ProteinBERT = A self-supervised model for proteins

ProteinBERT & the language of proteins:

Deep-learning protein language models for sequence and function

Dan Ofer

Overview

1. Proteins & Text & NLP (Oh my!)
2. (Deep) Language Models (BERT) & transfer-learning
3. ProteinBert

ELSEVIER

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

The language of proteins: NLP, machine learning & protein sequences

Dan Ofer [a], Nadav Brandes [b,*], Michal Linial [c]

[a] *Medtronic, Inc, Israel*
[b] *The Rachel and Selim Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel*
[c] *Department of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel*

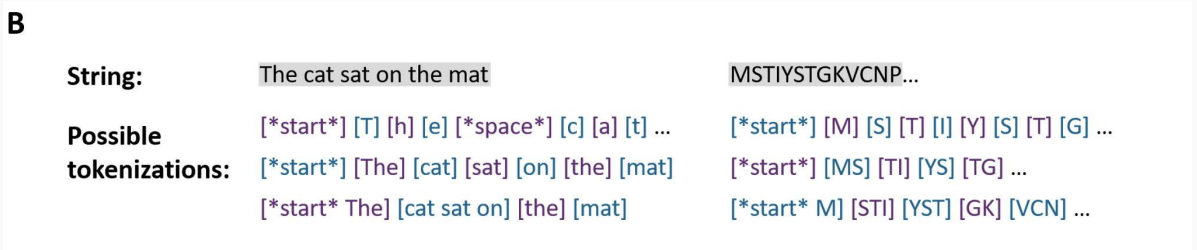
https://www.sciencedirect.com/science/article/pii/S2001037021000945

The language of proteins: NLP, machine learning & protein sequences. https://doi.org/10.1016/j.csbj.2021.03.022

Texts and proteins

Tokenization: A token = “fundamental unit of information”

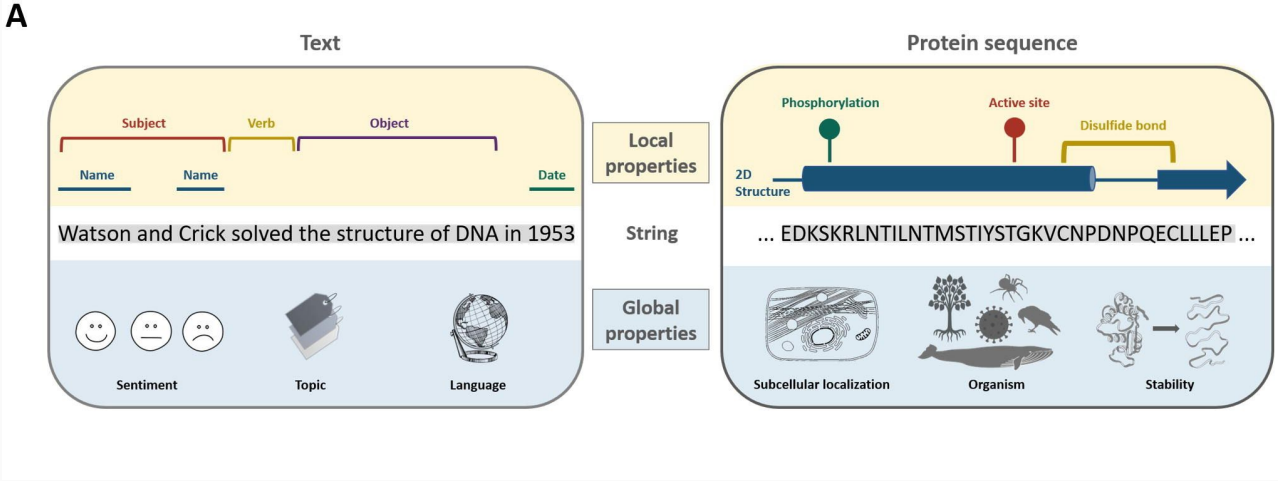
- Human Texts have phrases, words, subwords & letters
- Proteins have ~22 amino acids (“letters”) & (unknown) structures



Source: The language of proteins: NLP, machine learning & protein sequences. D Ofer, N Brandes, M Linial. https://doi.org/10.1016/j.csbj.2021.03.022

Texts and proteins

Texts and Proteins can be represented as strings of letters and processed with NLP (Natural Language Processing) methods

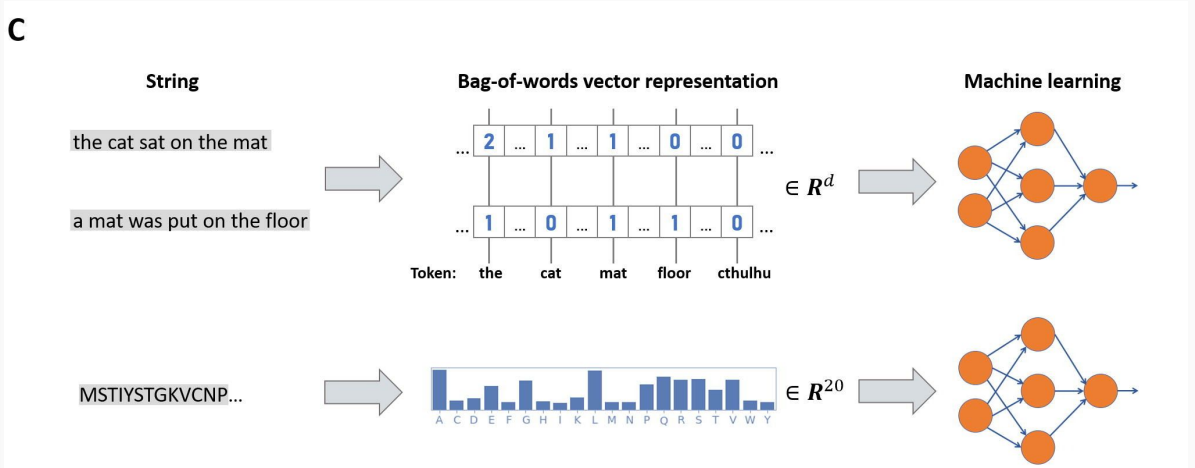


Source: The language of proteins: NLP, machine learning & protein sequences. D Ofer, N Brandes, M Linial. https://doi.org/10.1016/j.csbj.2021.03.022

NLP: Bag-of-words

The “Classic” approach: Count occurrence of tokens in text.

- Ignores order (unless using n-grams)
- Simple & effective
- Variants: normalized frequency, Term Frequency–Inverse Document Frequency (TF-IDF), count word combinations (n-grams/k-mers)
- For more variants: ProFET, Feature engineering 101



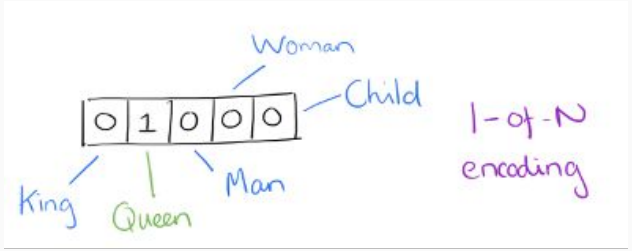
Source: The language of proteins: NLP, machine learning & protein sequences. D Ofer, N Brandes, M Linial. https://doi.org/10.1016/j.csbj.2021.03.022


WE NEED TO GO
DEEPER
memegenerator.net

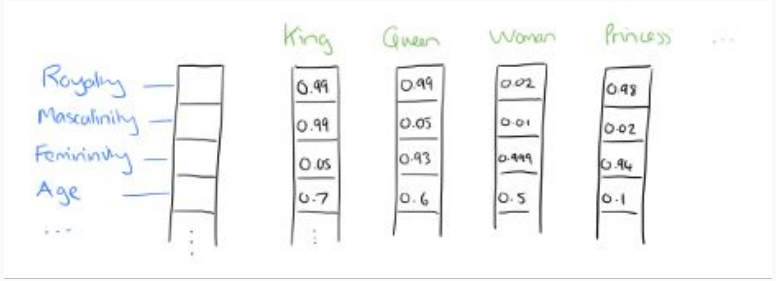
Word Embeddings & Word2Vec

- Bag of words is very high-dimensional, sparse, doesn't capture similarities.
- **Idea**: represent words as **dense vectors in embedding space**

~~[0 1 0 0 0 0 0 0 0 0]~~



[0.315 0.136 0.831]

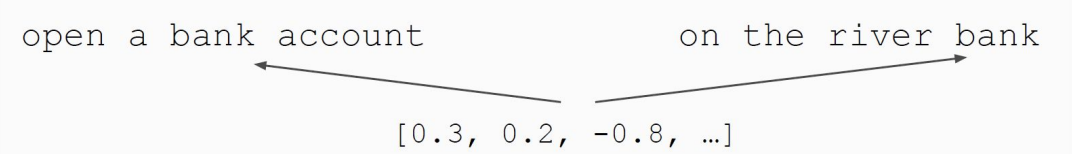


Proteins & Word2Vec

- Word embedding on biological sequences: Proteins/Amino acids, peptide k-mers, DNA, RNA etc’
  - Amino acid embeddings similarity matrix resembles classical substitution matrix (BLOSUM)
  - Multiple works

Word2Vec vs Contextualized embeddings

Problem: Word embeddings are applied in a context free manner



Contextual representations take the given context into account.

- Derived with deep learning models

Proteins != "Natural" Language

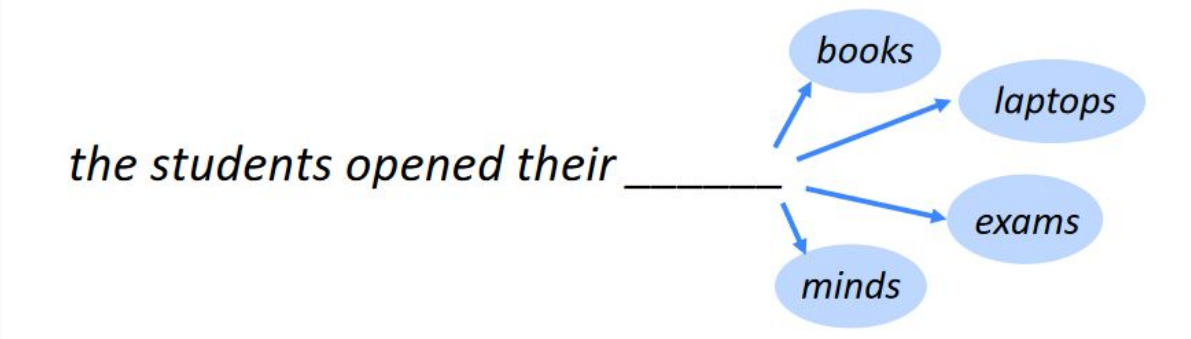
Biological sequences (Proteins, DNA, RNA) are NOT just another language - they are different from “natural”, human (or animal) languages!

- We can’t speak/read/write them
- No words, clear units of meaning, or white-space/seperators (“.”)
- Proteins vary far more in length vs human sentences
- Physical constructs (proteins = long string machine) vs semantics

...

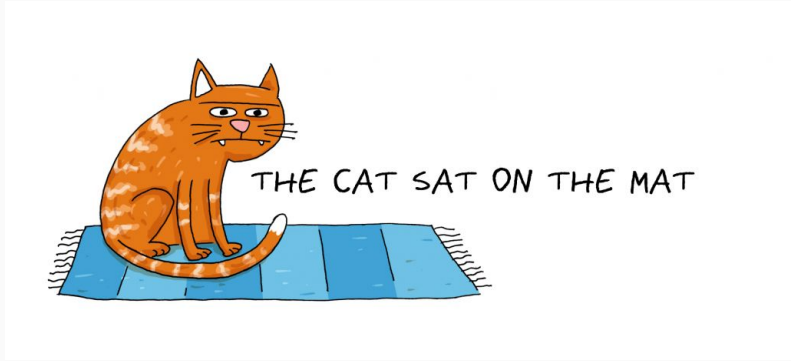
Language models

Language models: Predict the missing word



Language models: Predict the most *likely* word

"The ___ sat on the mat"



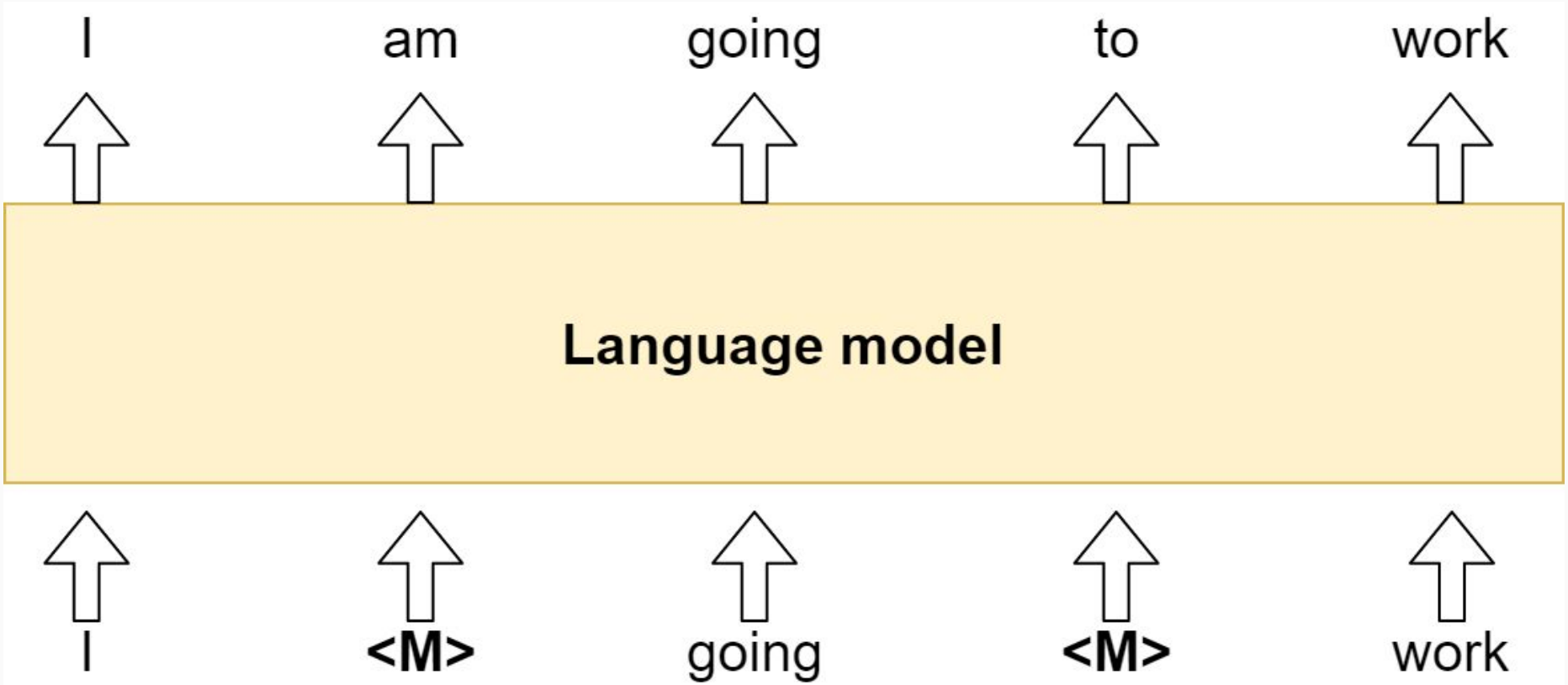
Cthulhu

running

Deep Language models

= Language models with Neural networks

- Transformers (Attention) (BERT) = State of the art
- RNNS/LSTMs also very popular and effective (ULMFiT, ELMO etc')
- Contextualized embeddings vs Word2Vec
- BERT & derived works use bidirectional masked language modelling:



Bidirectional Masked Language modelling

Language Models are Unsupervised Multitask Learners

Predicting likely words requires understanding of many features (context, vocabulary, syntax, common-sense etc”) !

Model must implicitly learn a lot in order to do this.

The task can be done **unsupervised** (no annotations needed), **quickly**, on **massive datasets**

AKA: Self-Supervised learning

But How does it help us?

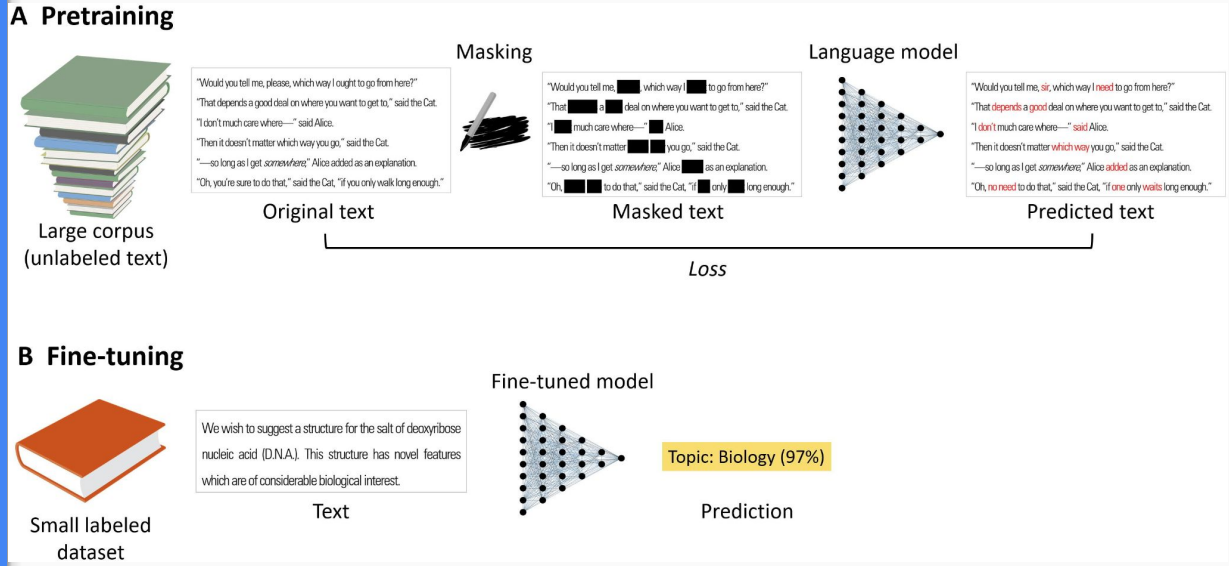
Transfer Learning

Pretrain on large “unsupervised” data and use trained model for “downstream” task of interest

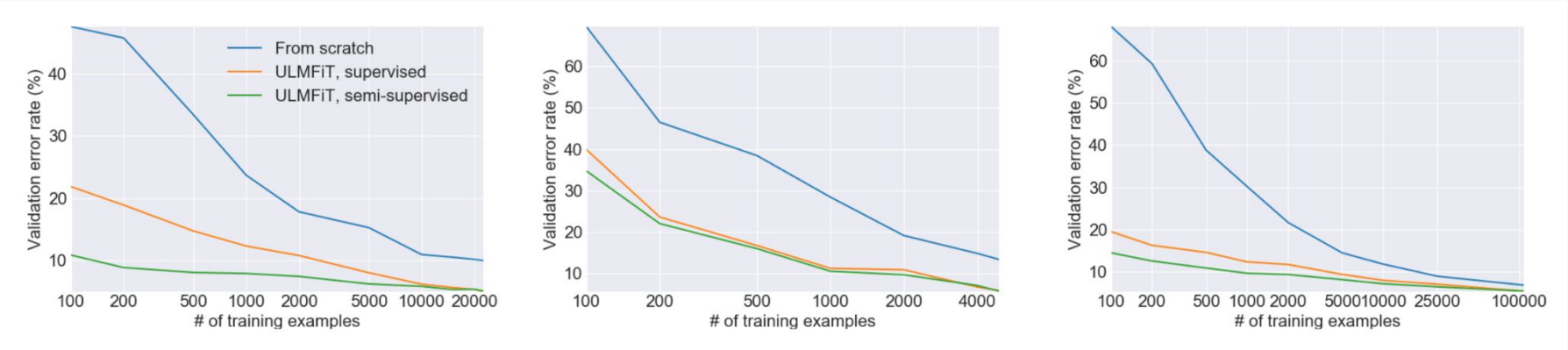
2 Approaches:

- Feature extractor: get contextualized embeddings and hidden layer outputs
- Fine-tune: retrain model with different output layer(s)

“Imagenet Moment”



Self-Supervised learning (Pretraining) with deep language models works!



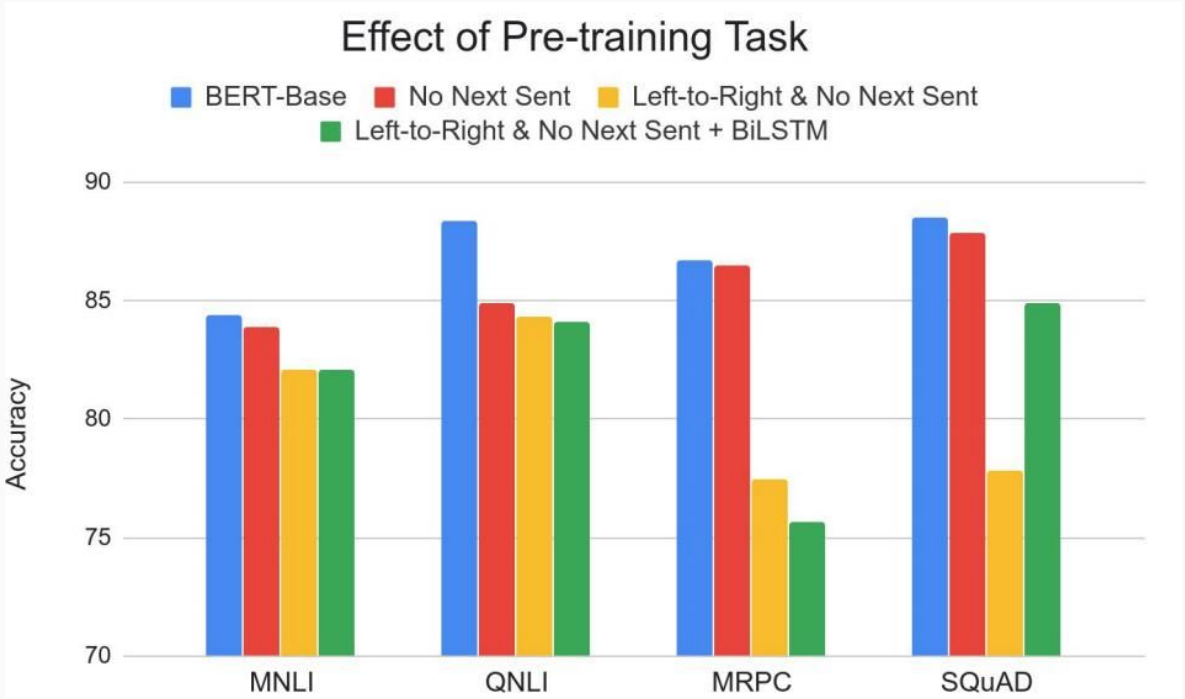
Validation error rates for ULMFiT model vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (NLP datasets)

Figure source: [ULMFiT: Universal Language Model Fine-tuning for Text Classification]
J Howard, S Ruder. Proceedings of ACL 2018

Pretraining deep language models works!

BERT: Masked vs “Causal” language modelling benefits:

- GLUE (NLP benchmarks)
- Masked LM (compared to left-to-right LM) is very important on some tasks
- Another task, Next Sentence Prediction was important on other tasks.

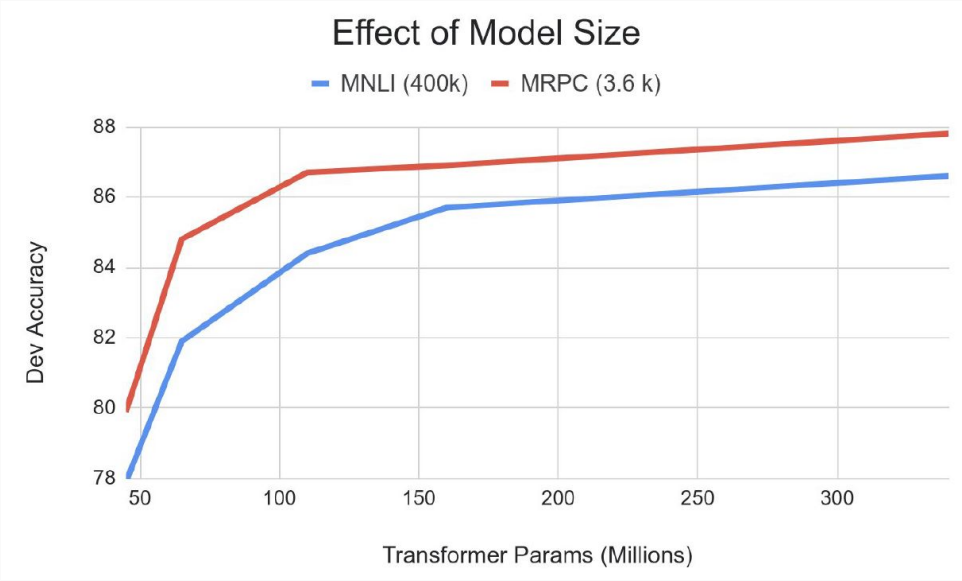


Source: https://nlp.stanford.edu/seminar/details/jdevlin.pdf

“With pre-training, bigger == better, without clear limits”

- Jacob Devlin (BERT)

Bigger = Better Models



- Big models help a lot
- Going from 110M -> 340M params helps even on datasets with 3,600 labeled examples
- (Pretraining) Improvements have not asymptoted

Source: https://nlp.stanford.edu/seminar/details/jdevlin.pdf

“My money is on self-supervised learning”

- Yann LeCunn

ProteinBERT:
A universal deep-learning model of protein sequence and function

Dan Ofer, Nadav Brandes, Yam Peleg, Nadav Rappoport, Michal Linial

Protein Language models

Recent development in the field (of Biology), but now rapidly growing!

Same general idea as in NLP:

1. Self-supervised training of a large, deep learning model over a dataset of sequences
2. Retrain or use as feature extractor for transfer learning
3. Profit???

Recipe for a Protein language model

1. Take an existing architecture (BERT) from HuggingFace-Transformers
2. Train on protein sequences with default pretraining tasks (Masked Language Modelling & next-sentence prediction)
    - Uniprot/Uniref 50/90/100, PFAM, UniParc, metagenomics, etc’
3. Fine-tune on benchmarks
    - Bonus points for Tasks Assessing Protein Embeddings (TAPE)
    - Secondary structure, remote homology (Family/Fold/class prediction), stability, etc’

(Some) Protein Language model papers

- UDSMProt
- Evaluating Protein Transfer Learning with TAPE
- Evolutionary Scale Modeling (ESM)- Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences
- UniRep (Unified rational protein engineering with sequence-based deep representation learning)
- ProGen: Language Modeling for Protein Generation
- ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Learning
- MSA Transformer
- BERTology Meets Biology: Interpreting Attention in Protein Language Models
- **ProteinBERT: A universal deep-learning model of protein sequence and function**
    - **Overview** of the field: **The language of proteins: NLP, machine learning & protein sequences**
- **...**

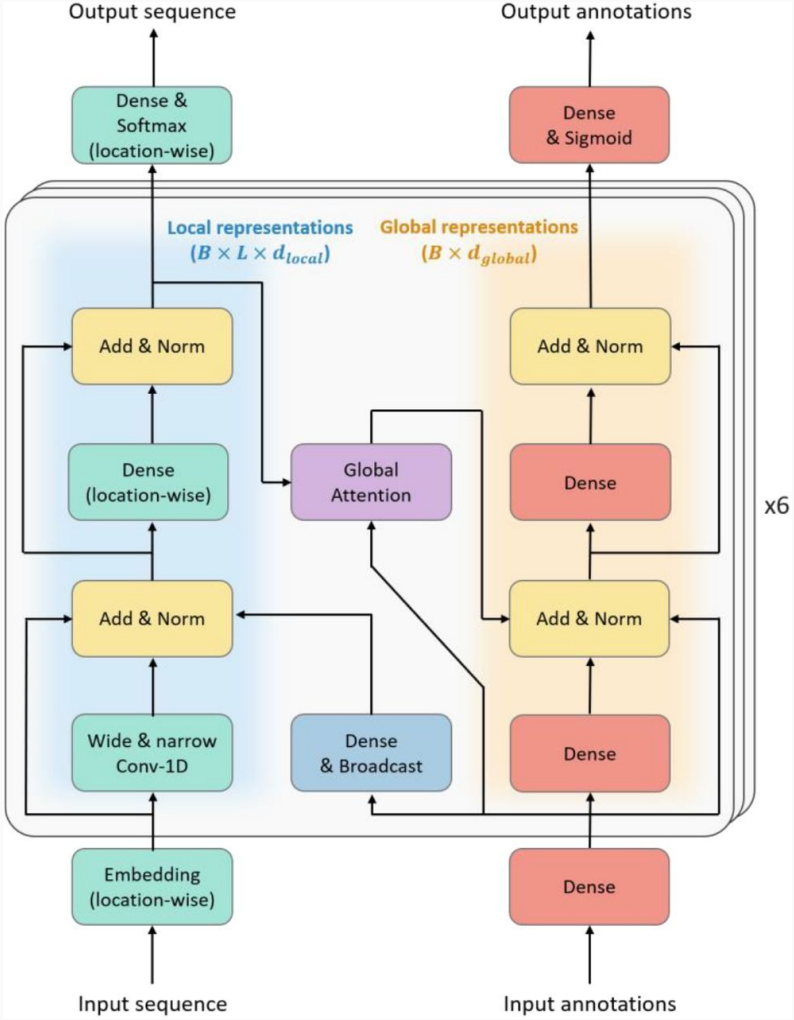
ProteinBERT at a glance

- GO (Gene Ontology) annotations pretraining task & inputs
- Denoising based pretraining
- Different model architecture
  - Faster, smaller
  - Global (Linear) attention
  - Flexible sequence lengths
  - Based on transformer/attention + Convolutions
- Strong results on benchmarks despite size

ProteinBERT Architecture

Inspired by BERT/Transformers, but with many changes!

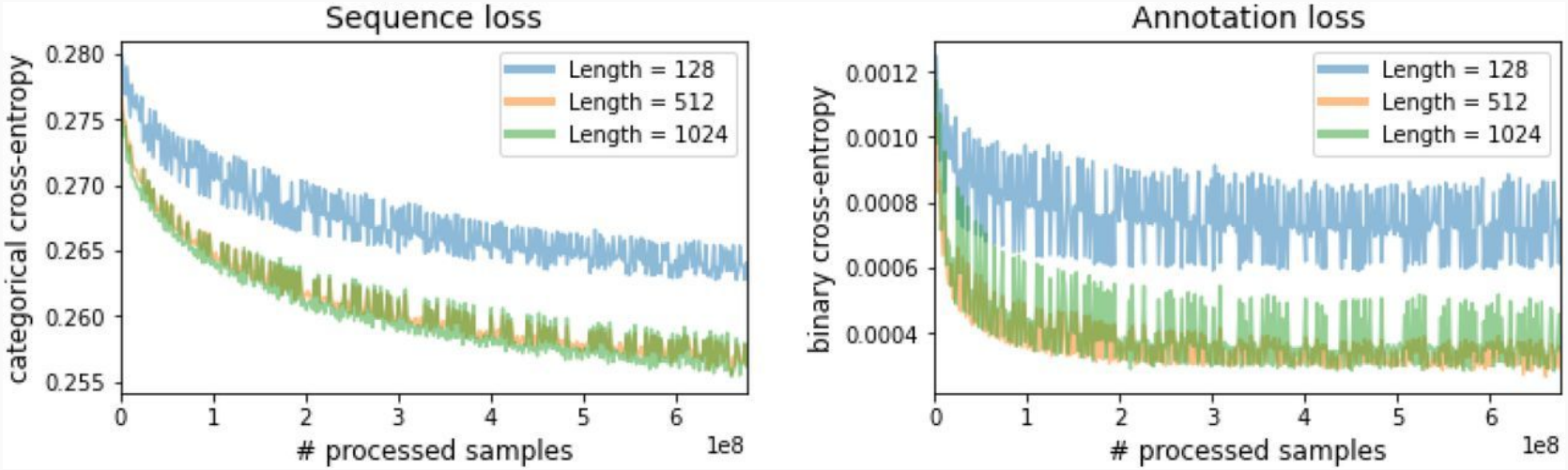
- GO Annotations task
- (Linear) Global attention
- No sequence self-attention
- No positional embeddings
- No dropout/regularization
- 1D convolutions
- Denoising instead of MLM
- Global input features (GO)
- Sequence-length agnostic
- Far smaller and faster (16M vs 38M for TAPE-Transformer, 110M BERT's)



ProteinBERT Pretraining

- Dataset: **~106,000,000** UniRef90 proteins & **8,943** Gene Ontology annotations
  - Sequences similar to benchmarks' test sets removed
- Gene Ontology (GO): Cellular, biological, molecular functions
  - ~43M proteins had any annotation, 2.3 annotations per protein
  - Multilabel task, with unknown negatives, noisy labels
- Trained for 1 month on 1 GPU: ~6.4 epochs, 670M sequences
  - Notably faster throughput than comparable works

Pretraining tasks loss



Training loss over the two pretraining tasks:

1. Sequence loss: protein sequence language modeling denoising (no mask tokens)
2. GO annotation recovery

TAPE benchmarks

TAPE (Tasks Assessing Protein Embeddings) Benchmarks

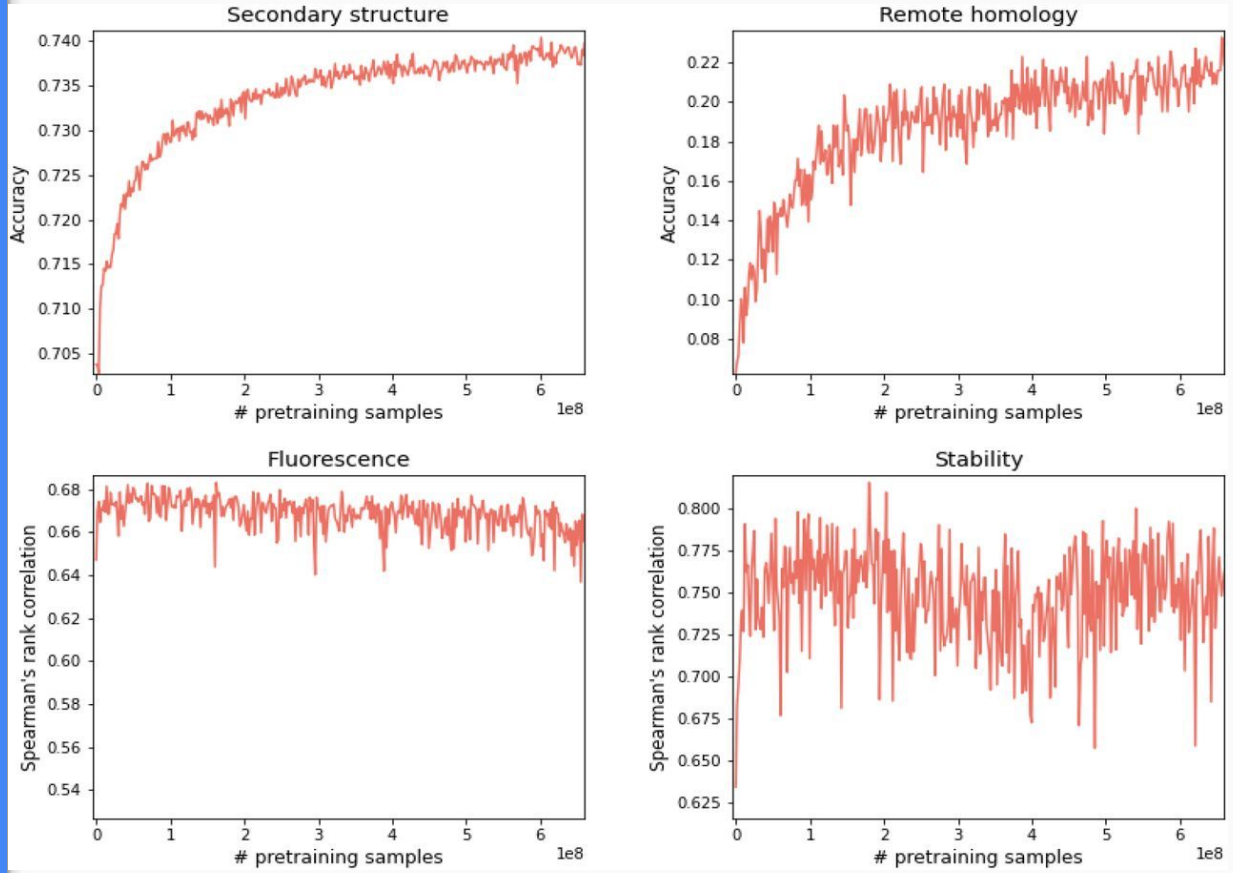
- Compared to other models, pretrained using language modelling on ~30M PFAM proteins
- ProteinBert does better than standard transformer (TAPE-Transformer), despite smaller size
- Pretraining helps

	Method	Structure	Evolutionary	Engineering	
		Secondary structure	Remote homology	Fluorescence	Stability
Without Pretraining	TAPE Transformer	0.70	0.09	0.22	-0.06
	LSTM	0.71	0.12	0.21	0.28
	ProteinBERT	0.70	0.06	0.65	0.63
With Pretraining	TAPE Transformer	0.73	0.21	0.68	0.73
	LSTM	0.75	0.26	0.67	0.69
	UniRep mLSTM	0.73	0.23	0.67	0.73
	ProteinBERT	0.74	0.22	0.66	0.76

Pretraining improves downstream tasks

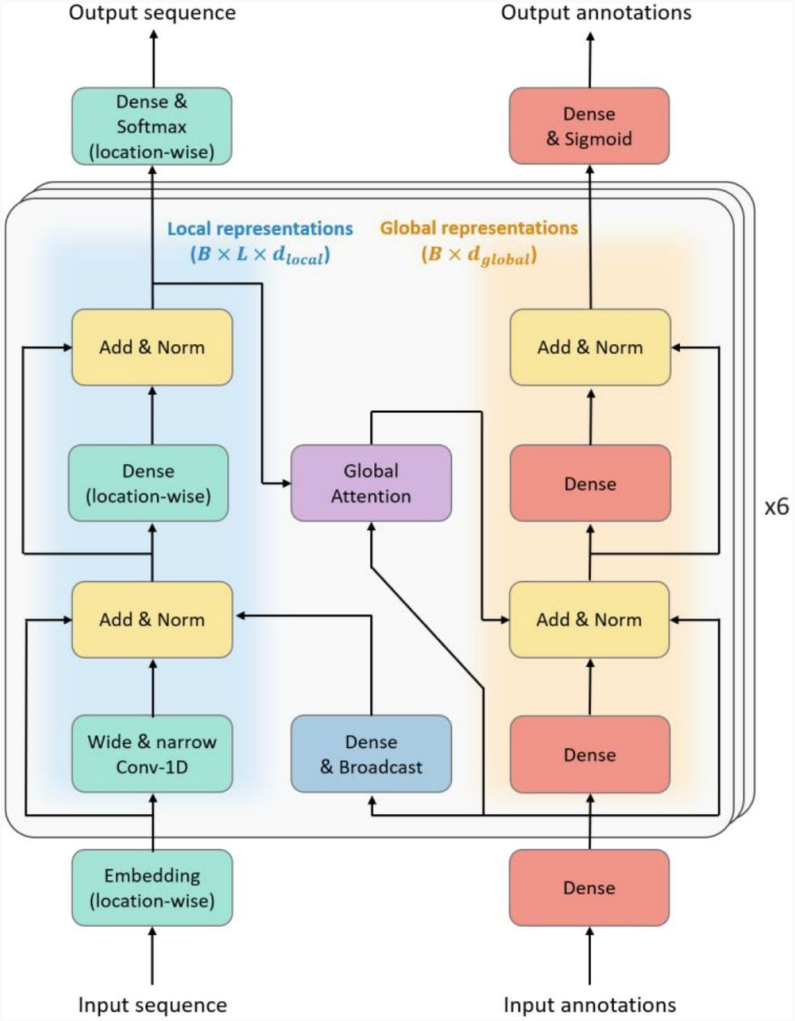
Performance of fine-tuned ProteinBERT models over 4 TAPE benchmarks as a function of pretraining amount (measured by number of processed proteins).

- Massive improvement vs no pretraining
- Pretraining benefits saturate for some tasks

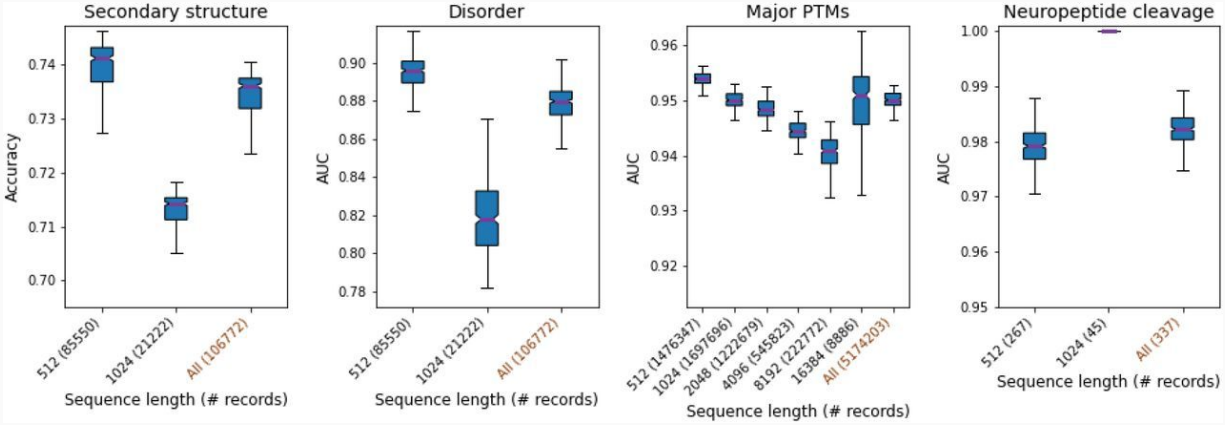


ProteinBERT Architecture

- GO Annotations task & inputs
- Global attention (Linear complexity, interpretable)
- No sequence self-attention
- 1D convolutions
- Denoising instead of Masked Language modelling
- No positional embeddings
- No dropout/regularization
- Sequence-length agnostic
- Smaller and faster (16M vs 38M for TAPE-Transformer, 110M BERT, 650M ESM)



Performance across sequence lengths



Test-set performance of fine-tuned ProteinBERT models with different input sequence lengths.

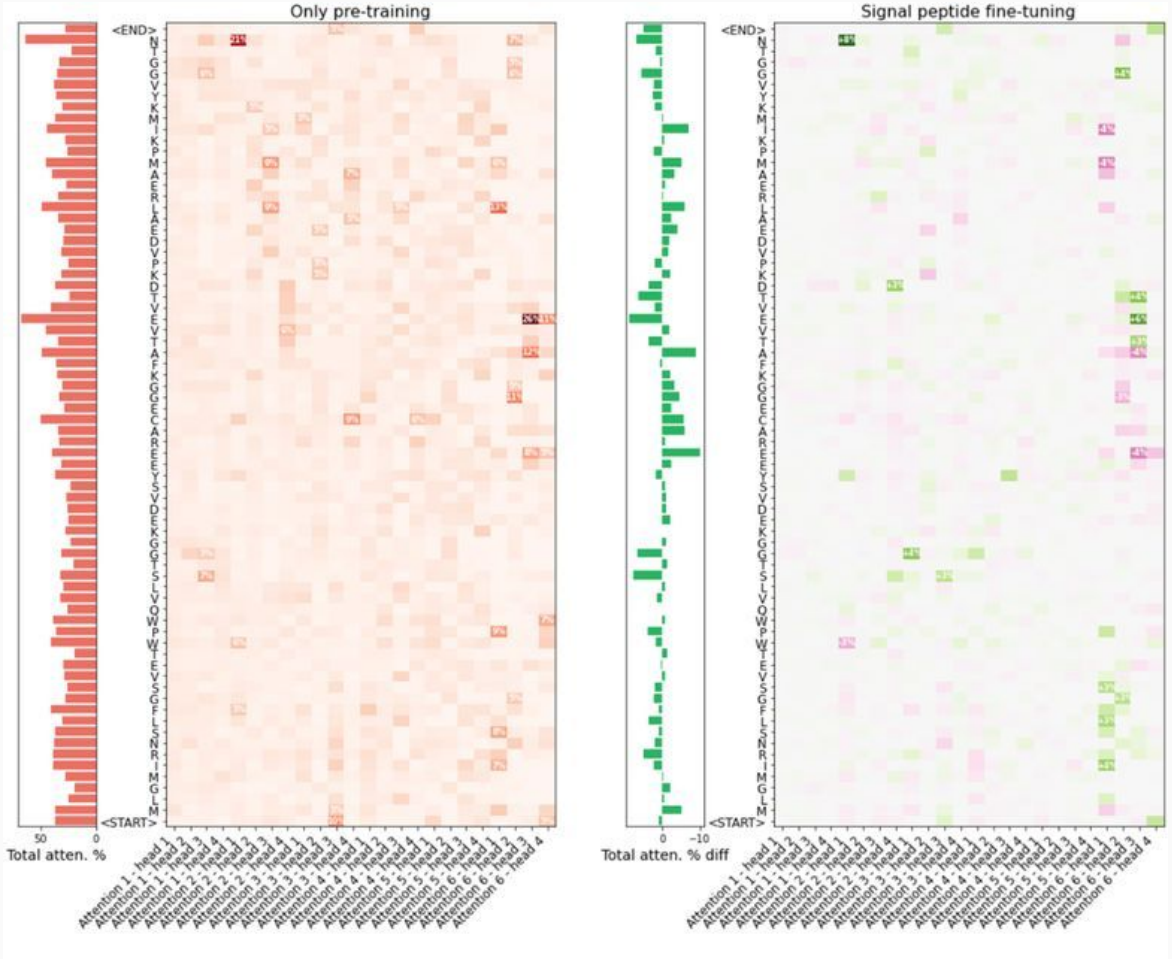
Understanding global attention: Signal Peptide prediction

Global attention before and after fine-tuning on signal peptide prediction.

Heme-binding protein 1 (Hebp1)

Left = Attention values per head & layer from unsupervised language model

Right: attention in fine-tuned ProteinBert model



Takeaways

- Pretraining works, and Proteins are similar enough to languages to benefit from self-supervised NLP models
- Proteins != language - models should take advantage of this
- Unsupervised representations contain a lot of information
- You don't need "thousands of GPUs" to get a ~SOTA model, even with pretraining!
- **ProteinBert = especially useful for "small data" tasks - easy to adapt to new tasks**

New Task?

Take a pretrained model and try transfer learning!

Code:

https://github.com/nadavbra/protein_bert

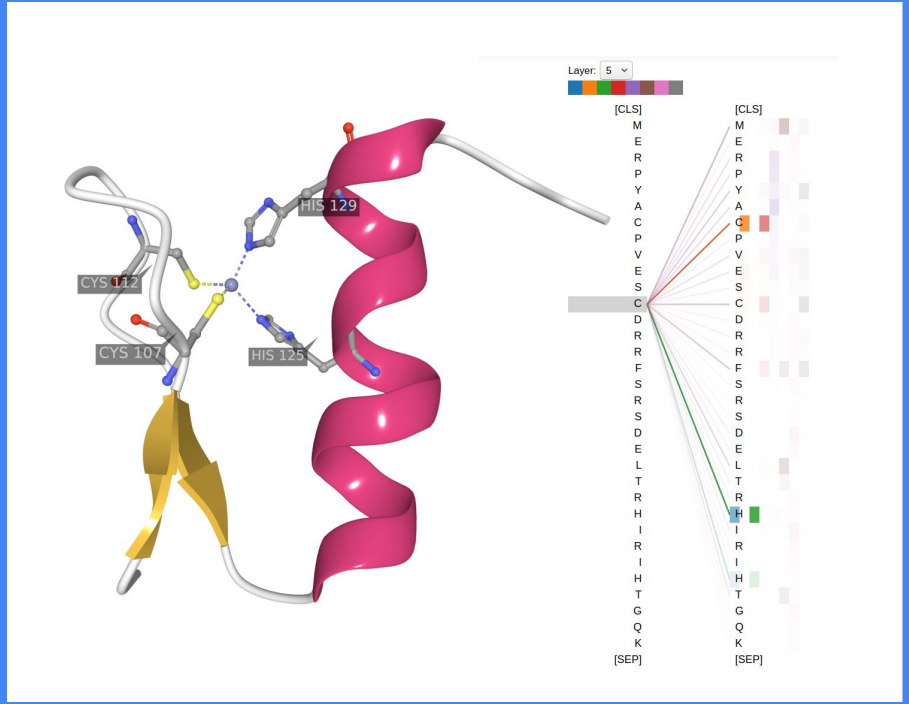


Image source: https://github.com/agemagician/ProtTrans#prottrans

“My money is on self-supervised learning”

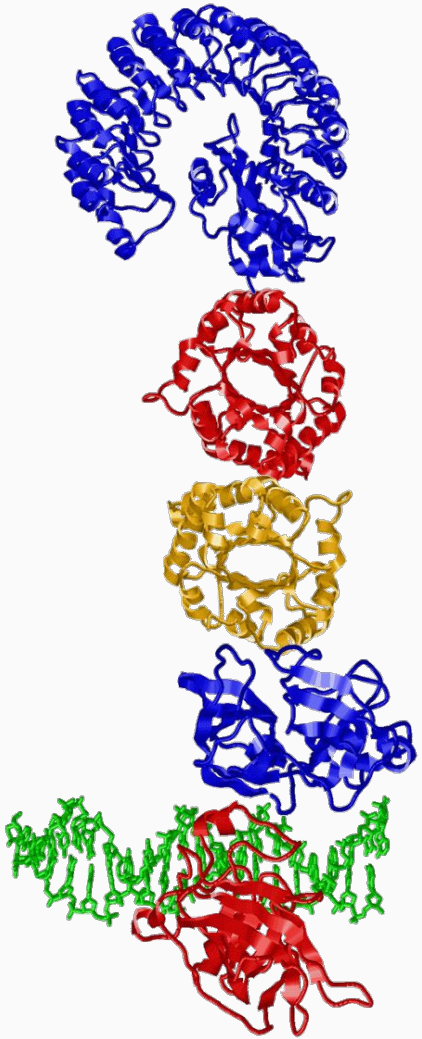
- Yann LeCunn

Thanks!

Contact us:

Dan Ofer
ddofer@gmail.com

Linial Lab:
https://michal-linial.huji.ac.il



Paper references

ProteinBERT: A universal deep-learning model of protein sequence and function

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, Michal Linial

bioRxiv. doi: https://doi.org/10.1101/2021.05.24.445464

Code: https://github.com/nadavbra/protein_bert

The language of proteins: NLP, machine learning & protein sequences

Dan Ofer, Nadav Brandes, Michal Linial

https://doi.org/10.1016/j.csbj.2021.03.022