Samanta , Manjit Kaur Abeer Ali Alnuaim, Nouf Aljaffan, and Mohammad Aman Ullah , “Software Systems Security Vulnerabilities Management by Exploring the Capabilities of Language Models Using NLP”, Computational Intelligence and Neuroscience Volume 2021, Article ID 8522839, 19 pages https://doi.org/10.1155/2021/8522839
17. Bonan Min , Hayley Ross , Elior Sulem , Amir Pouran Ben Veyseh ,Thien Huu Nguyen4, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth, “Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey”.
18. Xiaozhi Wang , Tianyu Gao , Zhaocheng Zhu , Zhengyan Zhang , Zhiyuan Liu , Juanzi Li , and Jian Tang , “KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation”.
19. Xipeng Qiu*, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai & Xuanjing Huang, “Pre-trained Models for Natural Language Processing: A Survey”.
20. Adithya V Ganesan, Matthew Matero, Aravind Reddy Ravula, Huy Vu, H. Andrew Schwartz,” Empirical Evaluation of Pre-trained Transformers for Human-Level NLP: The Role of Sample Size and Dimensionality”, Proc Conf. 2021 June ; 2021: 4515–4532. doi:10.18653/v1/2021.naacl-main.357.

21. Anthony Gillioz,” Overview of the Transformer-based Models for NLP Tasks”, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 179–183 DOI: 10.15439/2020F20 ISSN 2300-5963 ACSIS, Vol. 21
22. Cong Sun1, Zhihao Yang1*, Lei Wang2*, Yin Zhang2, Hongfei Lin1 and Jian Wang1,” Deep learning with language models improves named entity recognition for PharmaCoNER”, Sun et al. BMC Bioinformatics (2021) 22:602 https://doi.org/10.1186/s12859-021-04260-y
23. Donghan Yu1 ——————————————, Chenguang Zhu2————————————, Yiming Yang1, Michael Zeng2,” JAKET: JOINT PRE-TRAINING OF KNOWLEDGE GRAPH AND LANGUAGE UNDERSTANDING”, 2 Oct 2020
24. Fang Liu, Ge Li∗, Yunfei Zhao, Zhi Jin∗,” Multi-task Learning based Pre-trained Language Model for Code Completion”, 29 Dec 2020
25. Hai Liu,1,2 Yuanxia Liu,1 Leung-Pun Wong,3 Lap-Kei Lee ,3 and Tianyong Hao,” A Hybrid Neural Network BERT-Cap Based on Pre-Trained Language Model and Capsule Network for User Intent Classification”, Volume 2020, Article ID 8858852, 11 pages https://doi.org/10.1155/2020/8858852
26. Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, XintaoWu, Ben Graff, Dongwon Lee,” MathBERT:A Pre-trained Language Model for General NLP Tasks in Mathematics Education”, 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Workshop on Math AI for Education (MATHAI4ED)
27. Junyi Li1;3y , Tianyi Tang2y , Wayne Xin Zhao1;3 —————————————— and Ji-Rong Wen,” Pretrained Language Models for Text Generation: A Survey”, Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21) Survey Track
28. Linqing Liu,” Towards Elective Utilization ofPretrained Language Models”, 2020
29. Ming Zhou, Nan Duan, Shujie Liu, Heung-Yeung Shum,” Progress in Neural NLP: Modeling, Learning, and Reasoning”, Received 30 April 2019 Revised 30 August 2019 Accepted 13 October 2019 Available online 7 January 2020
30. Nankai Lin, Yingwen Fu, Chuwei Chen, Ziyu Yang, Shengyi Jiang,” LaoPLM: Pre-trained Language Models for Lao”,2020.
31. Wikipedia, n-Gram, https://en.wikipedia.org/wiki/N-gram, 2021.
32. Xiaozhi Wang , Tianyu Gao , Zhaocheng Zhu , Zhengyan Zhang , Zhiyuan Liu , Juanzi Li , and Jian Tang , “KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation”.
33. Patrick S. H. Lewis, Ethan Perez, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In NeurIPS, 2020.
34. Tom Kwiatkowski, Jennimaria Palomaki, et al. Natural Questions: A Benchmark for Question Answering Research. TACL, 2019.
35. Zhilin Yang, Peng Qi, et al. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In EMNLP, 2018.
36. US NIST. The Ace Evaluation Plan. US National Institute for Standards and Technology (NIST), 2004.