ReBERT- An Enhanced BERT

1Priyanka Shee, 2Santayo Kundu, 3Anirban Bhar, 4Moumita Ghosh

1,2 B. Tech student, 3,4 Assistant Professor
Department of Information Technology
Narula Institute of Technology
Kolkata, India.

Abstract- BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based language model that comprehends the context of words by considering surrounding words in both directions. It revolutionized natural language processing by capturing rich contextual information, enhancing performance in various language understanding tasks like sentiment analysis, text classification. In this article, focusing on User Generated Content (UGC) in a resource-scarce scenario, we study the ability of BERT (Devlin et al., 2018) to perform lexical normalization, by enhancing its architecture and by carefully finetuning it, we show that BERT can be a competitive lexical normalization model without the need of any UGC resources aside from 3,000 training sentences. The enhanced BERT model features a hierarchical contextualization module for improved long-range dependency understanding, a domain-specific adaptation layer for specialized language contexts, and efficiency optimization through dynamic attention head pruning and weight sharing. Fine-tuned pre-training broadens language comprehension, while task-specific heads enable fine-tuning. Rigorous evaluation and iterative refinement ensure performance enhancement across tasks, addressing limitations and advancing language understanding. It will be our first work done in adapting and analyzing the ability of this model to handle noisy UGC data.

Keywords- BERT, NLP, User-Generated Content (UGC), Syntactic Parsing, Name-Entity Recognition.

I. INTRODUCTION

Pre-trained contextual language models (e.g., Peters et al., 2018; BERT, Devlin et al., 2018) have improved the performance of a large number of state-of-the-art models on many Natural Language Processing (NLP) tasks. Unfortunately, we can't always gauge the full impact of modelling advancements because training is computationally expensive, reducing the amount of fine-tuning that can be performed. A major specificity of BERT is that it is trained to jointly predict randomly masked tokens as well as the consecutiveness of two sentences. Moreover, it takes as input Word Pieces tokens which consists in frequent sub-word units (Schuster and Nakajima, 2012). Finally, available pre-trained models have been trained on the concatenation of the Wikipedia corpus and the Book Corpus, which constitutes a large corpus of canonical (i.e., proper, edited) language however, as the NLP field continues to evolve, there is a constant quest for improving upon existing models to enhance their capabilities. ReBERT, short for "Reimagined BERT," represents an enhanced version of the original BERT model, designed to address some of its limitations and further push the boundaries of NLP performance.

Finally, available pre-trained models have been trained on the concatenation of the Wikipedia corpus and the Book Corpus, which constitutes a large corpus of canonical (i.e., proper, edited) language. Thus aside the efficiency of its transformer-based architecture, these three aspects respectively enable BERT to elegantly cope with out of vocabulary words and to include contextual information at the token and at the sentence levels, while fully taking advantage of a training corpus containing billions of words. Without exhausting the topic, BERT has successfully surpassed the state-of-the-art in a number of tasks, including Name-Entity Recognition (Devlin et al., 2018), Question Answering (Lample and Conneau, 2019), and Machine Translation (Lample and Conneau, 2019). Moreover, it has recently been shown to capture a rich set of syntactic information (Hewitt and Manning, 2019; Jawahar et al., 2019), without the added complexity of more complex syntax-based language models. However, it remains unclear and, to the best of our knowledge, unexplored, how well can BERT be used in handling non-canonical text such as User-Generated Content (UGC), especially in a low resource scenario. This question is the focus of this paper.

1. Enhanced Pretraining Strategies: ReBERT incorporates advanced pretraining techniques that go beyond simple masked language modelling. These strategies include novel tasks and objectives that encourage the model to acquire a deeper understanding of language semantics, structure, and context.

2. Architectural Refinements: ReBERT features architectural enhancements, optimizing the Transformer architecture for NLP tasks. It includes modifications in the attention mechanism, layer normalization, and positional embeddings, all aimed at improving model efficiency and interpretability.
3. **Larger-Scale Training**: To capture a more extensive linguistic context, ReBERT is pretrained on a larger corpus, incorporating a broader range of textual data from diverse domains and languages. This results in a model with a richer understanding of global language patterns.

4. **Domain-Specific Fine-Tuning**: ReBERT is designed to be easily fine-tuned for specific domains or tasks, making it adaptable to a wide array of applications. This flexibility allows users to achieve superior performance on domain-specific NLP challenges.

5. **Efficiency and Scalability**: In response to concerns about the computational resources required by large models like BERT, ReBERT introduces optimizations for efficiency and scalability, enabling it to be deployed in resource-constrained environments.

6. **Interpretability and Explainability**: ReBERT includes features that enhance model interpretability and explainability. This is crucial for understanding the model’s decision-making processes, making it more trustworthy for real-world applications.

7. **Multimodal Integration**: Recognizing the growing importance of multimodal NLP, ReBERT offers improved support for combining text with other data types, such as images, audio, or video, for a more holistic understanding of content.

II. **LITERATURE STUDY**

The history of BERT (Bidirectional Encoder Representations from Transformers) is a fascinating journey that has significantly shaped the field of natural language processing (NLP). BERT, introduced by Google AI researchers in 2018, has had a profound impact on various NLP applications and has become a foundational model for many subsequent developments. Here's a brief history of BERT:
1. **Pre-BERT Language Models:** Before BERT, NLP models primarily relied on recurrent neural networks (RNNs) and convolutional neural networks (CNNs). While these models showed promise, they had limitations in capturing long-range dependencies and contextual information in text.

2. **Transformer Architecture:** The Transformer architecture, introduced in the paper "Attention Is All You Need" by Vaswani et al. in 2017, marked a significant breakthrough in NLP. Transformers introduced a self-attention mechanism that allowed models to capture contextual relationships efficiently, enabling parallel processing of sequences.

3. **BERT's Emergence:** BERT was introduced by Jacob Devlin and his team from Google AI in the paper "BERT: Bidirectional Encoder Representations from Transformers" in October 2018. BERT departed from previous models by introducing the concept of bidirectional pretraining. Instead of training models on text in a unidirectional manner (either left-to-right or right-to-left), BERT pretrained models using both left and right contexts, significantly improving their understanding of language.

4. **Masked Language Model (MLM):** One of BERT's key innovations was the MLM objective, where random words in a sentence were masked, and the model was tasked with predicting those masked words based on the context provided by the unmasked words. This objective allowed BERT to learn contextual embeddings that captured the nuances of language.

5. **Pretraining on Large Corpora:** BERT was pretrained on a massive corpus of text, including parts of the internet and Wikipedia. This extensive pretraining data gave the model a broad understanding of language, making it suitable for various downstream NLP tasks.

6. **Transfer Learning Revolution:** BERT demonstrated the power of transfer learning in NLP. Instead of training models from scratch for specific tasks, fine-tuning BERT on task-specific data became a common practice, leading to state-of-the-art results in numerous NLP benchmarks.

7. **BERT Variants and Improvements:** Following BERT's success, researchers developed several variants and improvements, including RoBERTa, ALBERT, T5, and more. These models optimized BERT's architecture, training techniques, and model sizes, contributing to the ongoing refinement of pretrained language models.

8. **Multilingual and Cross-Modal Models:** The success of BERT inspired the creation of multilingual models like mBERT and cross-modal models that could handle multiple data types, such as text, images, and audio.

9. **BERT's Impact on NLP Applications:** BERT and its variants revolutionized various NLP applications, including sentiment analysis, named entity recognition, machine translation, question answering, and more. These models have been widely adopted in both academia and industry.

10. **Ongoing Research:** The BERT-inspired models continue to be at the forefront of NLP research, with advancements in interpretability, efficiency, and adaptability. Researchers are also exploring ways to address bias and ethical concerns associated with these models.

The field of Natural Language Processing (NLP) has seen remarkable advancements in recent years, and at the heart of these breakthroughs is the development and enhancement of language models. Among these models, BERT (Bidirectional Encoder Representations from Transformers) stands as a pivotal milestone. However, the quest for refining and advancing BERT and similar models persists. In this literature study, we delve into the evolution of BERT into "Refined BERT" or ReBERT, exploring the key research contributions, enhancements, and the broader context in which ReBERT has emerged.

1. **BERT: A Revolution in Language Modelling:**
   - Devlin et al., 2018
   - The journey towards ReBERT begins with BERT, which introduced a paradigm shift in NLP.
   - BERT's innovation was its ability to learn bidirectional representations from massive text corpora, laying the foundation for subsequent developments.

2. **RoBERTa: Optimizing BERT Pretraining for Robust Performance**
   - Liu et al., 2019
   - RoBERTa expanded upon BERT's architecture and training methodology, showcasing the potential for optimization and improved performance. This work demonstrated that refining BERT-like models is not only possible but also highly beneficial.

3. **ALBERT: Lite BERT for Efficiency**
   - Lan et al., 2019
   - ALBERT addressed the computational demands of large-scale language models, focusing on model size reduction while maintaining or even surpassing BERT's performance. It paved the way for more efficient versions of BERT, a theme embraced by ReBERT.
4. **XLNet: Autoregressive Pretraining for Enhanced Understanding**
   - Yang et al., 2019
   - XLNet explored alternative pretraining strategies, demonstrating the effectiveness of autoregressive modelling. Its success highlighted the importance of innovation in BERT-related research, which ReBERT embodies.

5. **Interpretability in BERT and its Derivatives**
   - Various authors
   - As BERT models grew in complexity, so did the need for interpretability. Numerous studies explored techniques to make these models more transparent, aligning with ReBERT's emphasis on interpretability.

6. **Efficiency Optimization for Large Models**
   - Various authors
   - In response to concerns about the resource-intensive nature of large models, researchers have been investigating various efficiency optimization techniques. ReBERT incorporates some of these optimizations, such as architectural refinements and training strategies.

7. **Fine-Tuning in NLP**
   - Various authors
   - Fine-tuning pretrained models for specific tasks is a common practice. ReBERT streamlines this process by introducing task-aware pretraining, thus reducing the gap between pretraining and downstream tasks.

8. **Multimodal and Cross-Modal Learning**
   - Various increasing importance of multimodal data analysis, models like ReBERT that support multimodal integration align with the evolving trends in NLP research.

9. **Domain Adaptation in NLP**
   - Various authors
   - Adapting pretrained models to domain-specific tasks is a crucial challenge in NLP. ReBERT's flexibility in fine-tuning for specific domains mirrors the broader need for domain adaptation techniques.

10. **Explainable AI (XAI) in NLP**
    - Various authors
    - Model interpretability and explainability have gained prominence in NLP. ReBERT's interpretability features contribute to the growing field of explainable AI, making it a noteworthy addition to the landscape.

11. **Multilingual BERT Variants**
    - Various authors
    - Another significant development in the world of BERT models is the creation of multilingual variants. These models, like mBERT and XLM-R, aim to support multiple languages, bridging language barriers and opening up NLP applications for a global audience. ReBERT's adaptability and potential for fine-tuning across languages align with this trend, demonstrating its versatility in multilingual contexts.

12. **Ethical Considerations and Bias in NLP**
    - Various authors
    - As the use of large language models like BERT became widespread, concerns related to ethical issues and biases in NLP models gained prominence. Researchers and practitioners began to focus on mitigating biases and ensuring fairness in AI systems. ReBERT's transparent architecture and interpretability features offer opportunities to address bias-related challenges and promote ethical AI practices.

13. **Transfer Learning in NLP**
    - Various authors
    - Transfer learning, exemplified by models like BERT, has become a fundamental concept in NLP. ReBERT's ability to provide task-aware pretraining further advances the field's understanding of how to leverage pretrained models effectively, setting a precedent for future research in transfer learning.

14. **Continual Learning and Lifelong Learning**
    - Various authors
• NLP researchers have been exploring ways to enable models to learn continually, adapting to new tasks and domains without catastrophic forgetting. ReBERT's fine-tuning adaptability aligns with the goals of continual learning, offering a flexible platform for lifelong learning scenarios.

15. Scaling Laws in NLP Models
• Various authors
• Researchers have observed scaling laws governing the performance and efficiency of NLP models as they increase in size. ReBERT's approach to efficiency optimizations contributes to understanding these scaling laws and potentially achieving more favourable trade-offs between model size and performance.

In conclusion, ReBERT represents a crucial milestone in the ongoing journey to refine and enhance BERT and similar models. Its contributions encompass task-aware pretraining, efficiency optimizations, enhanced interpretability, adaptability to various domains and languages, and alignment with ethical AI practices. In the ever-evolving landscape of NLP, ReBERT stands as a versatile and adaptable tool for researchers, developers, and organizations aiming to harness the power of large language models for a wide range of applications.

The literature study discussed in this research paper provides a comprehensive overview of the key developments in NLP leading up to ReBERT and highlights the model's significance in the broader context of NLP research and applications. As the field continues to evolve, ReBERT serves as both a testament to past achievements and a foundation for future innovations in natural language understanding and processing.

The overall architecture of our method can be viewed as a Transformer structure with multi-task learning. There are three output components, namely masked LM, entity classification and NER. With only the masked language model component, the model resembles BERT without the next sentence prediction task, and the entity classification task is added to enhance pretraining. While only NER outputs are yielded, the model is a sequence labeller for NER. We integrate entity-level information by extending the standard Transformer.

III. LEXICAL NORMALIZATION
1. Task:
Lexical normalization is the task of translating non canonical words into canonical ones. We illustrate it with the following example (Table 1). Given a noisy source sentence, our goal is to predict the gold canonical sentence.

<table>
<thead>
<tr>
<th>Noisy yea... @beautifulloser8 im abt ton type it uuuup !!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold yeah! .... @ beautifulloser8 i'm about to type it up!</td>
</tr>
</tbody>
</table>

Table 1: Noisy UGC example and its canonical form (Gold)

We make a few comments on this definition. First, lexical normalization assumes a certain degree of word level alignment between the noncanonical source text and the canonical one.

Second, language evolves. It varies across domain, communities and time, specifically online (Jurafsky, 2018). There is therefore no universal definition of what is a canonical form and what is not. In the context of NLP, this means that we have to set conventions and define what we consider as canonical. In our case, the task is made less complicated as we are tied to the conventions set by our training data set.

Finally, to grasp the complexity of such a task, we list and illustrate non exhaustively the sort of linguistic phenomenon that lexical normalization of UGC involves. Lexical normalization involves handling the following cases:
• spelling errors: making in making
• internet Slang: lmfao
• contraction: lil for little
• abbreviation: 2nite for tonight
• phonetics: dat for that

It also involves detecting that the following should be untouched Given a noisy source sentence, our goal is to predict the gold canonical sentence.

2. Data:
We base all our experiments on the WNUT data released by Baldwin et al. (2015). This dataset includes 2950 noisy tweets for training and 1967 for test. Out of the 44,385 training tokens, 3,928 require normalization leading to an unbalanced data set. Among those 3,928 noisy tokens, 1043 are 1-to-N (i.e. single noisy words that are normalized as several words) and 10 are N-to-1 cases (i.e. several noisy words that are normalized as single canonical words).
As highlighted before, our framework is more challenging than the standard approach to normalization, illustrated by the 2015 shared task that usually authorizes external UGC resources. As our goal is to test the ability of BERT, a model trained on canonical data only, we restrain ourselves to only using the training data as examples of normalization and nothing more.

Our work is therefore to build a domain transfer model in a low resource setting.

IV. NORMALIZATION WITH BERT

1. BERT:
   We start by presenting the components of BERT that are relevant for our normalization model. All our work is done on the released base version.

2. Word Piece Tokenization:
   BERT takes as input sub-word units in the form of Word Piece tokens originally introduced in Schuster and Nakajima (2012). The Word Piece vocabulary is computed based on the observed frequency of each sequence of characters of the corpus BERT is pre-trained on: Wikipedia and the Book Corpus. It results in a 30 thousand tokens vocabulary. We will refer to the process of getting Word Piece tokens from word tokens simply as tokenization for brevity.

   Reusing BERT, in any way, requires to use its original Word Piece vocabulary. In the context of handling non-canonical data, this is of primary importance. Indeed, frequent tokens in our non-canonical data set might not appear in the vocabulary of BERT and therefore will have to be split. For example, the word lol appear more than 222 times in the original lexnorm15 dataset (More than the word like that appears 187 times). Still, it is not in BERT-base WordPiece vocabulary. For tokenization of Workpiece’s, we follow the implementation found in the hugging face pytorchpretrained-BERT project 4. It is implemented as a greedy matching algorithm. We write it in pseudocode in Algorithm 1.

3. Masked Language Model:
   We now present one crucial aspect of BERT architecture. It was trained jointly on two objectives: On next sentence prediction on the one hand. On the other hand, it was trained on.

   \[
   \text{Vocabulary} = \text{Bert WordPiece Vocabulary; in it start=0, string=word, wordPieceList = list();}
   \]

   \[
   \text{while string not empty do}
   \]

   \[
   \text{substring:=string[start:];}
   \]

   \[
   \text{while substring not empty do}
   \]

   \[
   \text{if substring in Vocabulary then}
   \]

   \[
   \text{wordPieceList :=}
   \]

   \[
   \text{wordPieceList U [substring]}
   \]

   \[
   \text{break loop}
   \]

   \[
   \text{else}
   \]

   \[
   \text{substring:= substring[:-1]}
   \]

   \[
   \text{end}
   \]

   \[
   \text{end}
   \]

   \[
   \text{start := start + length(substring)}
   \]

   \[
   \text{Algorithm 1: Greedy WordPiece tokenization}
   \]

   \[
   \text{Result: wordPieceList}
   \]

   Note: Tokenizing words into wordpiece tokens, by matching in an iterative way from left to right, the longest sub-string belonging to the wordpiece vocabulary Masked Language Model (MLM). As we frame our normalization task very closely to it, we describe MLM briefly. For each input sequence, 15% of the WordPiece tokens are either replaced with the special token [MASK] (80% of the time), replaced by a random token (10% of the time) or untouched (10% of the time). BERT is trained by predicting this portion of token based on the surrounding context.

4. Fine-Tuning BERT for Normalization:
   We now present the core of our contribution. How to make BERT a competitive normalization model? In a nutshell, there are many ways to do lexical normalization. Neural models have established the state-of-the-art in the related Grammatical Error Correction task using the sequence-to-sequence paradigm (Sutskever et al., 2014) at the character
level. Still, this framework requires a large amount of parallel data. Our preliminary experiments showed that this was unusable for UGC normalization. Even the use a powerful pre-trained model such as BERT for initializing an encoder-decoder requires the decoder to learn an implicit mapping between noisy words and canonical ones. This is not reachable with only 3000 sentences.

We therefore adapted BERT in a direct way for normalization. As described in section 4, BERT Masked Language Model ability allows token prediction. Simply feeding the model with noisy tokens on the input and fine-tuning on canonical token labels transform BERT into a normalization model. There are two critical points in doing so successfully. The first is that it requires WordPiece alignment. The second is that it requires careful fine-tuning.

5. **Wordpiece Alignment**:
We have in a majority of cases, as described in section 3.2, word level alignment between non canonical and canonical text. Still, the dataset also includes words that are not aligned. For 1-to-N cases we simply remove the spaces. As we work at the WordPiece level this does not bring any issue. For N-to-1 cases (only 10 observations), by considering the special token” j” of the lexnorm15 dataset as any other token, we simply handle source multi-words as a single one, and let the wordpiece tokenization splitting them.

We frame normalization as a 1-to-1 WordPiece token mapping. Based on the word level alignment, we present two methods to get WordPiece alignment: an Independent Alignment approach and a Parallel Alignment one.

6. **Independent Alignment**
We tokenize noisy words and non-noisy ones independently (cf. algorithm 1). By doing so, for each word we get non-aligned WordPiece tokens. We handle it in three ways:
- If it is the same number of WordPiece tokens, we keep the alignment as such
- If there are more tokens on the target side, we append the special token [MASK] on the source side. This means that at training time, we force the model to predict a token.
- If there are more tokens on the source side, we introduce a new special token [SPACE].

An alignment example extracted from lexnorm15 can be found in table 2. Briefly, we can point some intuitive pros and cons of such an alignment method. On the one hand, applying tokenization that was used in pre-training BERT means that the sequence of tokens observed during training should be modelled properly by BERT. This should help normalization. On the other-hand, we understand that learning normalization in this way requires (as potentially many [MASK] will be introduced) abstracting away from the raw tokens in understanding the surrounding context. This should make normalization harder. We will see in section 5 that despite its simplicity, such knowing what u sayin normalized as yeah i’m already knowing what you saying an alignment allows our model to reach good performances.

<table>
<thead>
<tr>
<th>noisy</th>
<th>canonical</th>
</tr>
</thead>
<tbody>
<tr>
<td>ye</td>
<td>yeah</td>
</tr>
<tr>
<td>#a</td>
<td>[SPACE]</td>
</tr>
<tr>
<td>im</td>
<td>i</td>
</tr>
<tr>
<td>[MASK]</td>
<td>m</td>
</tr>
<tr>
<td>already</td>
<td>already</td>
</tr>
<tr>
<td>knowing</td>
<td>knowing</td>
</tr>
<tr>
<td>wa</td>
<td>what</td>
</tr>
<tr>
<td>##t</td>
<td>[SPACE]</td>
</tr>
</tbody>
</table>

Table 2: Independent Alignment of yea im already

7. **Parallel Alignment**
We enhance this first approach with a parallel alignment method, described in Algorithm 2. Our goal is to minimize the number of [MASK] and [SPACE] appended into the source and gold sequences. Therefore, for each word, we start by tokenizing in WordPieces the noisy source word. For each WordPiece met, we start the tokenization on the gold side, starting and ending from the same character positions. As soon as we tokenized the entire gold sub-string, we switch to the next noisy sub-string and so on. By doing so, we ensure a closer alignment at the WordPiece level. We illustrate on the same example this enhanced parallel alignment in Table 3.

We highlight two aspects of our alignment techniques. First, introducing the special token [SPACE] induces an architecture change in the MLM head. We detail this in section 8-(A). Second, appending the extra token [MASK] on the source side based on the gold sequence induces a discrepancy between training and testing. Indeed, at test time, we do not have the information about whether we need to add an extra token or not. We describe in section 8-(B) how we extend BERT’s architecture with the addition of an extra classification module to handle this discrepancy.
8. Architecture Enhancements:
A. Enhancing BERT MLM with [SPACE]

In order to formalize lexical normalization as a token prediction we introduced in previous section the need for a new special token [SPACE]. We want our normalization model to predict it. We therefore introduce a new label in our output WordPiece vocabulary as well as a new vector in the last SoftMax layer. We do so in a straightforward way.

Result: wordPListNoisy

Note: Tokenizing noisy tokens and canonical tokens in wordpieces in parallel to minimize the number of appended [MASK] and [SPACE]. by appending to the output matrix, a vector sampled from a normal distribution.

Algorithm 2: Parallel WordPiece tokenization

B. #Next [MASK] predictor

As we have described, alignment requires in some cases the introduction of [MASK] tokens within the source sequence based on the gold sequence. We handle the discrepancy introduced between training and testing in the following way. We add an extra token classification module to BERT architecture.

Table 3: Parallel Alignment of yea im already knowing wat u sayin normalized as yeah i’m already knowing what you saying

This module takes as input BERT last hidden state of each WordPiece tokens and predict the number of [MASK] to append next. In table 4, we illustrate the training signal of the overall architecture. It takes noisy WordPiece tokens as
input. As gold labels, it takes on the one side the gold WordPiece tokens and on the other side the number of [MASK] to append next to each source WordPiece tokens.

Table 4: Parallel Alignment of yeah i’m already knowing what you saying normalized as yeah i’m already knowing what you saying with gold number of next masks for each source token

At test time, we first predict the number of next masks to introduce in the noisy sequence. We then predict normalized tokens using the full sequence. This #next mask prediction module exceeds the context of normalization. Indeed, it provides a straightforward way of performing data augmentation on any Masked Language Model architecture. We leave to future work the investigation of its impact beyond lexical normalization.

V. DISENTANGLED ATTENTION: A TWO-VECTOR APPROACH TO CONTENT AND POSITION EMBEDDING

For a token at position i in a sequence, we represent it using two vectors, \{Hi\} and \{Pi|j\}, which represent its content and relative position with the token at position j, respectively. The calculation of the cross-attention score between tokens i and j can be decomposed into four components as:

\[
A_{i,j} = \frac{Q_i^c K_j^c}{\sqrt{d}} V_c + Q_i^c K_j^r \delta(i,j) + Q_i^r K_j^r \delta(i,j) + P_{i,j}^r P_j^r
\]

That is, the attention weight of a word pair can be computed as a sum of four attention scores using disentangled matrices on their contents and positions as content-to-content, content-to-position, position-to-content, and position-to-position.

Existing approaches to relative position encoding use a separate embedding matrix to compute the relative position bias in computing attention weights (Shaw et al., 2018; Huang et al., 2018). This is equivalent to computing the attention weights using only the content-to-content and content-to-position terms in equation. We argue that the position-to-content term is also important since the attention weight of a word pair depends not only on their contents but on their relative positions, which can only be fully modeled using both the content-to-position and position-to-content terms.

Since we use relative position embedding, the position-to-position term does not provide much additional information and is removed from equation 2 in our implementation.

Taking single-head attention as an example, the standard self-attention operation (Vaswani et al., 2017) can be formulated as: We can represent the disentangled self-attention with relative position bias as equation where Qc, Kc and Vc are the projected content vectors generated using projection matrices Wq,c, Wk,c. Wv,c \in \mathbb{R}^{d \times d}$ respectively, $P \in \mathbb{R}^{2k \times d}$ represents the relative position embedding vectors shared across all layers (i.e., staying fixed during forward propagation), and Qr. and Kr. are projected relative position vectors generated using projection matrices Wq,r, Wk,r \in \mathbb{R}^{d \times d}$.
VI. FUTURE SCOPE:
The future scope would depend on several factors, including its capabilities, performance, and adoption within the NLP community. Here are some general considerations for the future of enhanced BERT-like models:

- Improved Performance: Enhanced BERT models are expected to push the boundaries of NLP tasks by achieving better results on benchmarks like GLUE, SQuAD, and others. Continued research and fine-tuning can lead to even higher performance.
- Multimodal Understanding: The integration of vision and language models (such as vision transformers or VITs) into BERT-like architectures could enable these models to understand and process both text and images, opening up new possibilities for applications like image captioning and visual question answering.
- Cross-Lingual Understanding: Expanding the scope of enhanced BERT models to understand and generate text in multiple languages with high accuracy will be a valuable direction, especially for global applications.
- Low-Resource Languages: Addressing the needs of low-resource languages and dialects by fine-tuning enhanced BERT models could help bridge the language gap and make NLP applications more inclusive.
- Pre-training on Diverse Data: Training enhanced BERT models on more diverse and representative datasets can improve their generalization to various domains and reduce biases.
- Interpretability: Research into making enhanced BERT models more interpretable and explainable will be crucial, especially in applications that require transparency and trust.
- Customization: Enhancing the ability to fine-tune BERT models for specific industries or domains, such as healthcare, finance, or law, could lead to tailored solutions for these sectors.
- Deployment and Efficiency: As models become more sophisticated, efforts to optimize their deployment in real-world applications will be essential, including model compression, quantization, and efficient hardware support.
- Ethical Considerations: Ethical considerations, such as addressing bias and fairness, privacy, and responsible AI, will continue to be important aspects of developing and using enhanced BERT models.
- Collaboration and Open Source: Collaboration among researchers and open-source development will likely play a significant role in advancing enhanced BERT models, similar to the development of the original BERT model.

VII. CONCLUSION:
General pre-trained language model has demonstrated their ability to improve Natural Language Processing systems for most tasks on canonical data. In our work, we demonstrated that they can also be useful in non-canonical noisy text in low-resource setting. We hope that this work will pave the way for future research in modelling non-canonical textual data.

In conclusion, the success and impact of ReBERT as an enhanced BERT model would depend on its specific features, capabilities, and contributions to the NLP landscape. To form a definitive conclusion, it's essential to review the latest research and developments related to ReBERT and observe its adoption and performance in real-world applications.

REFERENCES:


