Guided Research - Technical Report: Masking Strategies for Efficient Deep Language Model Pre-Training

Marcel Braasch
Technical University of Munich
marcelbraasch@gmail.com

Lütfi Kerem Şenel
CIS, LMU Munich
lksenel@cis.lmu.de

Abstract

In this guided research project we examined working with deep language modelling pre-training approaches such as BERT and RoBERTa in an academic setting. Working with these models entails days of computations usually using specialized and expensive hardware. We followed Izsak et al. (2021)'s budget BERT pre-training recipe in order to run experiment with a series of masking strategies for pre-trained language modelling. In this work, we compare classic masked language modelling approaches based on uniform masking with more informed methods. Our tool of choice is a statistical measure called pointwise mutual information (PMI). We showcase an extension of existing PMI-based masking strategy, conduct a theoretical analysis and present empirical results of our approach. Unfortunately, the method tried was not able to beat the strong baselines, however, we aim to correct flaws in subsequent work.

1 Introduction

Deep pre-trained language models equipped with Transformers currently dominate the NLP landscape (Devlin et al., 2018; Radford et al., 2019; Vaswani et al., 2017). They follow a clear paradigm. Models are pre-trained in a self-supervised fashion given an abundance of textual data (Devlin et al., 2018). Next, they are fine-tuned and evaluated on downstream benchmarks such as GLUE and SQuAD (Rajpurkar et al., 2016; Wang et al., 2018).

Many resources currently flow into scaling models and data to enormous sizes (Wei et al., 2021; Chowdhery et al., 2022). While these approaches yield stunning results and are proof of excellent engineering efforts, they do not focus on increased model efficiency.

The rationale for this work is therefore assessing approaches to create more efficient and more effective deep learning methods. The cornerstone is motivated by (**1**) ecological concerns, (**2**) ethical issues, and (**3**) possible obstacles hindering research. Current methods are extremely resource hungry, among the most expensive across all deep learning disciplines (Sharir et al., 2020). This implies that only a selected group of people is able to work on these methods posing major ethical concerns in terms of accessibility. Further, crucial directions in current research practices in artificial intelligence are hindered (Strubell et al., 2019).

This work is structured as following. In Section 2 we give a broad overview of the model in use. Once the foundations are laid current practices in masking strategies are discussed in Section 3. Section 4 deep-dives into a statistical measure called pointwise mutual information. In Section 5 we discuss our masking approach. Section 6 provides implementation details. In Sections 7-9 we discuss results, challenges and draw a conclusion, respectively.

2 Masked Language Modelling

There are many paradigms in deep language modelling, among the most popular being auto-regressive modelling (Radford et al., 2019; Yang et al., 2019), sequence-to-sequence modelling (Sutskever et al., 2014; Le et al., 2017), text-to-text modelling (Raffel et al., 2019) and masked language modelling (MLM) (Devlin et al., 2018; Liu et al., 2019). This work focuses exclusively on the later and its masking strategies in the pre-training setting. Thus MLMs are the subject of this section. The model used in this is BERT with a few changes adapted from RoBERTa.

Architecture BERT's architecture is simple as L Transformer encoders are simply stacked on top of each other. An exact examination of Transformers is neglected for the sake of conciseness. For more details we refer to Vaswani et al. (2017). Our exact design choices can be found in Section 6.

Objective The MLM objective of BERT follows a self-supervised auto-encoding idea. Training is conducted on a large corpus $\mathcal{C}$ of textual data. Assume a sequence $X = (x_1, \ldots, x_N) \in \mathcal{C}$ where each x_i corresponds to a token in the sequence. A subset $Y \subset X$ of tokens is sampled and replaced by a [MASK]-token. Tokens and the respective vocabulary are typically generated by tokenization algorithms such as WordPiece or Byte Pair Encoding (Wu et al., 2016; Sennrich et al., 2015). This procedure enhances the model with handling out-of-vocabulary words. For example, the word "oxygen" could be broken up into the tokens "o-xy-gen", depending on the corpus statistics.

Given the unmasked tokens in the sequence, the MLM objective is then to predict the masked sample. Formally, this can be expressed as

$$log\, P(\bar{x}_{\mathcal{I}}|x;\Theta) = \sum_{k\in\mathcal{I}} log\, P(\bar{x}_k|x;\Theta) \qquad (1)$$

where $\mathcal{I}$ is the index set (e.g. $\mathcal{I} = \{3,5\}$), $\bar{x}_{\mathcal{I}}$ refers to the masked tokens to predict (e.g., $x_{\mathcal{I}} = \{x_3, x_5\}$) and x simply refers to the entire sequence (e.g., $x = (x_1, x_2, [M], x_4, [M])$.

The original BERT model is jointly trained on a second task called next sentence prediction (NSP). A sentence A is chosen and the model predicts whether another sentence B follows A. With a probability of 50% B is either a random sentence sampled from the corpus or it is the actual next sentence. However, there is much evidence that the NSP objective does not benefit, sometimes even harm, training (Izsak et al., 2021; Joshi et al., 2020; Liu et al., 2019). Therefore many approaches (including our implementation) does not regard the objective at all.

Batching Strategy BERT is originally trained on two sentences at a time. RoBERTa modified this strategy with the aim of increasing sampling efficiency. In particular, instead of training on two sentences the authors of RoBERTa examined two new strategies. The full-document strategy simply retrieves contiguous spans of sentences until the token limit is reached. If a document is shorter than the token limit, another document is appended in the same manner. The document-based strategy follows the same idea except that the sample creation is terminated once a document has finished, and no other document is added.

3 Masking Strategies

In the following an overview of current deep language modelling masking strategies is given.

3.1 Random Masking

In the original BERT implementation the random masking strategy was employed. Given a training sequence 15% of all tokens are masked. Out of these, BERT follows an 80-10-10 strategy, that is, 80% of tokens are replaced by the [MASK]-token, 10% are corrupted by a random token and 10% are left unchanged. The issue with this implementation is that some tokens are very easy to predict. Some words are broken into sub word tokens according to a tokenizer like WordPiece (Sennrich et al., 2015). Often these words are rare words, or plural forms of which have not been seen during tokenizer training. The surrounding context of a token is simply giving away what a specific token likely would be. Assuming the masked sequence to be "the cat [MASK] are running", the token "are" is a strong hint that the missing token is simply the plural-s. In the masked sequence "Ei-gen-[MASK]-lue" a model would quickly learn utilize the immediate context to predict the token "va".

3.2 Whole Word Masking

The whole word masking strategy tries to overcome this downfall by masking entire words which were previously broken into sub words. Assume the input text "the man jumped up , put his basket on phil am mon ' s head". While random masking would produce "[MASK] man [MASK] up , put his [MASK] on phil [MASK] mon ' s head" whole word masking will mask according to the following pattern: "the man [MASK] up , put his basket on [MASK] [MASK] [MASK] ' s head".

3.3 Span Masking

Span masking was first introduced by Joshi et al. (2020) and can be understood as an extension of whole word masking. When masking one word, it is likely that the immediate context surrounding the word inherits more information about the target than words further away. Therefore, masking more than just a word but parts of its direct context is an effective measure of increasing task difficulty and thus learning effectiveness. For each sample, the span length l (i.e., the amount of consecutive words masked) will be sampled according to the geometric distribution $l \sim geom(p = 0.2)$. To not

mask an entire sequence without any signals the distribution is clipped at 10 tokens. This produces spans of an average size of 3.8.

3.4 PMI-Based N-Gram Masking

Levine et al. (2020) introduced an alternative to random uniform and span masking. A statistical measure called pointwise mutual information (PMI) forms the foundation of their approach. Here, PMI(w_1, w_2) refers to the mutual information between two random variables w_1 and w_2 *anywhere* in the sequence. It is a measure of how much information one obtains about the realization of w_1, knowing w_2. For a thorough overview of PMI we refer to Section 4.

Important to note is that in Levine et al. (2020)'s approach, the PMI between two words w_1 and w_2 are regarded as a bi-gram limiting the context window to a step size of one disregarding all other possible combinations with words beyond that. The authors show an intuitive formulation of the problem naively (hence "N"-PMI) expanded to arbitrary n-grams. This formulation yields

$$\text{N-PMI}_n(w_1, \ldots, w_n) = log \frac{P(w_1, \ldots, w_n)}{\prod_{i=1}^{n} P(w_i)}. \tag{2}$$

The authors show a clear limitation to the approach. One can show, e.g., that N-PMI$_n$ is equivalent to PMI(w_1, w_2)+$log \frac{P(w_1,w_2,w_3)}{P(w_1,w_2)P(w_3)}$ which means that even though one is looking for a high PMI tri-gram the bi-gram is building up the baseline of the expression, distorting the ranking heavily.

To overcome this another measure is introduced which defines the PMI value by its weakest link amongst all n-grams. Formally, this is expressed as

$$\text{W-PMI}_n(w_1, \ldots, w_n) = \min_{\sigma \in \mathcal{S}} log \frac{p(w_1, \ldots, w_n)}{\prod_{s \in \sigma} P(s)} \tag{3}$$

where $\mathcal{S} = \text{seq}(w_1, \ldots, w_n)$ is the set of all possible contiguous segmentations of the n-gram.

3.5 Increased Masking Percentage

In a fairly recent paper, Wettig et al. (2022) examine the effect of increasing the 15% default masking percentage of current MLMs. Surprisingly, they find that increasing masking probability to 40% increases downstream model performance for many tasks. In addition to that, they claim that prediction on a larger subset of masked tokens increasingly becomes competitive with more sophisticated strategies such as PMI-based n-gram masking and span masking. Clearly, once a higher portion of tokens is masked, tokens which will be selected by a PMI measure or selected by spanning are more likely to be covered. A particular selection criterion is not as important anymore.

Performance Comparison A performance comparison of these strategies can be found in Figure 1. One can see that Levine et al. (2020)'s PMI masking approach outperforms other approaches in most settings. Importantly to note is that the effectiveness of PMI increases when scaling the dataset in use.

4 Pointwise Mutual Information

Pointwise mutual information (PMI) lies at the core of our attempts of an advanced masking strategy. Therefore we will shed light on diverse variations and discuss these thoroughly.

4.1 Classic PMI

PMI refers to a measure of co-occurrence of events in a sequence of random variables (Levine et al., 2020; Role and Nadif, 2011). In an NLP setting, PMI expresses how much two words' co-occurrence deviates from what we would expect under the independence assumption (Role and Nadif, 2011). From an information theoretic perspective one is asking how much we know about one random variable, if the other is given. Formally this can be expressed as

$$\text{PMI}(w_1, w_2) = log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \tag{4}$$

where the factorized probabilities are the normalized word frequencies and the joint probability account for the probability of w_1 and w_2 co-occurring in a fixed window. Parameters are simply estimated through maximum likelihood for both the factorized and joint probabilities.

To create an intuition about the measure we examine three cases.

Case 1 If $P(w_1, w_2) = P(w1)P(w2)$ then $PMI(w_1, w_2) = log1 = 0$. In other words, words w_1 and w_2 are independent. Masking these pairs is usually not of interest.

BERT Base with different maskings	SQuAD2.0 EM	SQuAD2.0 F1	RACE Acc.	GLUE Avg
1M training steps on WIKIPEDIA+BOOKCORPUS*(16G):*				
Random-Token Masking	76.4/–	79.6/–	67.8/66.2	83.1/–
Random-Span Masking	77.1/–	80.3/–	68.6/66.9	83/–
Naive-PMI-Masking	78.2/–	**81.3**/–	69.7/67.8	**84.1**/–
PMI-Masking	**78.5**/–	**81.4**/–	**70.1/68.4**	**84.1**/–
2.4M training steps on WIKIPEDIA+BOOKCORPUS*(16G)*				
Random-Span Masking	79.7/80.0	82.7/82.8	71.9/69.5	**84.8**/79.7
Naive-PMI-Masking	**80.3**/80.2	**83.2**/83.2	71.7/69.8	84.5/80.0
PMI-Masking	**80.2/80.9**	**83.3/ 83.6**	**72.3/70.9**	84.7/**80.3**
2.4M training steps on WIKIPEDIA+BOOKCORPUS+OPENWEBTEXT*(54G):*				
Random-Span Masking	80.1/80.4	83.2/83.3	74.0/72.2	85.1/80.1
Naive-PMI-Masking	80.4/80.0	83.3/83.0	73.9/71.4	85.6/80.3
PMI-Masking	**80.9/82.0**	**83.9/84.9**	**74.8/73.2**	**86.0/80.8**

Figure 1: Dev/Test performance on the SQuAD, RACE, and GLUE benchmarks using various masking strategies employed with BERT Base. Table according to (Levine et al., 2020). EM means exact match.

Case 2 If $P(w_1, w_2) < P(w1)P(w2)$ then $\frac{P(w_1,w_2)}{P(w1)P(w2)} < 1$ and thus $PMI(w_1, w_2) = log\frac{P(w_1,w_2)}{P(w1)P(w2)} < 0$. PMI will be low, which means w_1 and w_2 rarely occur together. Though this is not equivalent to independence, it can be interpreted as a weaker version of such in the sense that knowing w_1 does not give away much information about w_2.

Case 3 If $P(w_1, w_2) > P(w1)P(w2)$ then $\frac{P(w_1,w_2)}{P(w1)P(w2)} > 1$ and thus $PMI(w_1, w_2) = log\frac{P(w_1,w_2)}{P(w1)P(w2)} > 0$. PMI will be high, meaning that if w_1 occurs, there is a (high) chance w_2 will be present. Typically, these pairs are the ones we are looking for since they indicate (as the name suggests) most mutual information.

Our is aim to prevent the model using trivial statistical clues to conduct correct predictions. E.g., predicting the masked token in "I am considering moving to [MASK] York" is requires almost no effort since "New" and "York" will have very high PMI and the model does not need to use any other information from the context. This way we force the model to regard a more global view and take more context into consideration.

4.2 PMI Variations

A known downfall of classic PMI is a bias towards rare words. Words which are very rare but co-occur frequently tend to get high PMI. To mitigate this bias, we can smoothen the distribution of the joint probability of two co-occurring words. A typical measure to achieve this is the so called PMI^k-family (Daille, 1994), defined as

$$PMI(w_1, w_2)^k = log\frac{P(w_1, w_2)^k}{P(w1)P(w2)} \quad (5)$$

where k is used to dampen the influence of the co-occurring moving scoring mass in favor of the factorized probabilities. This factor allows other terms in the ranking to surpass the specific terms.

Role and Nadif (2011) empirically showcase how the PMI^k family solves this discrepancy by computing the PMI of "football" with all other words in a corpus. The results can be inspected in Table 1. One can see that moving from the classic PMI score to a more general view helps create a more "global picture". Learning good representations of specific words require vast amounts of data. It is known that BERT pre-training suffers from the low-occurrence of rare words (Wu et al., 2020; Schick and Schütze, 2020). Therefore, one could argue that masking many rare words which often co-occur benefits pre-training. However, as discussed in Section 7 the PMI scores tend to select words which are *too* specific, masking named entities, proper nouns and very specific terms which may not yield the best training signal. Interestingly to show, Eq. (5) is equivalent to $PMI(w_1, w_2) + (k-1) \cdot logP(w_1, w_2)$ which shows that PMI^k in fact decreases the effect of low-occurring events by a linear factor of $k-1$. Note that $log\ x \in (-\infty, 0)$ for $x \in (0, 1)$, thus the score decreases with k increasing.

PMI	Word	PMI^3	Word
5.581	midfielder	-19.840	league
5.575	midfielders	-20.667	clubs
5.543	cornerbacks	-20.915	england
5.530	goalkeepers	-21.326	players
5.529	safeties	-21.922	season
5.475	linebackers	-22.043	college
5.475	striker	-22.224	club

Table 1: PMI scores for the word "football". We can observe that both measures produce very different ranking. While PMI associates very specific concepts with the target word, PMI^3 creates a much more general view. (Role and Nadif, 2011).

Besides ordinary PMI and PMI^k there are other variants of interest. One of them being Normalized PMI (NPMI) defined as

$$\text{NPMI}(w_1, w_2) = \frac{PMI(w_1, w_2)}{logP(w_1, w_2)} \quad (6)$$

which results in a value range of $[-1, 1]$ making the measure more interpretable. However, empircal examinations by (Role and Nadif, 2011) revealed that NPMI does not mitigate the low-frequency bias induced by PMI.

In addition to that, positive PMI (PPMI) can be defined which simply disregards all negative terms. This measure may help to randomize low-occurrence appearances and smoothen the distribution in the tails. However, as Table 1 suggests, PMI values can be in the negative domain, especially when utilizing a measure from the PMI^k family. Formally, PPMI can be defined as

$$\text{PPMI}(w_1, w_2) = max(PMI(w_1, w_2), 0) \quad (7)$$

and should be able to be combined with all other PMI measures. Unfortunately, we have not yet assessed the affects of this measure.

5 Global PMI Masking

In this section we give an introduction to our method which is built upon the previously discussed PMI-based n-gram masking. The rationale for our approach is manifold and will be discussed in the following.

Motivation Even though words may be far away in terms of position they can be highly correlated.

A class of words which showcase this are particle verbs. Consider the sentence "Can you turn the music which is playing on the radio down?". Clearly, the words "turn" and "down" form one entity, however, cannot be captured by an n-gram approach. Evidently, relative clauses play their part in this type of construction.

Another example forms a prominent phenomenon from the domain of syntax and semantics. Typically, though dependent on the language and context, sentences incorporate a subject which defines the main concept of the sentence. This is called the topic. Its underlying process is often referred to as topicalization (Miyagawa, 2017). A simple example could be "This book is really a masterpiece I enjoy reading!". Clearly, "book" forms the center and topic of this sentence. Predicting the word "reading" will not be very challenging if the word "book" is known. If it were masked as well, the task would be much more challenging.

One can find many of these types of examples where words stay in an obvious relationship to one another, however, are very far away and will not be captured by n-grams. Since sampling on document level (and not on sentence level, as the original BERT implementation suggests) is conducted, taking up a global view seems reasonable.

The main difference to the approach discussed in Section 3.4 is that we do not regard n-grams only but look for high PMI values across all words in the entire context. PMI-based n-gram masking is mainly seeking to mask entire entities which often co-occur.

There are two intuitive strategies to implement this approach.

Random PMI Select one random word and calculate PMI with all other words. The word with the highest PMI will be masked alongside the random word. This can be extended to arbitrary amounts of words. Once two words are chosen, calculate PMI of both words with respect to all other words. The word with the highest overall PMI will be chosen.

Maximum PMI Calculate PMI between all pairs of words. Choose the pair with the highest PMI. Repeat until masking budget is met.

Random PMI with two words is the strategy used in our experiments. Our rationale behind the design was that introducing randomness approximately retains the original distribution of masked words. It could be that the Maximum PMI strategy is too biased towards low-occurrence words as these are often the words with high PMI. However, due to

	WNLI Acc.	QNLI Acc.	QQP Acc./F1	RTE Acc.	SST-2 Acc.	MRPC Acc./F1	CoLA MCorr	STS-B. P/S Corr	Avg.
Random-15-80	40.8	86.9	90.3/86.9	53.4	88.9	76.0/83.7	37.7	85.7/83.4	74.0
Random-15-100	37.7	86.9	90.3/86.9	53.3	88.9	76.0/83.7	37.7	**85.7/85.4**	73.9
WWM-15-80	39.4	**89.0**	**90.6/87.4**	**59.2**	89.3	74.5/82.1	**43.3**	84.2/84.0	74.8
WWM-15-100	38.0	88.2	**90.6/87.4**	58.1	**89.8**	**80.9/86.7**	37.2	85.8/85.4	**75.3**
RPMI-15-80	15.5	85.6	90.2/86.8	54.1	89.2	70.8/79.7	34.0	81.6/81.4	69.9
RPMI-15-100	35.8	87.7	90.0/86.7	54.9	89.6	76.2/83.8	35.9	85.6/85.2	73.8
WWM-40-100	**45.1**	87.9	90.2/86.9	56.3	89.2	76.2/84.0	31.1	84.4/83.9	74.1

Table 2: Performance on GLUE evaluation sets. MCorr means Matthew's Correlation and P/S Corr mean Pearson's and Spearman's correlation, respectively.

the time and resource constrained setting we have not yet tested the Maximum PMI strategy.

6 Methods

As our backbone model we used an interpolation between BERT and RoBERTa following Izsak et al. (2021) which provide an efficient framework for training MLMs in a budgeted setting.

Model and Training The model architecture follows BERT-large as large models converge faster (You et al., 2019). The tokenizer in use is WordPiece with fixed vocabulary size of 30K (Sennrich et al., 2015). Instead of a batch size of 128 we train on batches of size 4096. Sequences are of length 128 instead of 512 increasing sample efficiency (since less padding tokens are used). The next sentence prediction task is completely dropped. The optimizer in use is AdamW (β_1=0.9, β_2=0.98, ϵ=1e-6, weight decay of 0.01) (Loshchilov and Hutter, 2017). Regular and attention dropout is employed with a rate of 0.1. Low-level optimizations are reached by using the software packages DeepSpeed (Rasley et al., 2020) for model training and APEX's LayerNorm[1] instead of the original implementation. For more details please refer to Izsak et al. (2021) as all parameters are left as is, if not otherwise mentioned.

Data The training corpus is pre-computed by duplicating the English Wikipedia 10 times. Due to budget constraints we decided not to train on BookCorpus but may extend in the future. Samples generated are on document level and randomly sliced per document. This ensures data variation. To calculate PMI scores, we calculated word counts and word co-occurrence over the entire corpus. As the co-occurrence calculations require immense amounts of main memory we neglect all word pairs which co-occur less than 10 times to save space.

Training Training was conducted in a budgeted environment. On a single RTX 3080 we trained for 48 hours. On a rack of 8 RTX 2080s we trained for 12 hours. On a rack of 8 RTX 1080s we trained for 48 hours. In all settings we trained for approximately 6000 training steps (according to Izsak et al. (2021)'s framework) accounting for 2.45 million seen samples.

Masking Strategies We generate 7 different sets of pre-computed datasets which are used to pretrain the model.

Random-15-80 BERT's classic uniform masking strategy with 15% sampling rate of which 80% are masked, 10% corrupted and 10% left unchanged.

WWM-15-80 Whole word masking with 15% sampling rate of which 80% are masked, 10% corrupted and 10% left unchanged.

RPMI-15-80 Random PMI strategy as explained in Section 5. Tokens are masked on the whole word level. The masking budget of 15% as well as the 80-10-10 strategy are kept. Word corruption and unmasking, however, are conducted on token level.

Random-15-100 Same as Random-15-80, except masking with 100% probability.

WWM-15-100 Same as WWM-15-80, except masking with 100% probability.

RPMI-15-10 Same as RPMI-15-80, except masking with 100% probability.

WWM-40-100 Following Wettig et al. (2022) we conduct WWM with a masking budget of 40%.

[1] https://github.com/NVIDIA/apex

Finetuning Downstream performance is evaluated through the GLUE benchmark assessed on 8 challenging natural language understanding tasks (Wang et al., 2018). We fine tune all models with the same set of parameters and forego any type of hyperparameter tuning due to resource constraints. Our hyperparameters are a learning rate of 5e-5, batch size of 32 and weight decay of 0.01. We train for 5 epochs and evaluate 50 times per training.

7 Evaluation

Our finetuning results on the GLUE benchmark can be seen in Table 2. As can be seen, both WWM baselines are the strongest among all methods tried. RPMI-15-100 gets close to them in the tasks of SST-2, STS-B and .

We noticed a downfall of our method when inspecting samples manually. Inspecting the following sequence one can notice two named entities split into the respective tokens according to the vocabulary: "was available to play for game 7 , vi ##gne ##ault chose to start lu ##ong ##o . he made]. Noticeably, with a probability of almost 100% the two named entities "vigneault" and "luongo" (among other very specific terms) will be selected in the PMI procedure. As discussed before in Section 4.2, selection of rare words is one of the shortcomings of classic PMI. Initially we assumed this bias may be helpful to create a more informed training setting. It seems, however, that words selected are too specific hindering the model to learn general concepts well.

Shifting our approach to a more general picture, i.e., employing measures from the PMI^k as well as combining these with NPMI and PPMI are reasonable next steps. We aim to review our work and keep working on improving the methods assessed.

8 Challenges

There were many challenges in this project. Setting up an appropriate training pipeline was our main concern for a really long time. Our aim was to "get our hands dirty" in a resource-hungry and complex pre-training settings. The resource hungriness we experienced indeed. Pre-training deep language models, even in the efficient setting suggested by Izsak et al. (2021), takes days of computation. This makes any sorts of assessments extremely time consuming and punishes errors with time lost. Besides, we underestimated time used to generate samples. Generating random samples on a high end CPU with 32 cores takes 8 hours. Generating the PMI dataset, takes 4.5 days (on a mid-range server with 1TB main memory to load the data).

9 Conclusion

In this report we broadly introduced the topic of masking strategies in masked language modelling (MLM). An information theoretic measure called pointwise mutual information (PMI) was thoroughly discussed. We implemented a selection algorithm based on PMI to create a more informed masking strategy. Our intention was to create a method requiring less compute time and achieving results on par with strong random baselines. Unfortunately, the method tried was not able to beat the baselines. With the advancement of the project we spotted flaws which we are seeking to correct in a follow up project, specifically our aim will be to asses the PMI^k-family further.

References

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Béatrice Daille. 1994. *Approche mixte pour l'extraction automatique de terminologie: statistiques lexicales et filtres linguistiques*. Ph.D. thesis, Ph. D. thesis, Université Paris 7.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to train bert with an academic budget. *arXiv preprint arXiv:2104.07705*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

An Nguyen Le, Ander Martinez, Akifumi Yoshimoto, and Yuji Matsumoto. 2017. Improving sequence to sequence neural machine translation by utilizing syntactic dependency information. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 21–29.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav

Shoham. 2020. Pmi-masking: Principled masking of correlated spans. *arXiv preprint arXiv:2010.01825*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Shigeru Miyagawa. 2017. Topicalization. *Gengo Kenkyu (Journal of the Linguistic Society of Japan)*, 152:1–29.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Francois Role and Mohamed Nadif. 2011. Handling the impact of low frequency events on co-occurrence based measures of word similarity. In *Proceedings of the international conference on Knowledge Discovery and Information Retrieval (KDIR-2011). Scitepress*, pages 218–223.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training nlp models: A concise overview. *arXiv preprint arXiv:2004.08900*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*.

Qiyu Wu, Chen Xing, Yatao Li, Guolin Ke, Di He, and Tie-Yan Liu. 2020. Taking notes on the fly helps bert pre-training. *arXiv preprint arXiv:2008.01466*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.