Training MLM models without softmax distribution

Karol Kaczmarek

Adam Mickiewicz University
Poznań

2023

Predicting Next Tokens - CLM

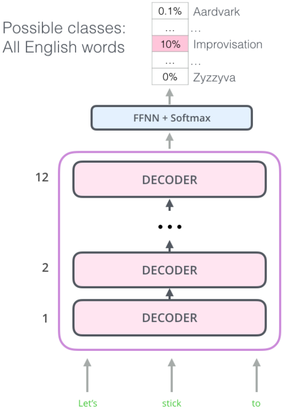


Image from: The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

Predicting Masked Tokens - MLM

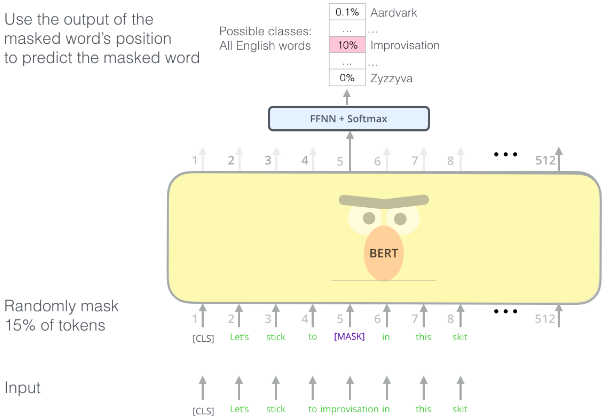
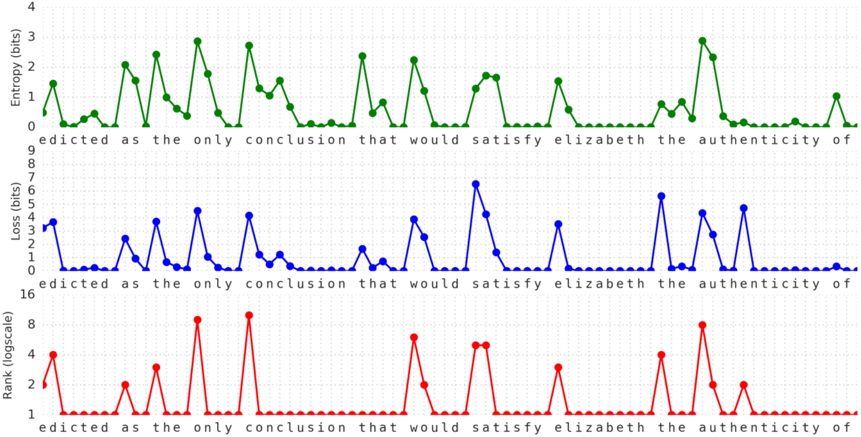


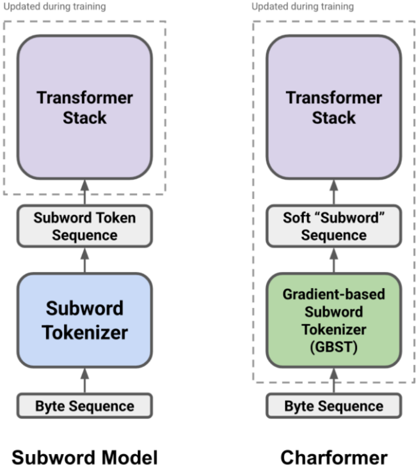
Image from: The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

T64 (2018) [1]



Entropy (bits)
4
3
2
1
0
edicted as the only conclusion that would satisfy elizabeth the authenticity of
Loss (bits)
9
8
7
6
5
4
3
2
1
0
edicted as the only conclusion that would satisfy elizabeth the authenticity of
Rank (logscale)
16
8
4
2
1
edicted as the only conclusion that would satisfy elizabeth the authenticity of

Charformer (2021) [2]

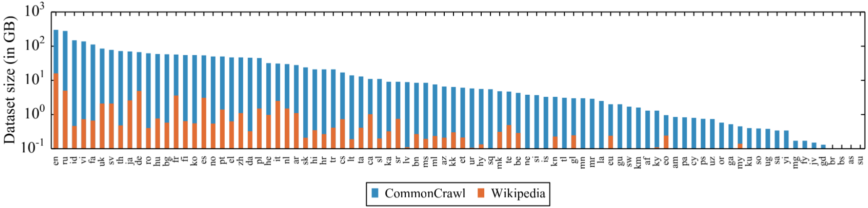


GPT-2 (2019) [3] - Byte BPE

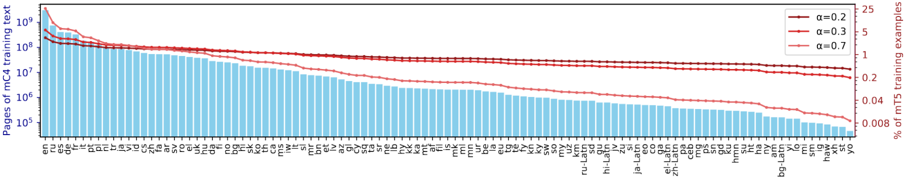
	BPE based on bytes	**BPE based on characters**
1	'I like cats.'	'I like cats.'
2	'I', ' like', 'cats', '.'	'I', 'like', 'cats', '.'
3	['0x49'], ['0x20', '0x6c', '0x69', '0x6b', '0x65'], ['0x20', '0x63', '0x61', '0x74', '0x73'], ['0x2e']	–
4	'I', 'Ġlike', 'Ġcats', '.'	–
5	'I', 'Ġli@@', 'ke', 'Ġca@@', 'ts', '.'	'I', 'li@@', 'ke' 'ca@@', 'ts', '.'

XLM-RoBERTa (2019) [4] / mT5 (2020) [5]

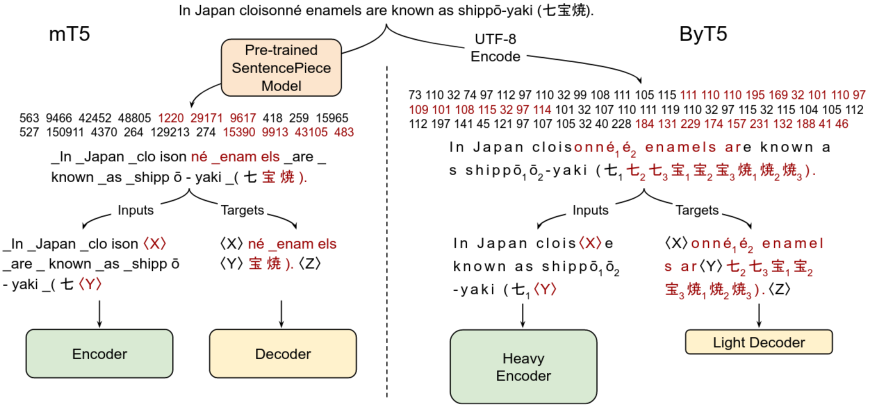
XLM-RoBERTa:



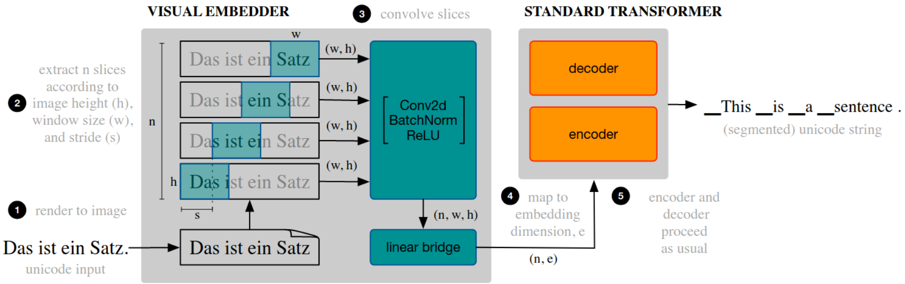
mT5:



ByT5 (2021) [6]


In Japan cloisonné enamels are known as shippō-yaki (七宝焼).
mT5
Pre-trained SentencePiece Model
563 9466 42452 48805 1220 29171 9617 418 259 15965
527 150911 4370 264 129213 274 15390 9913 43105 483
_In _Japan _clo ison né _enam els _are _
known _as _shipp ō - yaki _(七 宝 焼).
Inputs
Targets
_In _Japan _clo ison ⟨X⟩
_are _ known _as _shipp ō
- yaki _(七 ⟨Y⟩
⟨X⟩ né _enam els
⟨Y⟩ 宝 焼). ⟨Z⟩
Encoder
Decoder
UTF-8 Encode
ByT5
73 110 32 74 97 112 97 110 32 99 108 111 105 115 111 110 110 195 169 32 101 110 97
109 101 108 115 32 97 114 101 32 107 110 111 119 110 32 97 115 32 115 104 105 112
112 197 141 45 121 97 107 105 32 40 228 184 131 229 174 157 231 132 188 41 46
In Japan cloisonné$_1$é$_2$ enamels are known a
s shippō$_1$ō$_2$-yaki (七$_1$七$_2$七$_3$宝$_1$宝$_2$宝$_3$焼$_1$焼$_2$焼$_3$).
Inputs
Targets
In Japan clois⟨X⟩e
known as shippō$_1$ō$_2$
-yaki (七$_1$⟨Y⟩
⟨X⟩onné$_1$é$_2$ enamel
s ar⟨Y⟩七$_2$七$_3$宝$_1$宝$_2$
宝$_3$焼$_1$焼$_2$焼$_3$).⟨Z⟩
Heavy Encoder
Light Decoder

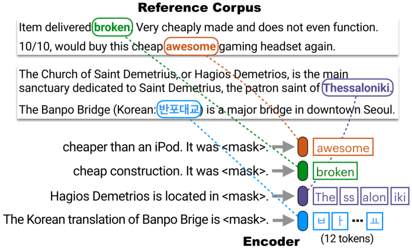
Visual Text Representation (2021) [7]



Non**P**arametric **M**asked Language Model (NPM) [8]

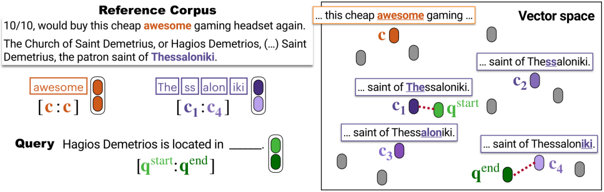
- December 2023, MetaAI - PyTorch (with code release: GitHub)
- predicts tokens based on a nonparametric distribution over phrases in a text corpus
- does not have a softmax over a fixed output vocabulary
- nonparametric distribution is defined by a function of the available data, not by a fixed set of parameters (LM-Head)
- predict extremely rare, unseen words and disambiguating word senses
- support effectively unlimited vocabulary sizes

Illustration of NPM



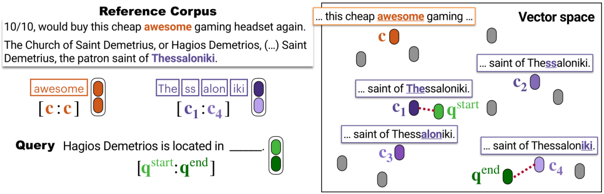
- NPM consists of an **encoder** and a **reference corpus**, and models a **nonparametric distribution** over a reference corpus
- key idea is to **map all the phrases in the corpus into a dense vector space** using the encoder
- at inference when given a query with a <MASK>, use the encoder to **locate the nearest phrase from the corpus** and **fill in the <MASK>**
- NPM can fill with **multiple tokens**

Mapping phrase into dense vector space



- encoder maps **every distinct phrase** in a reference corpus into a **dense vector space**
- standard indexing is expensive (indexing each token)
- represents a phrase with a **concatenation** of the token representation of the **start** and the **end** of the phrase
- phrase consisting of 4 BPE tokens c_1, c_2, c_3, c_4 is represented with a concatenation of vectors of c_1 and c_4

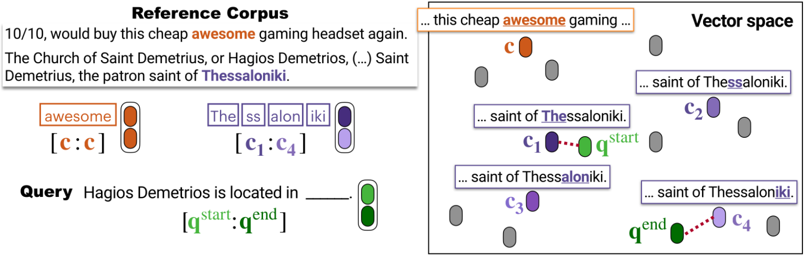
Retrieving phase



- replace <MASK> token to <MASK$_s$> and <MASK$_e$> tokens (representing the start and the end of the phase)
- replace each of token to vectors with the same vector space, respectively: q^{start} and q^{end} vectors
- use these vectors to **retrieve the start and the ending of the phrases**

$$q_1, ..., q_{t-1}, q^{start}, q^{end}, q_{t+2}, ..., q_L =$$
$$Encoder(t_1, ..., t_{t-1}, MASK_s, MASK_e, t_{t+2}, ..., t_L)$$

Approximation



- in practice, iterating over all phrases from corpus is infeasible
- approximation: using a fast nearest neighbor search for the start and the end separately – **take the top-k tokens with the highest similarity scores** with each of them, and **compute scores over spans composed by these tokens**
- use **scaled inner product** as similarity function

Training - issues

1. full corpus retrieval can make training very expensive
    - in-batch approximation to a full corpus – removing the need for keeping and updating the retrieval index during training
2. filling in a <MASK> with an arbitrary length phrase instead of a token is non-trivial
    - extensions to span masking and a contrastive objective which allow filling a <MASK> with a phrase

Masking

Sequence to mask

In the **2010** NFL season, **the Seattle Seahawks** made history by making it into the playoffs despite having a 7–9 record. (...) The Seahawks lost **to the** Bears in their second game, 35–24.

Other sequence in the batch

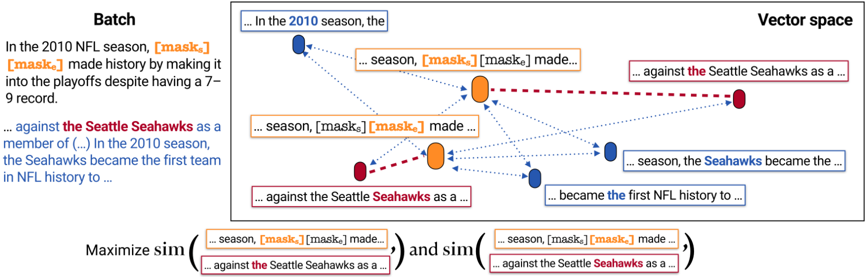
Russell Wilson's first game against **the Seattle Seahawks** (...) when they lost Super Bowl XLIX **to the** New England Patriots. In the **2010** season, the Seahawks became the first team in NFL history (..)

Masked sequence

In the **[mask$_s$] [mask$_e$]** NFL season, **[mask$_s$] [mask$_e$]** made history by making it into the playoffs despite having a 7–9 record. (...) The Seahawks lost **[mask$_s$] [mask$_e$]** Bears in their second game, 35–24.

1. mask spans if they co-occur in the other sequences in the batch
   - masked tokens: **2012** and **the Seattle Seahawks** and **to the**
   - **second game** will not used because **second** and **game** do not occur together in the other sequence in the batch
2. replace the whole span with two special tokens: <MASK$_s$> and <MASK$_e$>

Training Object – contrastive learning



- model should retrieve a phrase **the Seattle Seahawks** from other sequences in the reference corpus
- MASK$_s$ vector should be closer to **the Seattle Seahawks** (**positive sample**) while being distant from other tokens and should not be any **the** (from **became the first** – **negative sample**), similar to MASK$_e$ vector

Training Details

- corpus: English Wikipedia and English portion of CC-News – contains 13B tokens in total
- segmented into sequences, each with up to 256 tokens
- initialize from RoBERTa large
- training: 100 000 steps, 32 x 32GB GPUs
- one batch consists of 512 sequences GPUs

Scores

Model	# Params	AGN	Yahoo	Subj	SST-2	MR	RT	CR	Amz	RTE	Avg
Baselines (encoder-only)											
RoBERTa (Gao et al., 2021)	1.0x	-	-	51.4	83.6	80.8	-	79.5	-	51.3	-
RoBERTa	1.0x	71.3	41.4	67.6	84.5	81.7	81.1	80.4	83.5	57.4	72.1
Baselines (encoder-decoder)											
T5	2.2x	72.0	51.3	54.9	57.5	57.7	59.1	56.4	59.3	55.6	58.2
T5 3B	8.5x	**80.5**	53.6	54.8	59.6	58.6	57.3	53.7	57.0	58.5	59.3
Baselines (decoder-only)											
GPT-2 (Shi et al., 2022)	2.2x	67.4	49.7	60.8	55.3	54.6	53.0	66.2	57.6	53.1	57.5
+ PMI (Shi et al., 2022)	2.2x	65.1	48.8	62.5	76.5	74.6	74.1	82.8	76.2	54.2	68.3
GPT-2 kNN† (Shi et al., 2022)	2.2x	29.8	37.0	50.0	47.1	49.9	49.1	69.3	57.4	54.1	49.3
GPT-2 kNN-LM† (Shi et al., 2022)	2.2x	78.8	51.0	62.5	84.2	78.2	80.6	84.3	85.7	55.6	73.4
GPT-3 (Holtzman et al., 2021)	500x	75.4	53.1	66.4	63.6	57.4	57.0	53.8	59.4	56.0	60.2
+ PMI (Holtzman et al., 2021)	500x	74.7	54.7	64.0	71.4	76.3	75.5	70.0	75.0	**64.3**	69.5
Ours (encoder-only, nonparametric)											
NPM SINGLE†	1.0x	74.2	**54.8**	61.7	86.8	83.5	84.7	**84.9**	**88.5**	56.3	75.1
NPM†	1.0x	74.5	53.9	**75.5**	**87.2**	**83.7**	**86.0**	81.2	83.4	61.7	**76.4**
Full fine-tuning (reference)											
RoBERTa (Gao et al., 2021)	1.0x	-	-	97.0	95.0	90.8	-	89.4	-	80.9	-

Other topics

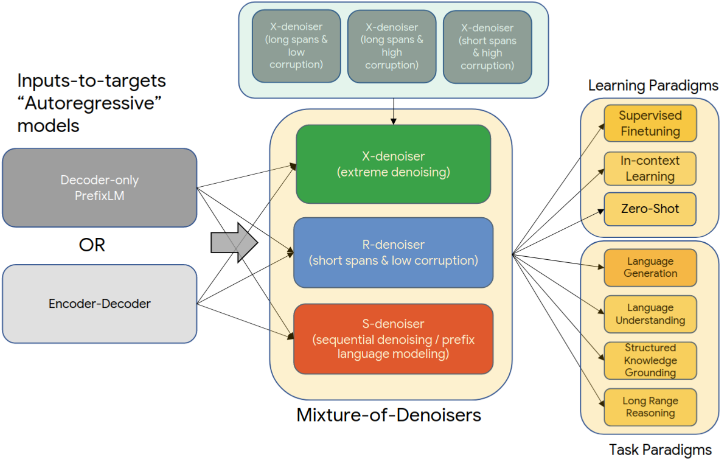
- **GLM-130B** [9] – bilingual (English and Chinese) pre-trained language model with 130 billion parameters
- **Lion** [10] – new optimization algorithm
- **ChatRWKV** [GitHub] – ChatRWKV [11] is like ChatGPT but powered by my RWKV (100% RNN) language model
- **Hyena** [12] – subquadratic drop-in replacement for attention constructed by interleaving implicitly parametrized long convolutions and data-controlled gating
- **SpikeGPT** [13] – generative language model with pure binary, event-driven spiking activation units, inspired by RWKV models
- **LLaMA** [14] – collection of foundation language models ranging from 7B to 65B parameters

Other topics

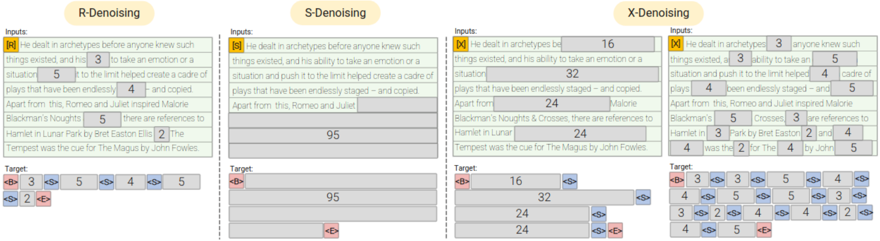
- **KOSMOS-1** [15] – Multimodal Large Language Model (MLLM) t trained on web-scale multi-modal corpora, including arbitrarily interleaved text and images, image-caption pairs, and text data
- **PaLM-E** [16] – fine-tune PaLM on multiple embodied tasks including sequential robotic manipulation planning, visual question answering, and captioning
- **ParaFormer** [17] – fast and accurate parallel transformer
- **Dropout** [18] – early dropout and late dropout
- **huggingface.js** [GitHub] – JS libraries to interact with the Hugging Face API, with TS types included
- **pandas 2.0** and the Arrow revolution [datapythonista blog]

Unifying Language Learning Paradigms (UL2)

- **UL2** (T5 UL2/FLAN-UL2) - release FLAN-UL2 20B [GitHub]

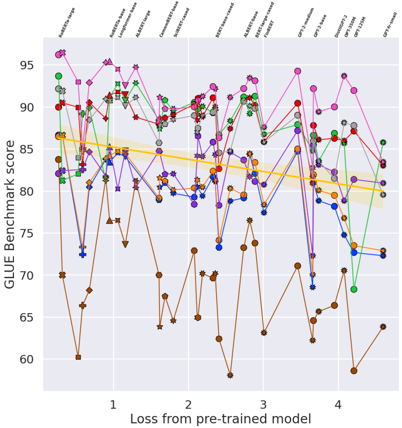


Unifying Language Learning Paradigms (UL2)



Some of my future

- checking if there is a correlation between **pre-trained model "loss"** and **downstream task**
- tested models base on Transformer architecture (encoder, decoder, encoder-decoder) – check ∼60 models
- not all models are available – they are not released
- not all models can be used – some weights missing or are too big
- training some models are not trivial – fast training is not trivial!
- make sure all experiments are easy do reproduce



References I

[1] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, “Character-level language modeling with deeper self-attention,” in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI’19/IAAI’19/EAAI’19, AAAI Press, 2019.

[2] Y. Tay, V. Q. Tran, S. Ruder, J. Gupta, H. W. Chung, D. Bahri, Z. Qin, S. Baumgartner, C. Yu, and D. Metzler, “Charformer: Fast character transformers via gradient-based subword tokenization,” in *International Conference on Learning Representations*, 2022.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” 2019.

[4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 8440–8451, Association for Computational Linguistics, July 2020.

[5] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 483–498, Association for Computational Linguistics, June 2021.

[6] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, “ByT5: Towards a token-free future with pre-trained byte-to-byte models,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[7] E. Salesky, D. Etter, and M. Post, “Robust open-vocabulary translation from visual text representations,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 7235–7252, Association for Computational Linguistics, Nov. 2021.

References II

[8] S. Min, W. Shi, M. Lewis, X. Chen, W.-t. Yih, H. Hajishirzi, and L. Zettlemoyer, “Nonparametric masked language modeling,” 2022.

[9] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma, Y. Xue, J. Zhai, W. Chen, P. Zhang, Y. Dong, and J. Tang, “Glm-130b: An open bilingual pre-trained model,” *arXiv preprint arXiv:2210.02414*, 2022.

[10] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le, “Symbolic discovery of optimization algorithms,” 2023.

[11] P. Bo, “Blinkdl/rwkv-lm: 0.01,” Aug. 2021.

[12] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. Baccus, Y. Bengio, S. Ermon, and C. Ré, “Hyena hierarchy: Towards larger convolutional language models,” 2023.

[13] R.-J. Zhu, Q. Zhao, and J. K. Eshraghian, “Spikegpt: Generative pre-trained language model with spiking neural networks,” 2023.

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[15] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, B. Patra, Q. Liu, K. Aggarwal, Z. Chi, J. Bjorck, V. Chaudhary, S. Som, X. Song, and F. Wei, “Language is not all you need: Aligning perception with language models,” 2023.

[16] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, “Palm-e: An embodied multimodal language model,” in *arXiv preprint arXiv:2303.03378*, 2023.

[17] X. Lu, Y. Yan, B. Kang, and S. Du, “Paraformer: Parallel attention transformer for efficient feature matching,” 2023.

References III

[18] Z. Liu, Z. Xu, J. Jin, Z. Shen, and T. Darrell, “Dropout reduces underfitting,” 2023.