Domain-Specific NLP

Contents

Domain-Specific Models

Domain-Specific Models

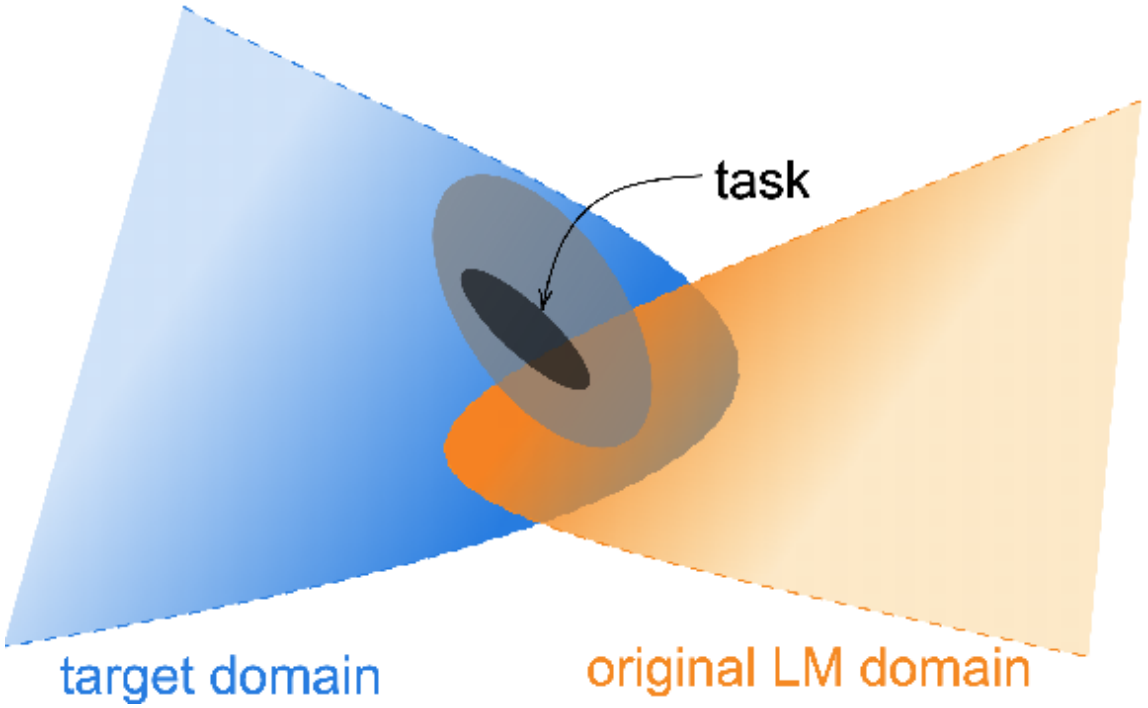
Pretrained (Large) Language Models are trained on content crawled over the internet, books, reports and news papers and are, hence **are open-domain**.

A **textual domain** is the **distribution over language characterizing a given topic or genre** [1].

- You are more likely to see the word "integer" in computer science than in news papers.
- An (L)LM will be more perplex to the word "integer" even though the input comes from a StackOverflow post.

Don’t Stop Pretraining

Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. [1]



Don’t Stop Pretraining

Domain	Pretraining Corpus	# Tokens	Size	$\mathcal{L}_{\text{RoB.}}$	$\mathcal{L}_{\text{DAPT}}$
BIOMED	2.68M full-text papers from S2ORC (Lo et al., 2020)	7.55B	47GB	1.32	0.99
CS	2.22M full-text papers from S2ORC (Lo et al., 2020)	8.10B	48GB	1.63	1.34
NEWS	11.90M articles from REALNEWS (Zellers et al., 2019)	6.66B	39GB	1.08	1.16
REVIEWS	24.75M AMAZON reviews (He and McAuley, 2016)	2.11B	11GB	2.10	1.93
ROBERTA (baseline)	see Appendix §A.1	N/A	160GB	‡1.19	-

Table 1: List of the domain-specific unlabeled datasets. In columns 5 and 6, we report ROBERTA’s masked LM loss on 50K randomly sampled held-out documents from each domain before $(\mathcal{L}_{ROB.})$ and after $(\mathcal{L}_{DAPT})$ $DAPT$ (lower implies a better fit on the sample). ‡ indicates that the masked LM loss is estimated on data sampled from sources similar to ROBERTA’s pretraining corpus.

Don’t Stop Pretraining

Figure 2: Vocabulary overlap (%) between domains. PT denotes a sample from sources similar to ROBERTA’s pretraining corpus. Vocabularies for each domain are created by considering the top 10K most frequent words (excluding stopwords) in documents sampled from each domain.

Don’t Stop Pretraining

Domain	Task	RoBERTa	Additional Pretraining Phases		
			DAPT	TAPT	DAPT + TAPT
BioMed	ChemProt	$81.9_{1.0}$	$84.2_{0.2}$	$82.6_{0.4}$	$\mathbf{84.4}_{0.4}$
	†RCT	$87.2_{0.1}$	$87.6_{0.1}$	$87.7_{0.1}$	$\mathbf{87.8}_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$75.4_{2.5}$	$67.4_{1.8}$	$\mathbf{75.6}_{3.8}$
	SciERC	$77.3_{1.9}$	$80.8_{1.5}$	$79.3_{1.5}$	$\mathbf{81.3}_{1.8}$
News	HyperPartisan	$86.6_{0.9}$	$88.2_{5.9}$	$\mathbf{90.4}_{5.2}$	$90.0_{6.6}$
	†AGNews	$93.9_{0.2}$	$93.9_{0.2}$	$94.5_{0.1}$	$\mathbf{94.6}_{0.1}$
Reviews	†Helpfulness	$65.1_{3.4}$	$66.5_{1.4}$	$68.5_{1.9}$	$\mathbf{68.7}_{1.8}$
	†IMDB	$95.0_{0.2}$	$95.4_{0.1}$	$95.5_{0.1}$	$\mathbf{95.6}_{0.1}$

Table 5: Results on different phases of adaptive pretraining compared to the baseline RoBERTa (col. 1). Our approaches are $DAPT$ (col. 2, §3), $TAPT$ (col. 3, §4), and a combination of both (col. 4).

Don’t Stop Pretraining

"We show that **pretraining the model towards a specific task or small corpus can provide significant benefits**. Our findings suggest it may be valuable to complement work on ever-larger LMs with parallel efforts to **identify and use domain and task relevant corpora to specialize models**."

BioBERT

"[..] the word distributions of general and biomedical corpora are quite different, which can often be a problem for biomedical text mining models." [2]

BioBERT

: 1. List of text corpora used for BioBERT

Corpus	# of words (B)	Domain
English Wikipedia	2.5B	General
BooksCorpus	0.8B	General
PubMed Abstracts	4.5B	Biomedical
PMC Full-text articles	13.5B	Biomedical

Table 1. List of text corpora used for BioBERT

BioBERT

"We showed that **pre-training BERT on biomedical corpora is crucial in applying it to the biomedical domain**. Requiring minimal task-specific architectural modification, **BioBERT outperforms previous models on biomedical text mining tasks** such as NER, RE and QA."

SciBERT

"[...] while both BERT and ELMo have released pretrained models, they are still trained on general domain corpora such as news articles and Wikipedia." [3]

SciBERT

Field	Task	Dataset	SOTA	BERT-Base		SCIBERT	
				Frozen	Finetune	Frozen	Finetune
Bio	NER	BC5CDR (Li et al., 2016)	88.85[7]	85.08	86.72	88.73	**90.01**
		JNLPBA (Collier and Kim, 2004)	**78.58**	74.05	76.09	75.77	77.28
		NCBI-disease (Dogan et al., 2014)	**89.36**	84.06	86.88	86.39	88.57
	PICO	EBM-NLP (Nye et al., 2018)	66.30	61.44	71.53	68.30	**72.28**
	DEP	GENIA (Kim et al., 2003) - LAS	**91.92**	90.22	90.33	90.36	90.43
		GENIA (Kim et al., 2003) - UAS	**92.84**	91.84	91.89	92.00	91.99
	REL	ChemProt (Kringelum et al., 2016)	76.68	68.21	79.14	75.03	**83.64**
CS	NER	SciERC (Luan et al., 2018)	64.20	63.58	65.24	65.77	**67.57**
	REL	SciERC (Luan et al., 2018)	n/a	72.74	78.71	75.25	**79.97**
	CLS	ACL-ARC (Jurgens et al., 2018)	67.9	62.04	63.91	60.74	**70.98**
Multi	CLS	Paper Field	n/a	63.64	65.37	64.38	**65.71**
		SciCite (Cohan et al., 2019)	84.0	84.31	84.85	**85.42**	**85.49**
Average				73.58	77.16	76.01	79.27

Table 1: Test performances of all BERT variants on all tasks and datasets. [...]

SciBERT

Task	Dataset	BioBERT	SciBERT
NER	BC5CDR	88.85	90.01
	JNLPBA	77.59	77.28
	NCBI-disease	89.36	88.57
REL	ChemProt	76.68	83.64

Table 2: Comparing SciBERT with the reported BioBERT results on biomedical datasets.

SciBERT

NB: SciBERT was trained on curated textual data ; not trained on code or script for example --at leat not trained directly and purposefully on this kind of data

Galactica

"Computing has indeed revolutionized how research is conducted, but information overload remains an overwhelming problem [...]. In this paper, we argue for a better way through large language models. Unlike search engines, language models can potentially store, combine and reason about scientific knowledge." [4]

- Galactica was trained on a rather small highly curated dataset.
- All the data was standardized as markdown text.

Galactica

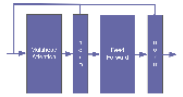
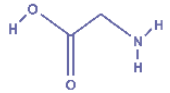
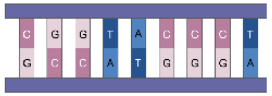
Modality	Entity	Sequence	
Text	Abell 370	`Abell 370 is a cluster...`	
LaTeX	Schwarzschild radius	`r_{s} = \frac{2GM}{c^2}`	$r_s = \frac{2GM}{c^2}$
Code	Transformer	`class Transformer(nn.Module)`	
SMILES	Glycine	`C(C(=O)O)N`	
AA Sequence	Collagen α-1(II) chain	`MIRLGAPQTL..`	
DNA Sequence	Human genome	`CGGTACCCTC..`	

Table 1: Tokenizing Nature. Galactica trains on text sequences that represent scientific phenomena.

Table 1: Tokenizing Nature. Galactica trains on text sequences that represent scientific phenomena.

Galactica

1. **Citations:** we wrap citations with special reference tokens [START_REF] and [END_REF].
2. **Step-by-Step Reasoning:** we wrap step-by-step reasoning with a working memory token , mimicking an internal working memory context.
3. **Mathematics:** for mathematical content, with or without LaTeX, we split ASCII operations into individual characters. Parentheses are treated like digits. The rest of the operations allow for unsplit repetitions. Operation characters are !"#$%&’*+,-./:;<=>?^_‘| and parentheses are ()[]{}.

4. **Numbers:** we split digits into individual tokens. For example 737612.62 -> 7,3,7,6,1,2,.,6,2.
5. **SMILES formula:** we wrap sequences with [START_SMILES] and [END_SMILES] and apply characterbased tokenization. Similarly we use [START_I_SMILES] and [END_I_SMILES] where isomeric SMILES is denoted. For example, C(C(=O)O)N → C,(,C,(,=,O,),O,),N.
6. **Amino acid sequences:** we wrap sequences with [START_AMINO] and [END_AMINO] and apply character-based tokenization, treating each amino acid character as a single token. For example, MIRLGAPQTL -> M,I,R,L,G,A,P,Q,T,L.

7. **DNA sequences:** we also apply a character-based tokenization, treating each nucleotide base as a token, where the start tokens are [START_DNA] and [END_DNA]. For example, CGGTACCCTC -> C, G, G, T, A, C, C, C, T, C.

Galactica

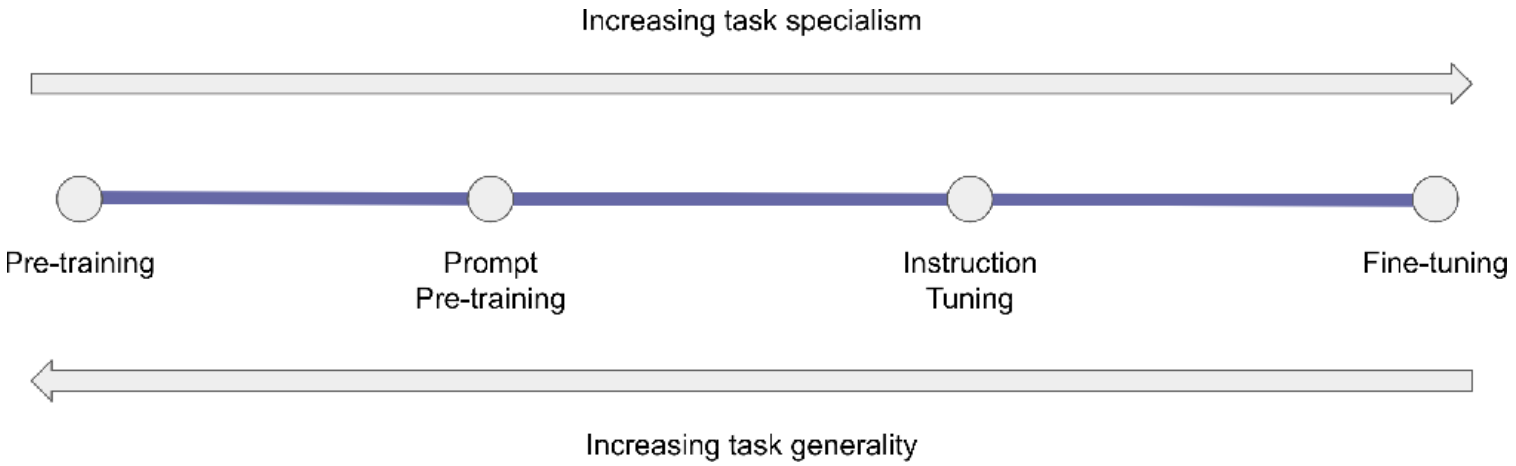


Figure 5: Prompt Pre-training. Pre-training weighs all tokens equally as part of the self-supervised loss. This leads to a weak relative signal for tasks of interest, meaning model scale has to be large to work. Instruction tuning boosts performance post hoc, and can generalize to unseen tasks of interest, but it risks performance in tasks that are distant from instruction set tasks. Prompt pre-training has a weaker task of interest bias than instruction tuning but less risk of

Galactica

- **GeLU Activation** - GeLU activations for all model sizes.
- **Context Window** - a 2048 length context window.
- **No Biases** - following PaLM, we do not use biases in any of the dense kernels or layer norms.
- **Learned Positional Embeddings** - learned positional embeddings for the model.
- **Vocabulary** - vocabulary of 50k tokens using BPE. The vocabulary was generated from a randomly selected 2% subset of the training data.

Galactica

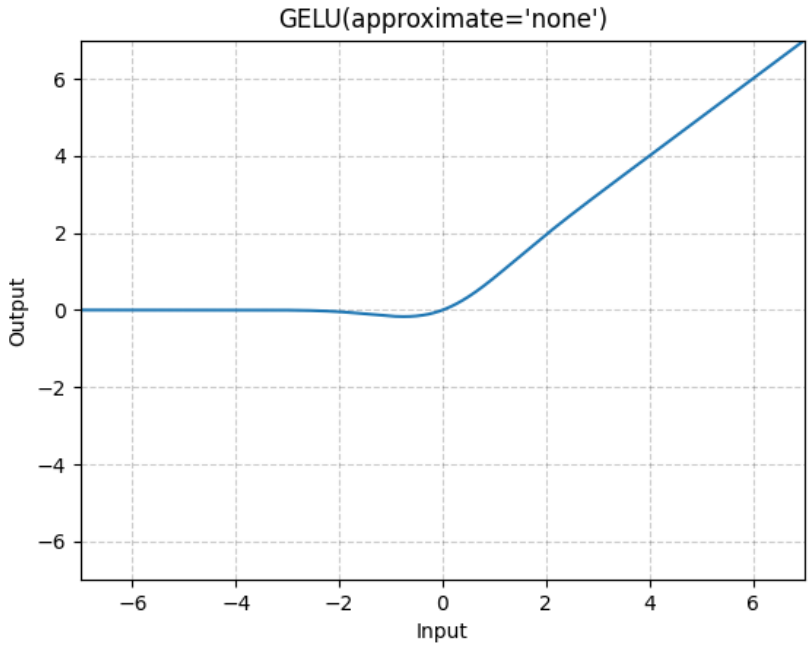
Gaussian Error Linear Units function (GeLu)

$$GELU(x) = x * \Phi(x)$$

Where $\Phi(x)$ is the Gaussian function.

$$GELU(x) \approx x * \frac{1}{2}(1 + Tanh(\frac{2}{\pi} * (x + 0.044715 * x^3)))$$

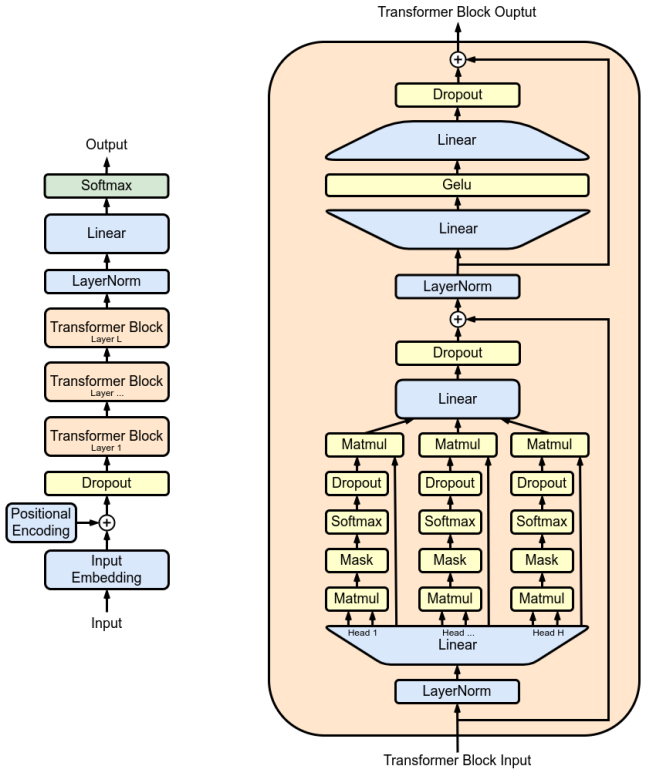
Galactica



- Allows small negative values when $x < 0$.
- Avoids the dying ReLU problem.

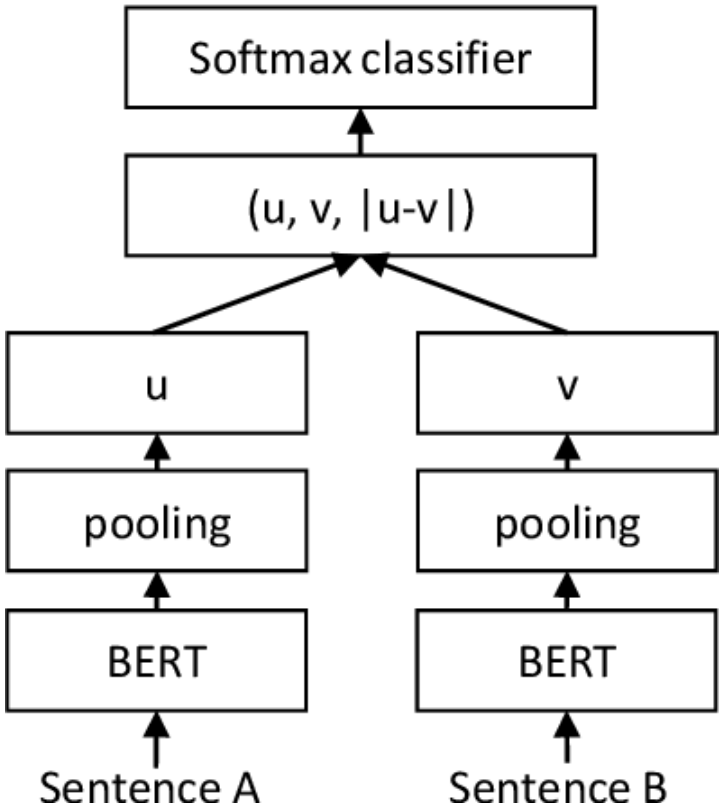
Galactica

Why no biases?



Unsupervised Classification Models

Document Representation



[6]

Document Representation

[5]

Document Representation

	NLI	STSb
Pooling Strategy		
MEAN	**80.78**	**87.44**
MAX	79.07	69.92
CLS	79.80	86.62
Concatenation		
(u, v)	66.04	-
$(\|u - v\|)$	69.78	-
$(u * v)$	70.54	-
$(\|u - v\|, u * v)$	78.37	-
$(u, v, u * v)$	77.44	-
$(u, v, \|u - v\|)$	**80.78**	-
$(u, v, \|u - v\|, u * v)$	80.44	-

[6]

Document Representation

(1) The data is being compressed mutliple time -> challeging document can be hard to embed.
(2) Slow to process, as we need to chunk the inputs to make multiple inferences.

Can we do better?

SimCSE

Contrastive learning uses similar data point and opposite ones in order for the model build close representations for the first ones and and more separated ones for the latter. [7]

- Unsupervised SimCSE: standard dropout as data augmentation
- Supervised SimCSE: use pairs in NLI datasets

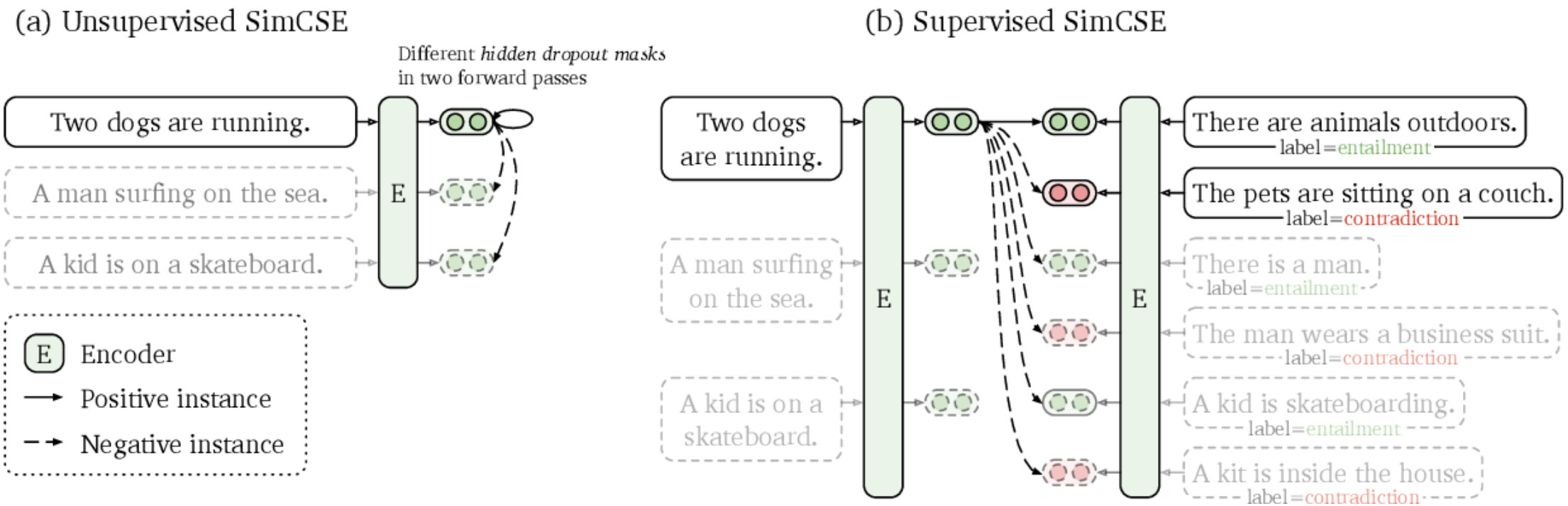
SimCSE

$$\mathcal{L}_{uns} = -log\frac{exp(\frac{sim(\mathbf{h}_i, \mathbf{h}_i^+)}{\tau})}{\sum_{j=1}^{N} exp(\frac{sim(\mathbf{h}_i, \mathbf{h}_j^+)}{\tau})}$$

$$\mathcal{L}_{sup} = -log\frac{exp(\frac{sim(\mathbf{h}_i, \mathbf{h}_i^+)}{\tau})}{\sum_{j=1}^{N} exp(\frac{sim(\mathbf{h}_i, \mathbf{h}_j^+)}{\tau}) + exp(\frac{sim(\mathbf{h}_i, \mathbf{h}_j^-)}{\tau})}$$

[8]

SimCSE


(a) Unsupervised SimCSE
Different hidden dropout masks in two forward passes
Two dogs are running.
A man surfing on the sea.
A kid is on a skateboard.
E
E Encoder
Positive instance
Negative instance
(b) Supervised SimCSE
Two dogs are running.
A man surfing on the sea.
A kid is on a skateboard.
E
E
There are animals outdoors.
label=entailment
The pets are sitting on a couch.
label=contradiction
There is a man.
label=entailment
The man wears a business suit.
label=contradiction
A kid is skateboarding.
label=entailment
A kit is inside the house.
label=contradiction

SimCSE

- The pretrained embeddings are being regularized to be more uniform.
- Semantically close pairs are better aligned.

Better performances, hence solving (1).

See also [9] [10]

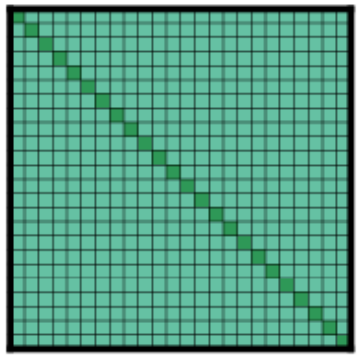
SimCSE

	Params	Class.	Clust.	Pair.	Rerank	Retr.	STS	Summ.	Avg
# of datasets →		12	11	3	4	15	10	1	56
Unsupervised models									
Glove	120M	57.3	27.7	70.9	43.3	21.6	61.9	28.9	42.0
BERT	110M	61.7	30.1	56.3	43.4	10.6	54.4	29.8	38.3
SimCSE	110M	62.5	29.0	70.3	46.5	20.3	74.3	31.2	45.5
$E5_{small}$	30M	67.0	41.7	78.2	53.1	40.8	68.8	25.2	54.2
$E5_{base}$	110M	67.9	43.4	79.2	53.5	42.9	69.5	24.3	55.5
$E5_{large}$	330M	69.0	44.3	80.3	54.4	44.2	69.9	24.8	56.4
GTE_{small}	30M	71.0	44.9	82.4	57.5	43.4	77.2	30.4	58.5
GTE_{base}	110M	71.5	46.0	83.3	58.4	44.2	76.5	29.5	59.0
GTE_{large}	330M	71.8	46.4	83.3	58.8	44.6	76.3	30.1	59.3
Supervised models									
SimCSE	110M	67.3	33.4	73.7	47.5	21.8	79.1	23.3	48.7
Contriever	110M	66.7	41.1	82.5	53.1	41.9	76.5	30.4	56.0
GTR_{large}	330M	67.1	41.6	85.3	55.4	47.4	78.2	29.5	58.3
Sentence-$T5_{large}$	330M	72.3	41.7	85.0	54.0	36.7	81.8	29.6	57.1
$E5_{small}$	30M	71.7	39.5	85.1	54.5	46.0	80.9	31.4	58.9
$E5_{base}$	110M	72.6	42.1	85.1	55.7	48.7	81.0	31.0	60.4
$E5_{large}$	330M	73.1	43.3	85.9	56.5	50.0	82.1	31.0	61.4
$InstructOR_{base}$	110M	72.6	42.1	85.1	55.7	48.8	81.0	31.0	60.4
$InstructOR_{large}$	330M	73.9	45.3	85.9	57.5	47.6	83.2	31.8	61.6
$OpenAI_{ada-001}$	n.a.	70.4	37.5	76.9	49.0	18.4	78.6	26.9	49.5
$OpenAI_{ada-002}$	n.a.	70.9	45.9	84.9	56.3	49.3	81.0	30.8	61.0
GTE_{small}	30M	72.3	44.9	83.5	57.7	49.5	82.1	30.4	61.4
GTE_{base}	110M	73.0	46.1	84.3	58.6	51.2	82.3	30.7	62.4
GTE_{large}	330M	73.3	46.8	85.0	59.1	52.2	83.4	31.7	63.1
Larger models									
$InstructOR_{xl}$	1.5B	73.1	44.7	86.6	57.3	49.3	83.1	32.3	61.8
GTR_{xxl}	4.5B	67.4	42.4	86.1	56.7	48.5	78.4	30.6	59.0
Sentence-$T5_{xxl}$	4.5B	73.4	43.7	85.1	56.4	42.2	82.6	30.1	59.5

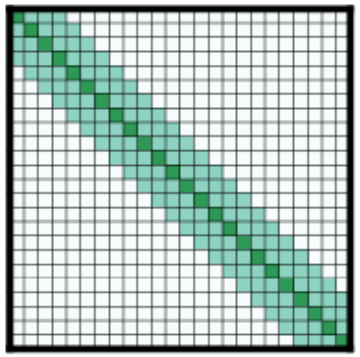
Learning Long-Range Dependencies

Long-range attention models

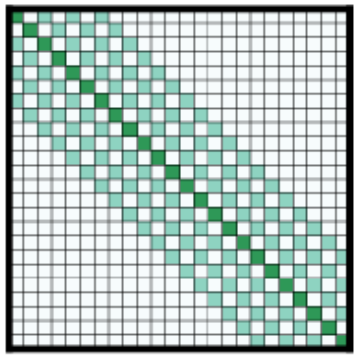
Sliding window attention: Longformer [11]



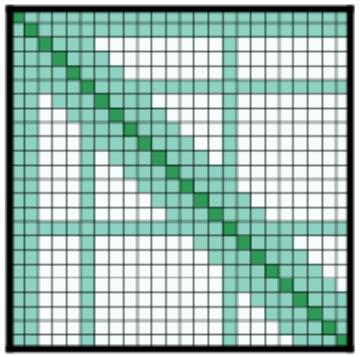
(a) Full n^2 attention



(b) Sliding window attention



(c) Dilated sliding window



(d) Global+sliding window

Long-range attention models

Sliding window attention: Mistral 7B [12]

2 Architectural details

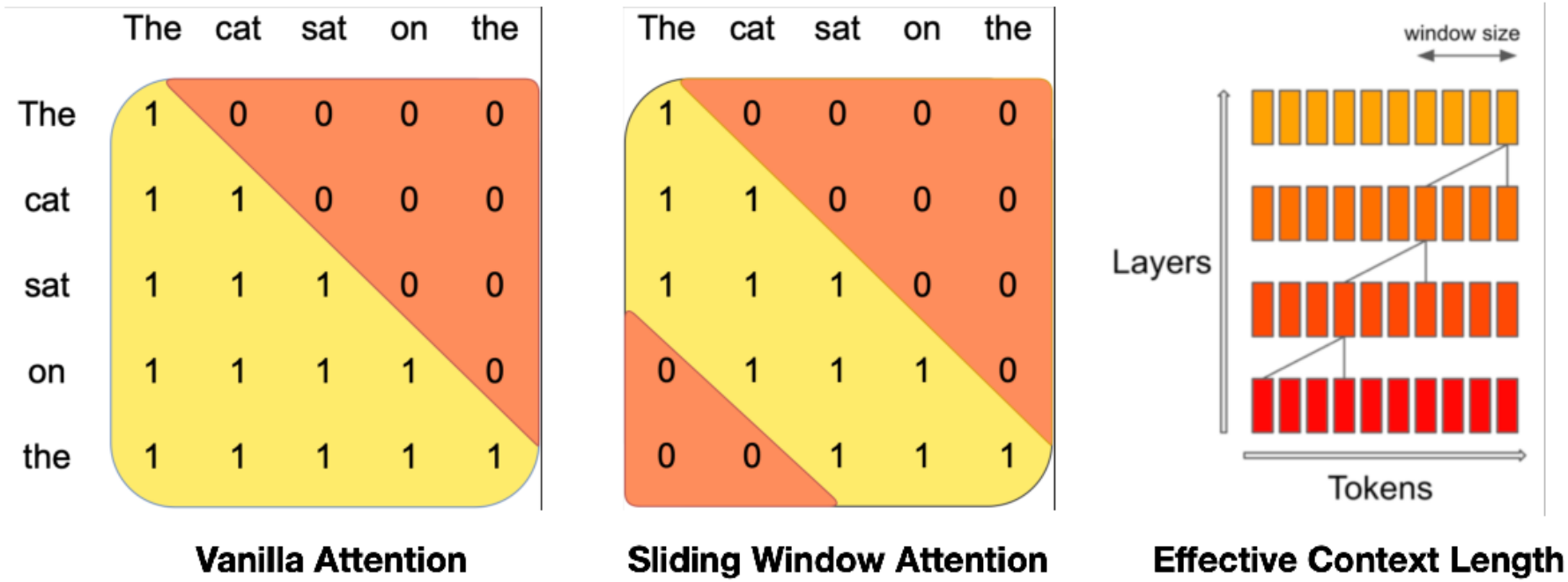


Figure 1: Sliding Window Attention. The number of operations in vanilla attention is quadratic in the sequence

State-space models: Mamba

The loss can be the likes of cross-entropy (CE), binary cross-entropy (BCE) or KL-Divergence (KL).

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n'=1}^{N} y^{(n)}.log(f(\mathbf{x},\theta)^{(n)})$$

$$\mathcal{L}_{BCE} = -y^{(n)}.log(f(\mathbf{x},\theta)^{(n)}) + (1-y^{(n)}).(1-f(\mathbf{x},\theta)^{(n)})$$

$$\mathcal{L}_{KL} = -\frac{1}{N}\sum_{n'=1}^{N} y^{(n)}.log(\frac{y^{(n)}}{f(\mathbf{x},\theta)^{(n)}})$$

Questions?

References

[1] Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks." In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 8342–60. Online: Association for Computational Linguistics, 2020.

[2] Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining." Bioinformatics 36, no. 4 (February 15, 2020): 1234–40.

[3] Beltagy, Iz, Kyle Lo, and Arman Cohan. "SciBERT: A Pretrained Language Model for Scientific Text." In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3615–20. Hong Kong, China: Association for Computational Linguistics, 2019.

[4] Taylor, Ross, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. “Galactica: A Large Language Model for Science.” arXiv, November 16, 2022.

[5] Nurmambetova, Elvira, et al. "Developing an Inpatient Electronic Medical Record Phenotype for Hospital-Acquired Pressure Injuries: Case Study Using Natural Language Processing Models." JMIR AI 2.1 (2023): e41264.

[6] Reimers, Nils, and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks.” In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3982–92. Hong Kong, China: Association for Computational Linguistics, 2019.

[7] Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings." In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 6894–6910. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021.

[8] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding." arXiv, January 22, 2019.

[9] Wang, Liang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. "Text Embeddings by Weakly-Supervised Contrastive Pre-Training." arXiv, December 7, 2022.

[10] Li, Zehan, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. "Towards General Text Embeddings with Multi-Stage Contrastive Learning." arXiv, August 6, 2023.

[11] Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer." arXiv, December 2, 2020.

[12] Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, et al. “Mistral 7B.” arXiv, October 10, 2023.