

# **Linguistic Knowledge and Transferability of Contextual Representations**

Nelson F. Liu Matt Gardner Yonatan Belinkov

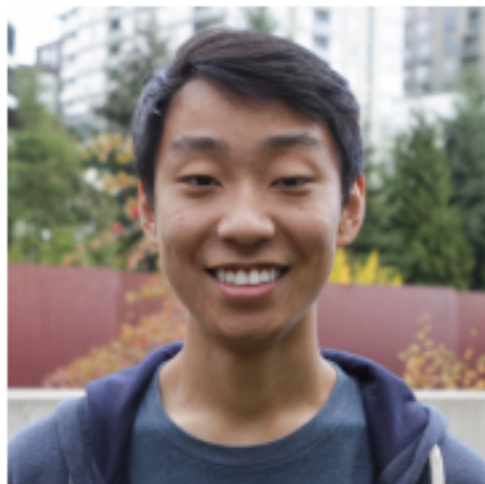
Matthew E. Peters Noah A. Smith

**NAACL19**

# Outline

- Author
- Tasks
- Model
- Experiment
- Multilingual BERT
- Conclusion

# Author



- Nelson F. Liu
- University of Washington
- B.S. Undergraduate
- **Scikit-learn** : Google Summer of Code Developer
- **AllenNLP** : research intern

## PUBLICATIONS

- [1] *Linguistic Knowledge and Transferability of Contextual Representations*  
**Nelson F. Liu**, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. To appear in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2019). June 2019.
- [2] *Inoculation by Fine-Tuning: A Method for Analyzing Challenge Datasets*  
**Nelson F. Liu**, Roy Schwartz, and Noah A. Smith. To appear in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2019). June 2019.
- [3] *LSTMs Exploit Linguistic Attributes of Data*  
**Nelson F. Liu**, Omer Levy, Roy Schwartz, Chenhao Tan and Noah A. Smith. In *Proceedings of the ACL Workshop on Representation Learning for NLP* (RepL4NLP 2018). July 2018.  
**Best Paper Award.**
- [4] *AllenNLP: A Deep Semantic Natural Language Processing Platform*  
Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, **Nelson F. Liu**, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. In *Proceedings of the ACL Workshop for Natural Language Processing Open Source Software* (NLP-OSS 2018). July 2018.
- [5] *Discovering Phonesthemes with Sparse Regularization*  
**Nelson F. Liu**, Gina-Anne Levow, and Noah A. Smith. In *Proceedings of the NAACL Workshop on Subword and Character Level Models in NLP* (SCLeM 2018). June 2018.
- [6] *ELISA System Description for LoReHLT 2017*  
Leon Cheung, Thamme Gowda, Ulf Hermjakob, **Nelson Liu**, Jonathan May, Alexandra Mayn, Nima Pourdamghani, Michael Pust, Kevin Knight, 32 others (names ordered alphabetically and by affiliation). In *Proceedings of the NIST LoReHLT 2017 Workshop*. September 2017.
- [7] *Crowdsourcing Multiple Choice Science Questions*  
Johannes Welbl, **Nelson F. Liu**, and Matt Gardner. In *Proceedings of the EMNLP Workshop on Noisy User-generated Text* (WNUT 2017). September 2017.

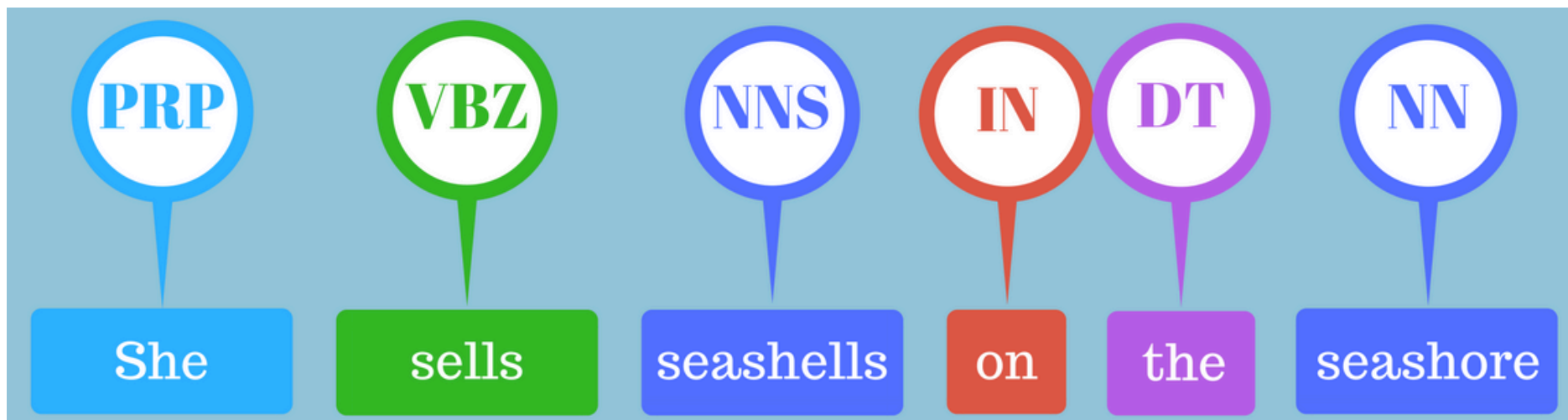
# Outline

- Author
- Tasks
  - **Sixteen diverse English probing tasks**
    - 1. Token Labeling**
    2. Segmentation
    3. Pairwise Relations
- Model
- Experiment
- Multilingual BERT
- Conclusion

# Part-of-speech tagging (POS)

- Whether CWRs capture **basic syntax**

人称代名词      动词（现在时态，第三人称单数）      名词（复数）      介词      限定词      名词（单数）



她在海边卖贝壳

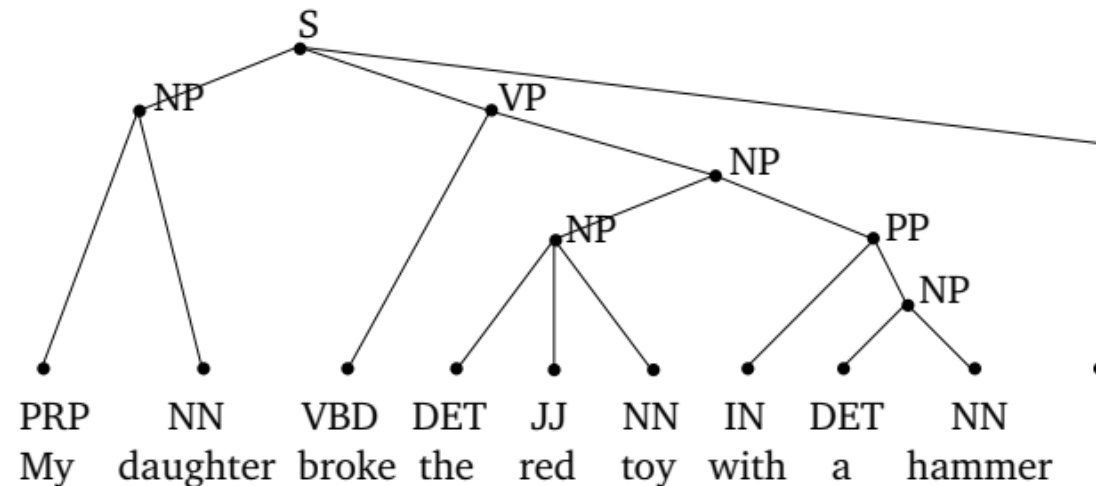
# CCG supertagging (CCG)

- The vectors' **fine-grained information about the syntactic roles of words in context.**
- CCG is lexicalized grammar formalism that has two kinds of categories:
- **atomic categories** (S, N, NP, and PP for sentence, nouns, noun phrases and prepositional phrases, respectively)
- **complex categories** that contain two parts: an argument and a result, denoted by slashes ('\ or '/') indicating whether the argument is expected to lie to the right or left

Mr.	Vinken	is	chairman	of	Elsevier	N.V.	,	the	Dutch	publishing	group	.
N/N	N	(S[decl]\NP)/NP	N	(NP\NP)/NP	N/N	N	,	NP[nb]/N	N/N	N/N	N	.

# Syntactic constituency ancestor tagging

- The vectors' knowledge of **hierarchical syntax**.
- **Constituent parsing** is a core problem in NLP where the goal is to obtain the syntactic structure of sentences expressed as a **phrase structure tree**.
- For a given word, the probing model is trained to predict the constituent label of its **parent (Parent)**, **grandparent (GParent)**, or **great-grandparent (GGParent)** in the phrase-structure tree (from the PTB).



# Semantic tagging task (ST)

- Tokens are assigned labels that reflect their **semantic role** in context.
- These semantic tags assess **lexical semantics**.

疑问 程度 现在时态 确切

How<sup>QUE</sup> tall<sup>DEG</sup> is<sup>NOW</sup> the<sup>DEF</sup> green\_monster<sup>ART</sup> at<sup>REL</sup> Fenway<sup>GEO</sup> ?<sup>QUE</sup>  
My<sup>HAS</sup> sister<sup>ROL</sup> went<sup>EPS</sup> to<sup>REL</sup> the<sup>DEF</sup> United\_States<sup>GPE</sup> to<sup>SUB</sup> study<sup>EXS</sup> English<sup>CON</sup> .<sup>NIL</sup>  
Any<sup>AND</sup> contribution<sup>CON</sup> was<sup>PST</sup> appreciated<sup>EXS</sup> but<sup>BUT</sup> we<sup>PRO</sup> have<sup>NOW</sup> n't<sup>NOT</sup> got<sup>EXT</sup> any<sup>DIS</sup> .<sup>NIL</sup>  
He<sup>PRO</sup> himself<sup>EMP</sup> can<sup>POS</sup> earn<sup>EXS</sup> \$<sup>UOM</sup> 100<sup>QUC</sup> a<sup>AND</sup> day<sup>UOM</sup> .<sup>NIL</sup>



# Preposition supersense disambiguation

- This task is a specialized kind of word sense disambiguation, and examines one facet of **lexical semantic knowledge**.
- The model is trained and evaluated on single-token prepositions (rather than making a decision for every token in a sequence).

- (1) I was booked **for**/DURATION 2 nights **at**/LOCUS this hotel **in**/TIME Oct 2007 .
- (2) I went **to**/GOAL ohm **after**/EXPLANATION $\rightsquigarrow$ TIME reading some **of**/QUANTITY $\rightsquigarrow$ WHOLE the reviews .
- (3) It was very upsetting to see this kind **of**/SPECIES behavior especially **in\_front\_of**/LOCUS **my**/SOCIALREL $\rightsquigarrow$ GESTALT four year\_old .

# Event factuality (EF) task

- Labeling **phrases** with the factuality of the events they describe
- The model is trained to predict a (non)factuality value in the **range**  $[-3, 3]$ .
- This task is treated as a **regression problem**, where a prediction is made only **for tokens corresponding to events** (rather than every token in a sequence).

Jo didn't remember to **leave**. leaving did not happen

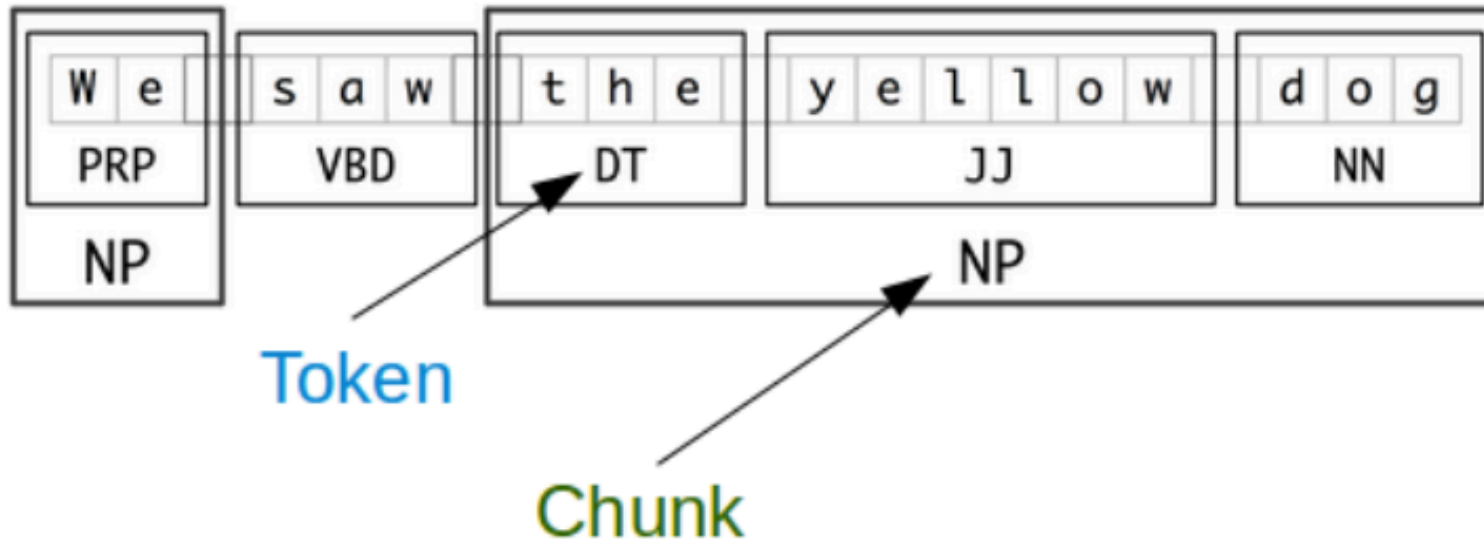
Jo didn't remember **leaving**.

# Outline

- Author
- Tasks
  - **Sixteen diverse English probing tasks**
    1. Token Labeling
    - 2. Segmentation**
    3. Pairwise Relations
- Model
- Experiment
- Multilingual BERT
- Conclusion

# Syntactic chunking (Chunk)

- Whether CWR s contain notions of **spans and boundaries**
- Segment text into shallow constituent chunks.



# Named entity recognition (NER)

- Whether CWRs encode information about **entity types**.

At the W party Date Thursday Time night at Location Chateau Marmont, Person Cate Blanchett barely made it up in the elevator.

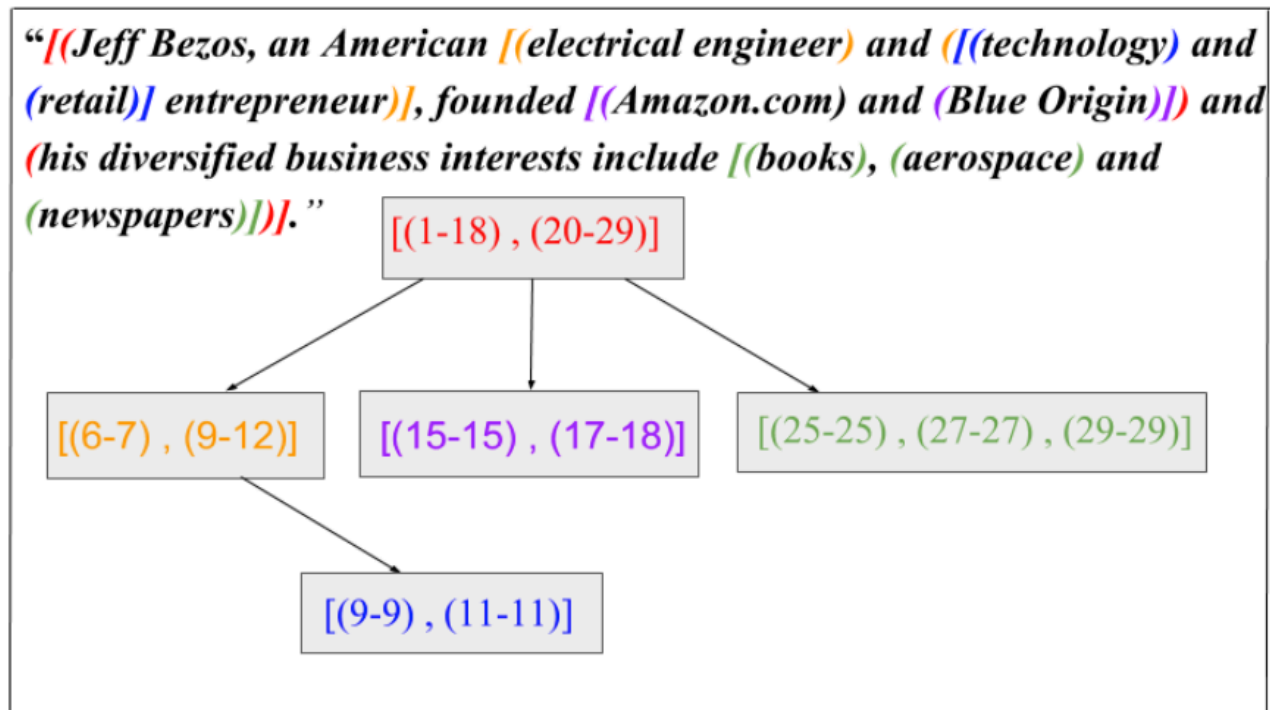
# Grammatical error detection (GED)

- Whether embeddings encode features that indicate **anomalies in their input**
- Task of identifying tokens which need to be edited in order to produce a grammatically correct sentence.

Grammatical Error Detection	I	am	here	at	business
1) Grammaticality Checking	0	0	0	1	0
2) Error Type Classification	None	None	None	PRP_LXC	None

# Conjunct identification (Conj)

- Requires **highly specific syntactic knowledge**.



# Outline

- Author
- Tasks
  - **Sixteen diverse English probing tasks**
    1. Token Labeling
    2. Segmentation
    - 3. Pairwise Relations**
- Model
- Experiment
- Multilingual BERT
- Conclusion



# Pairwise Relations

- Examine whether **relationships between words** are encoded in CWRs.
- **Syntactic dependency arc prediction**
  - The model is trained to predict whether the sentence's syntactic dependency parse contains a dependency arc two words
- **syntactic dependency arc classification**
  - The model is trained to predict the type of syntactic relation that link them (the label on that dependency arc).
- **Semantic dependency arc prediction**
- **Semantic dependency arc classification**
- **Coreference arc prediction**
  - The model is trained to predict whether two entities corefer from their CWRs.

# Outline

- Author
- Tasks
- **Model**
- Experiment
- Multilingual BERT
- Conclusion

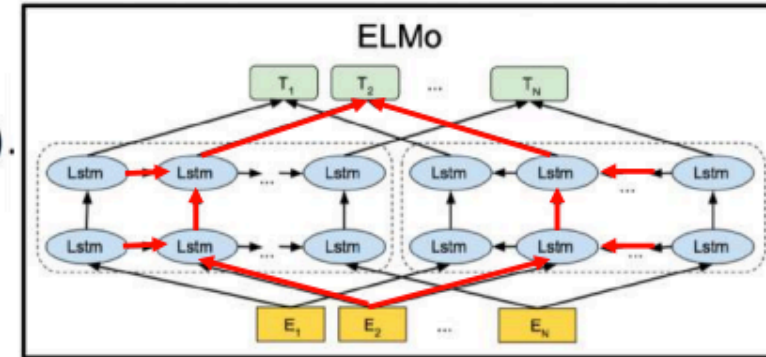
# ELMo (Embeddings from Language Models)

- **Forward** language model

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1}).$$

- **Backward** language model

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N)$$



- Jointly maximizes the log likelihood of the forward and backward directions

$$\sum_{k=1}^N ( \log p(t_k | t_1, \dots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s) + \log p(t_k | t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) )$$

$\Theta_x$  Token representation

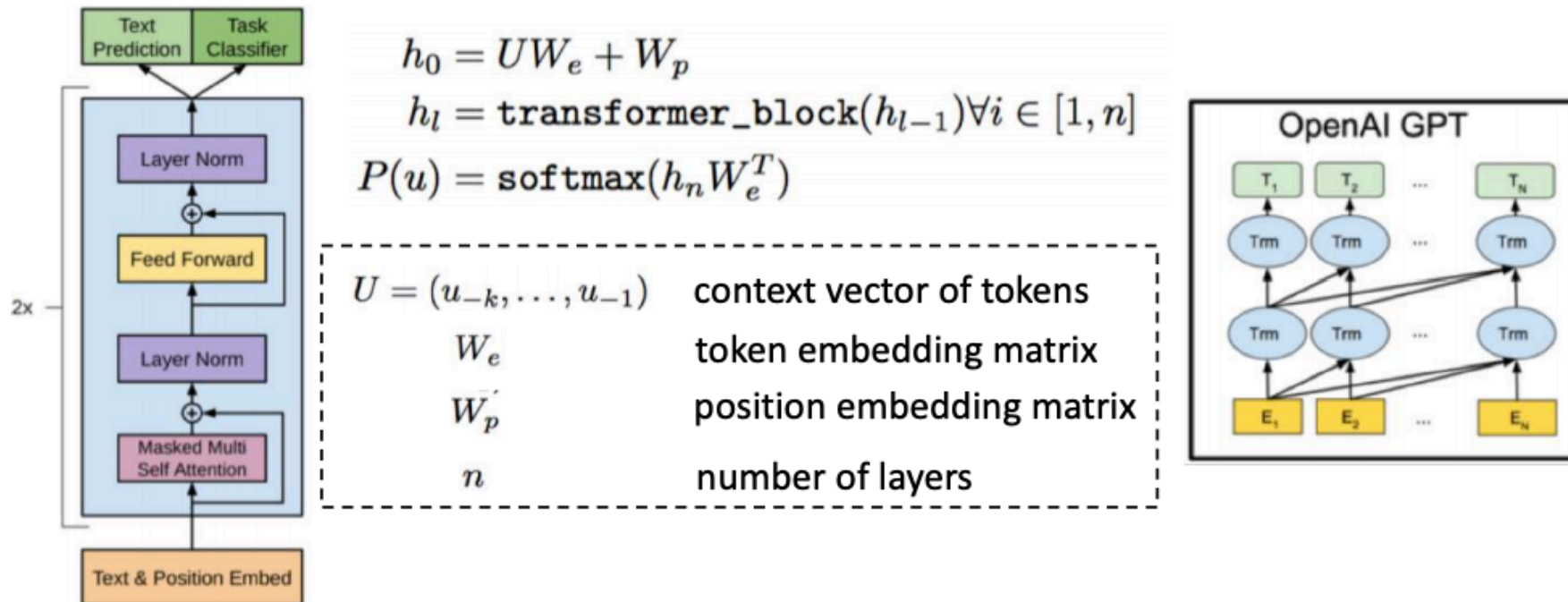
$\Theta_s$  Softmax layer

# OpenAI GPT(Generative Pre-trained Transformer)

- Use a **standard language modeling objective** to maximize the following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

- A **multi-layer transformer decoder** for the language model



# BERT(Bidirectional Encoder Representations from Transformers)

- BERT's model architecture is a **multi-layer bidirectional Transformer encoder**.

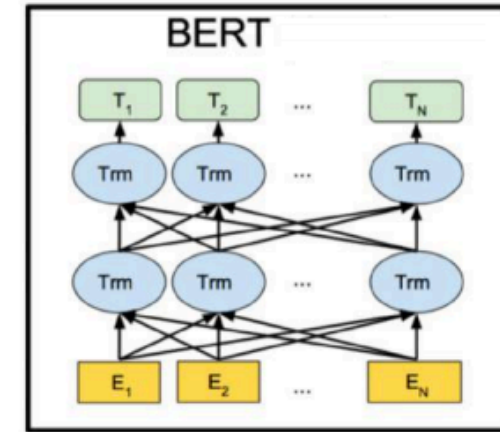
- L: number of layers
- H: hidden size
- A: number of self-attention heads.

- Model

- **BERT<sub>BASE</sub>** : L=12, H=768, A=12, Total Parameters=110M *(have an identical model size as OpenAI GPT for comparison purposes)*
- **BERT<sub>LARGE</sub>** : L=24, H=1024, A=16, Total Parameters=340M

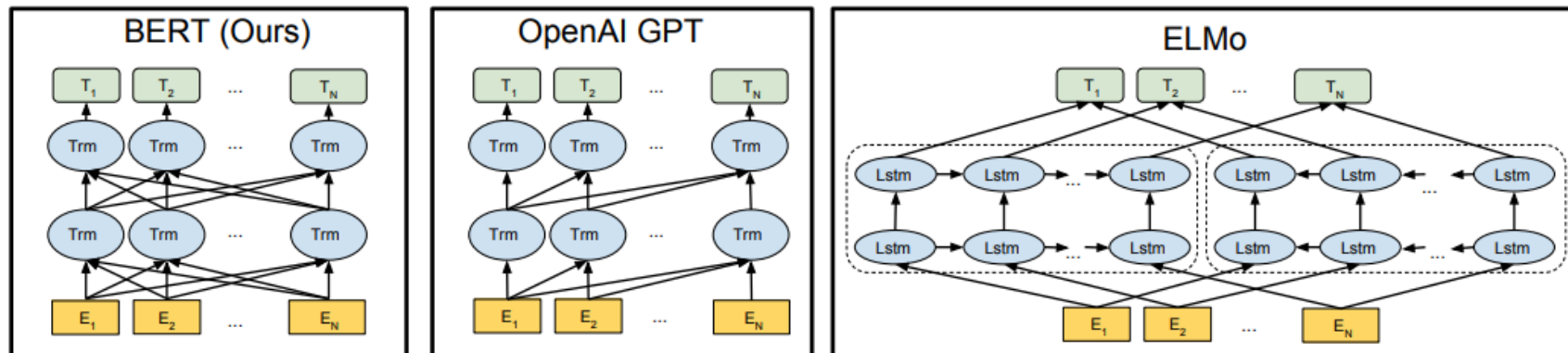
- Note:

- BERT: Bidirectional Transformer **encoder**
- OpenAI: Left-context-only Transformer **decoder**



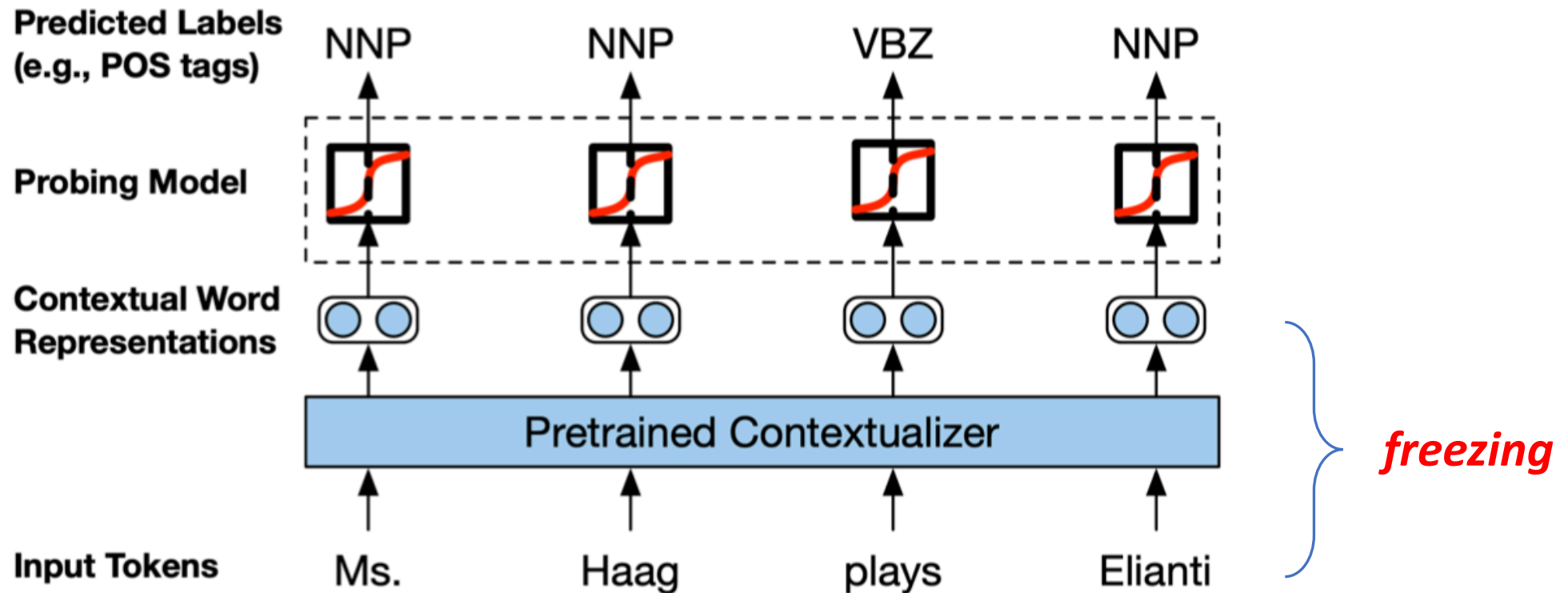
# Contextualizers

- **ELMo** (original) uses a 2-layer LSTM
- **ELMo** (4 layer) uses a 4-layer LSTM
- **ELMo** (transformer) uses a 6-layer transformer
- **OpenAI transformer** is a left-to-right 12-layer transformer language model
- **BERT** (base, cased), which uses a 12-layer transformer
- **BERT** (large, cased), which uses a 24-layer transformer



# Probing Model

- Use a **linear model** as our probing model; **limiting its capacity** enables us to **focus** on what information can be easily extracted from CWRs.



# Outline

- Author
- Tasks
- Model
- Experiment
  1. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks
  2. Universal Language Model Fine-tuning for Text Classification ACL18
- Multilingual BERT
- Conclusion



# Results and Discussion

## Tasks

## Models

Pretrained Representation			POS		Chunk	NER	ST	GED	Supersense ID		EF
	Avg.	CCG	PTB	EWT					PS-Role	PS-Fxn	
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

# Results and Discussion

ELMo (original), Layer 0	78.27	77.73	82.05	78.52
ELMo (original), Layer 1	89.04	86.46	96.13	93.01
ELMo (original), Layer 2	88.33	85.34	94.72	91.32
ELMo (original), Scalar Mix	89.30	86.56	95.81	91.69

Pretrained Representation	POS					Supersense ID					
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- The **Best layerwise** linear probing model for each contextualizer.
- A **GloVe-based** linear probing baseline.
- The previous state of the art.(**SOTA**)

# Results and Discussion

Pretrained Representation	POS			Supersense ID							
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- In all cases, **CWRs** perform significantly **better** than **the noncontextual baseline**.

# Results and Discussion

Pretrained Representation	POS							Supersense ID			
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- Probing models **rivaling or exceeding** the performance of (often carefully tuned and task-specific) state-of-the-art models.

# Results and Discussion

Pretrained Representation	POS							Supersense ID			
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- ELMo (4-layer) and ELMo (original) are essentially even, though **both recurrent models outperform ELMo (transformer)**.



# Results and Discussion

Pretrained Representation	POS			Supersense ID							
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- OpenAI transformer significantly **underperforms** the ELMo models and BERT. Given that it is also the only model trained in a **unidirectional (left-to-right) fashion**, this reaffirms that **bidirectionality is a crucial component for the highest quality contextualizers**
- The OpenAI transformer is the only model trained on **lowercased text**, which hinders its performance on tasks like NER.

# Results and Discussion

Pretrained Representation				POS					Supersense ID		
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- BERT significantly improves over the ELMo and OpenAI models.

# Results and Discussion

Pretrained Representation	POS					Supersense ID					
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- Current methods for CWR **do not** capture much transferable information about **entities** and **coreference** phenomena in their input



# Probing Failures

Pretrained Representation	POS					Supersense ID					
	Avg.	CCG	PTB	EWT	Chunk	NER	ST	GED	PS-Role	PS-Fxn	EF
ELMo (original) best layer	81.58	93.31	97.26	95.61	90.04	82.85	93.82	29.37	75.44	84.87	73.20
ELMo (4-layer) best layer	81.58	93.81	<b>97.31</b>	95.60	89.78	82.06	<b>94.18</b>	29.24	74.78	85.96	73.03
ELMo (transformer) best layer	80.97	92.68	97.09	95.13	93.06	81.21	93.78	30.80	72.81	82.24	70.88
OpenAI transformer best layer	75.01	82.69	93.82	91.28	86.06	58.14	87.81	33.10	66.23	76.97	74.03
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71	93.72	43.30	<b>79.61</b>	87.94	75.11
BERT (large, cased) best layer	<b>85.07</b>	<b>94.28</b>	96.73	<b>95.80</b>	<b>93.64</b>	<b>84.44</b>	93.83	<b>46.46</b>	79.17	<b>90.13</b>	<b>76.25</b>
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22	80.92	14.94	40.79	51.54	49.70
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38	95.15	39.83	66.89	78.29	77.10

- The CWR simply **does not encode** the pertinent information or any predictive correlates
- The **probing model does not have the capacity** necessary to extract the information or predictive correlates from the vector.

# Probing Failures

Probing Model	NER	GED	Conj	GGParent
Linear	82.85	29.37	38.72	67.50
MLP (1024d)	87.19	47.45	55.09	78.80
LSTM (200d) + Linear	<b>88.08</b>	<b>48.90</b>	<b>78.21</b>	<b>84.96</b>
BiLSTM (512d) + MLP (1024d)	90.05	48.34	87.07	90.38

Adding more parameters  
to the probing model

Task-trained LSTM

Full-featured model

ELMo (original) pretrained contextualizer

# Probing Failures

Probing Model	NER	GED	Conj	GGParent
Linear	82.85	29.37	38.72	67.50
MLP (1024d)	87.19	47.45	55.09	78.80
LSTM (200d) + Linear	<b>88.08</b>	<b>48.90</b>	<b>78.21</b>	<b>84.96</b>
BiLSTM (512d) + MLP (1024d)	90.05	48.34	87.07	90.38

- **Adding more parameters** (either by replacing the linear model with a MLP, or using a contextual probing model) leads to **significant gains** over the linear probing model

# Probing Failures

Probing Model	NER	GED	Conj	GGParent
Linear	82.85	29.37	38.72	67.50
MLP (1024d)	87.19	47.45	55.09	78.80
LSTM (200d) + Linear	<b>88.08</b>	<b>48.90</b>	<b>78.21</b>	<b>84.96</b>
BiLSTM (512d) + MLP (1024d)	90.05	48.34	87.07	90.38

- Very similar performance between the MLP and LSTM + Linear models—this indicates **that the probing model simply needed more capacity** to extract the necessary information from the CWRs.

# Probing Failures

Probing Model		NER	GED	Conj	GGParent
nearly the same number of parameters	Linear	82.85	29.37	38.72	67.50
	MLP (1024d)	87.19	47.45	55.09	78.80
	LSTM (200d) + Linear	<b>88.08</b>	<b>48.90</b>	<b>78.21</b>	<b>84.96</b>
BiLSTM (512d) + MLP (1024d)		90.05	48.34	87.07	90.38

- Adding parameters as a **task-trained component** of our probing model leads to large gains over **simply adding parameters** to the probing model.
- This indicates that the **pretrained contextualizers do not capture the information necessary for the task**, since such information is learnable by a task-specific contextualizer.

# Probing Failures

Probing Model	NER	GED	Conj	GGParent
Linear	82.85	29.37	38.72	67.50
MLP (1024d)	87.19	47.45	55.09	78.80
LSTM (200d) + Linear	<b>88.08</b>	<b>48.90</b>	<b>78.21</b>	<b>84.96</b>
BiLSTM (512d) + MLP (1024d)	90.05	48.34	87.07	90.38

- Confirm that **task-trained contextualization** is important when the end task requires specific information that may not be captured by the pretraining task
- Such end-task specific contextualization can come from either **fine-tuning CWRs** or using **fixed output features as inputs to a task-trained contextualizer**

# To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks

Pretraining	Adaptation	NER	SA	Nat. lang. inference		Semantic textual similarity		
		CoNLL 2003	SST-2	MNLI	SICK-E	SICK-R	MRPC	STS-B
Skip-thoughts	❄️	-	81.8	62.9	-	86.6	75.8	71.8
ELMo	❄️	91.7	<b>91.8</b>	<b>79.6</b>	<b>86.3</b>	<b>86.1</b>	<b>76.0</b>	<b>75.9</b>
	🔥	<b>91.9</b>	91.2	76.4	83.3	83.3	74.7	75.5
	$\Delta = \text{🔥} - \text{❄️}$	0.2	-0.6	-3.2	-3.3	-2.8	-1.3	-0.4
BERT-base	❄️	92.2	93.0	<b>84.6</b>	84.8	86.4	78.1	82.9
	🔥	<b>92.4</b>	<b>93.5</b>	<b>84.6</b>	<b>85.8</b>	<b>88.7</b>	<b>84.8</b>	<b>87.1</b>
	$\Delta = \text{🔥} - \text{❄️}$	0.2	0.5	0.0	1.0	2.3	6.7	4.2








Feature based









Fine tune

# To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks

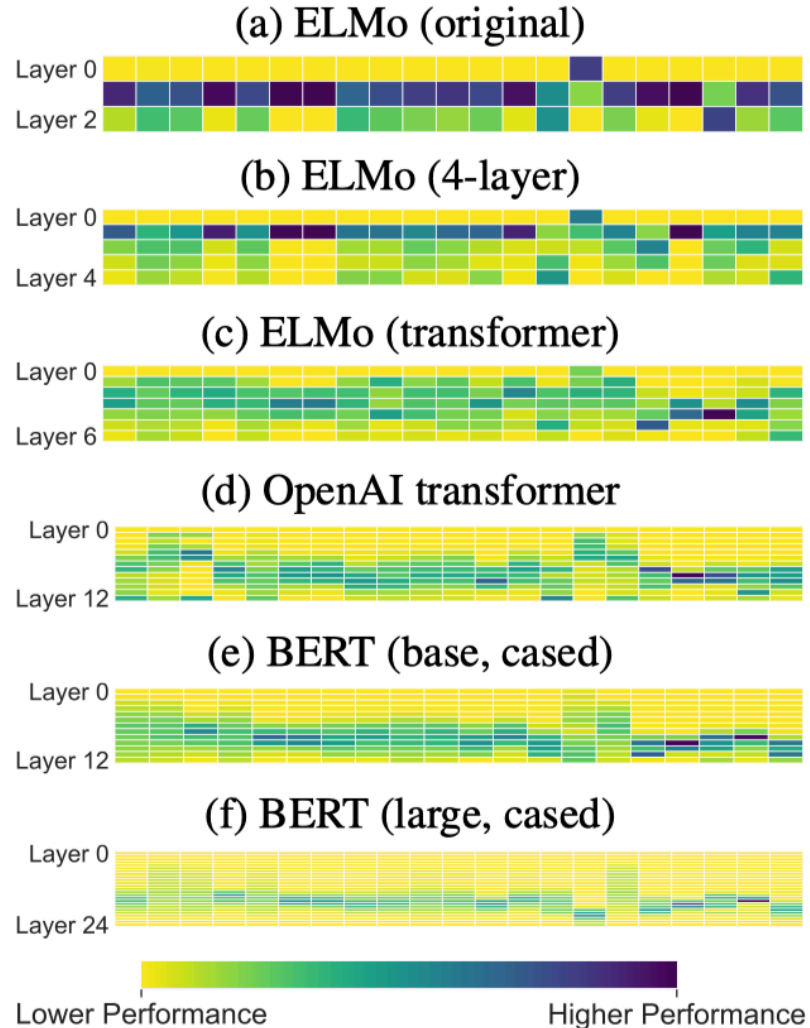
Model configuration	F <sub>1</sub>
 + BiLSTM + CRF	<b>95.5</b>
 + CRF	91.9
 + CRF + gradual unfreeze	<b>95.5</b>
 + BiLSTM + CRF + gradual unfreeze	95.2
 + CRF	95.1

NER

Conditions			Guidelines
Pretrain	Adapt.	Task	
Any		Any	Add many task parameters
Any		Any	Add minimal task parameters ⚠ Hyper-parameters
Any	Any	Seq. / clas.	 and  have similar performance
ELMo	Any	Sent. pair	use 
BERT	Any	Sent. pair	use 



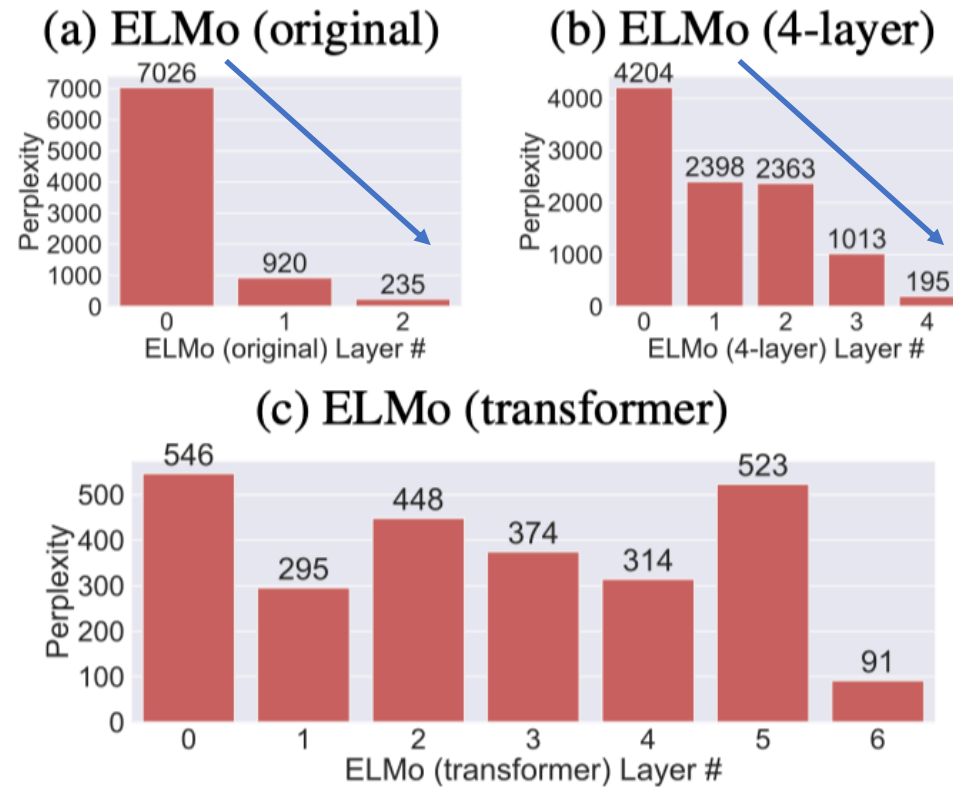
# Analyzing Layer-wise Transferability



The first layer of contextualization in recurrent models (original and 4-layer ELMo) is consistently the most transferable

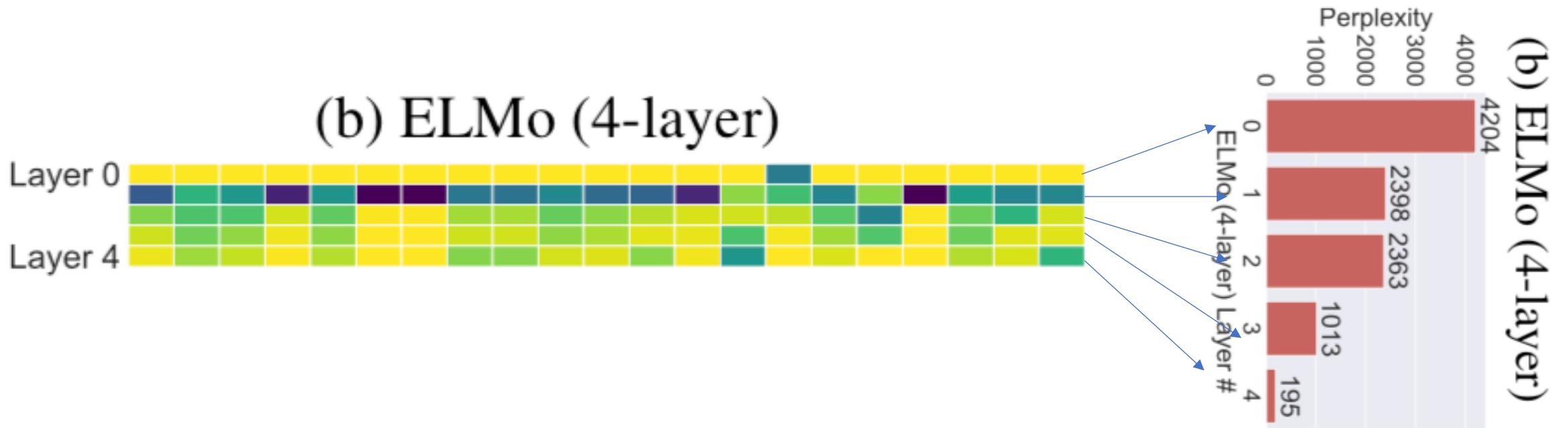
Transformer-based contextualizers have no single most-transferable layer; the best performing layer for each task varies, and is usually near the middle.

# Analyzing Layer-wise Transferability



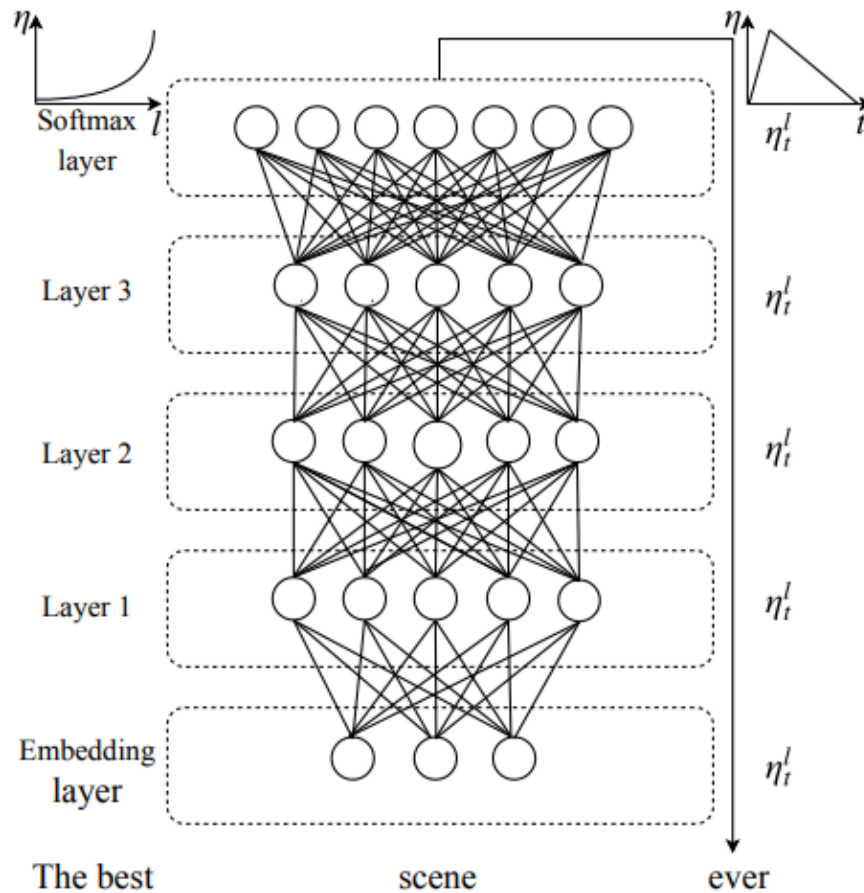
- Higher layers in recurrent models consistently achieve lower perplexities.
- The layers of the ELMo (transformer) model do not exhibit such a monotonic increase. While the topmost layer is best

# Analyzing Layer-wise Transferability



- Contextualizer layers **trade off** between **encoding general** and **task-specific features**.

# Universal Language Model Fine-tuning for Text Classification ACL18



$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^l} J(\theta)$$

$$\eta^{l-1} = \eta^l / 2.6$$

- The model layers are **progressively unfrozen** (starting from the final layer) during the finetuning process.
- **Higher-level LSTM layers** are less general (and more pretraining task-specific), they likely have to be **finetuned a bit more** in order to make them appropriately task specific.

# Transferring Between Tasks

Pretraining Task	Layer Average Target Task Performance			
	0	1	2	Mix
CCG	56.70	64.45	63.71	66.06
Chunk	54.27	62.69	63.25	63.96
POS	56.21	63.86	64.15	65.13
Parent	54.57	62.46	61.67	64.31
GParent	55.50	62.94	62.91	64.96
GGParent	54.83	61.10	59.84	63.81
Syn. Arc Prediction	53.63	59.94	58.62	62.43
Syn. Arc Classification	56.15	64.41	63.60	66.07
Sem. Arc Prediction	53.19	54.69	53.04	59.84
Sem. Arc Classification	56.28	62.41	61.47	64.67
Conj	50.24	49.93	48.42	56.92
BiLM	66.53	65.91	65.82	66.49
GloVe (840B.300d)	60.55			
Untrained ELMo (original)	52.14	39.26	39.39	54.42
ELMo (original) (BiLM on 1B Benchmark)	64.40	79.05	77.72	78.90

Table 3: Performance (averaged across target tasks) of contextualizers pretrained on a variety of tasks.

- ELMo (original) architecture
- The training data from each of the pretraining tasks is taken from the **PTB**.
- Noncontextual baseline (GloVe)
- Randomly-initialized, untrained ELMo (original) baseline
- The ELMo (original) model pretrained on the Billion Word Benchmark

# Transferring Between Tasks

Pretraining Task	Layer Average Target Task Performance			
	0	1	2	Mix
CCG	56.70	64.45	63.71	66.06
Chunk	54.27	62.69	63.25	63.96
POS	56.21	63.86	64.15	65.13
Parent	54.57	62.46	61.67	64.31
GParent	55.50	62.94	62.91	64.96
GGParent	54.83	61.10	59.84	63.81
Syn. Arc Prediction	53.63	59.94	58.62	62.43
Syn. Arc Classification	56.15	64.41	63.60	66.07
Sem. Arc Prediction	53.19	54.69	53.04	59.84
Sem. Arc Classification	56.28	62.41	61.47	64.67
Conj	50.24	49.93	48.42	56.92
BiLM	66.53	65.91	65.82	66.49
GloVe (840B.300d)	60.55			
Untrained ELMo (original)	52.14	39.26	39.39	54.42
ELMo (original) (BiLM on 1B Benchmark)	64.40	79.05	77.72	78.90

Table 3: Performance (averaged across target tasks) of contextualizers pretrained on a variety of tasks.

- Bidirectional language modeling pretraining is the most effective on average.
- Stronger results from training on more data (the ELMo original BiLM trained on the Billion Word Benchmark).

# Transferring Between Tasks

- Pretraining on syntactic dependency arc prediction (PTB), CCG supertagging, chunking, the ancestor prediction tasks, and **semantic dependency arc classification** all give better performance than **bidirectional language model pretraining**.

# Outline

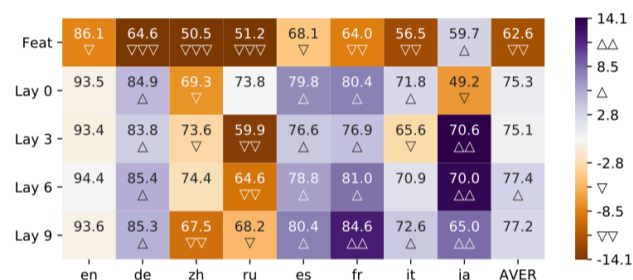
- Author
- Tasks
- Model
- Experiment
- **Multilingual BERT**
- Conclusion



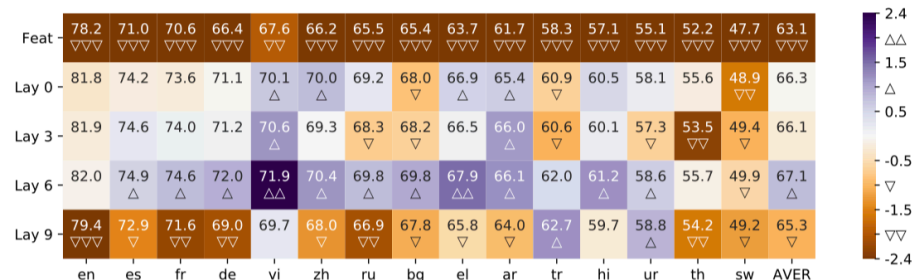
# Multilingual BERT

- <Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT>
- Except with data from Wikipedia in **104 languages**.
- Training makes no use of explicit cross-lingual signal
- **WordPiece** modeling strategy allows the model to share embeddings across languages
- **Subsample** words from languages with large Wikipedia and **oversample** words from languages with small Wikipedia
- **Zero shot cross-lingual transfer**, also known as single source transfer, refers to training and selecting a model in a source language, often a high resource language like English, then transferring directly to a target language.

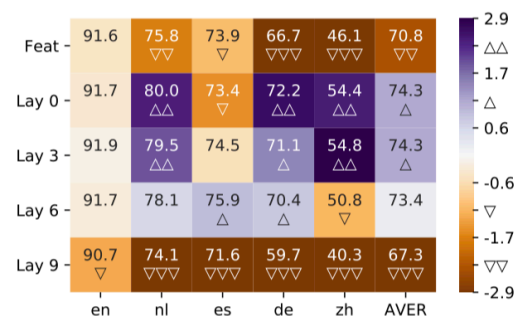
# Does mBERT vary layer-wise?



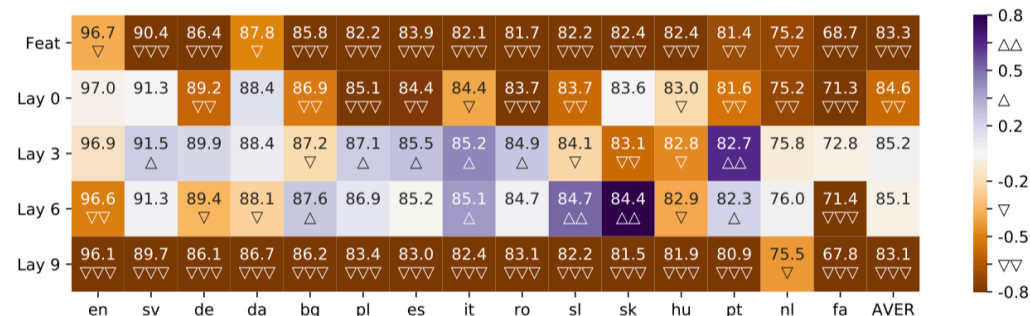
(a) Document classification (ACC)



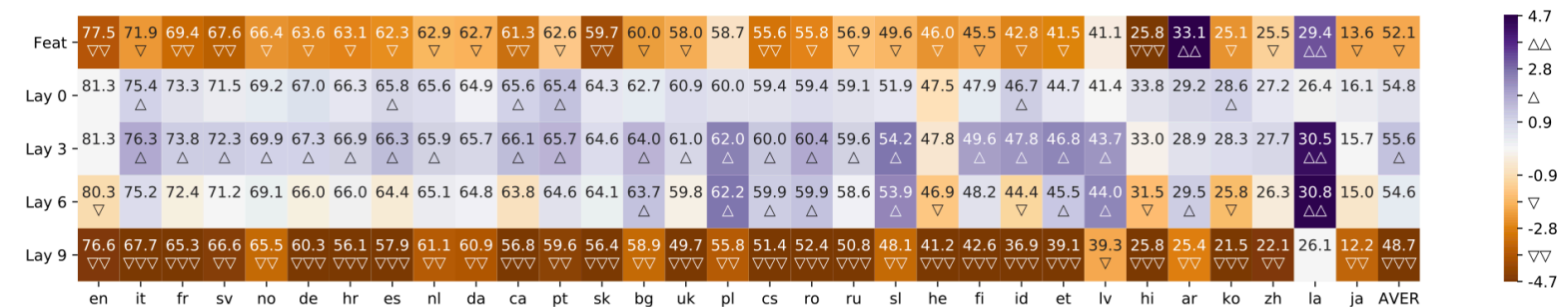
(b) Natural language inference (ACC)



(c) NER (F1)



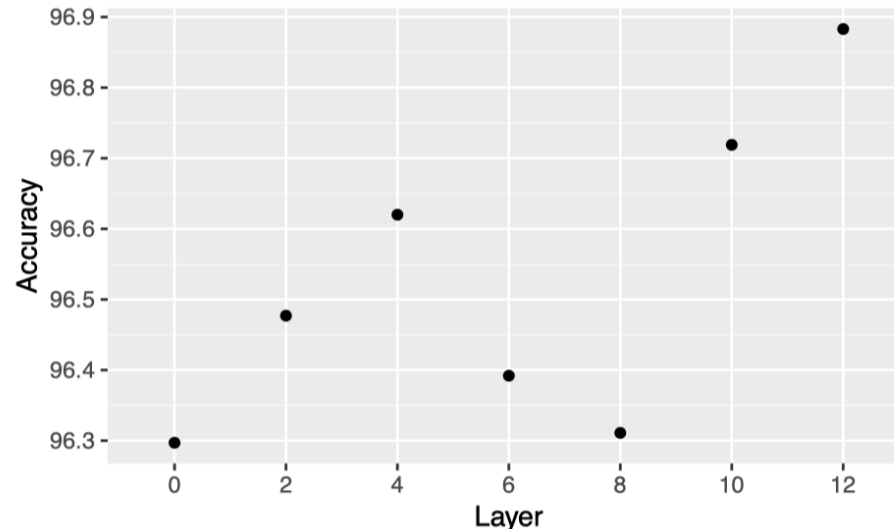
(d) POS tagging (ACC)



(e) Dependency parsing (LAS)

# Does mBERT retain language specific information?

- Since mBERT does so well at learning a crosslingual representation, it may do so by abstracting away from language specific information, thus **losing the ability to distinguish between languages**.
- **Task : Language identification**
- Across all tested layers around 96% accuracy
- mBERT needs to retain enough **language-specific information to perform the cloze task and select language-related subwords**.



# Conclusion

- CWRs (上下文词表征) 编码了语言的哪些feature ?
  - 在各类任务中, BERT > ELMo > GPT, 发现 “bidirectional” 是这类上下文编码器的必备要素
  - 相比于其他任务, 编码器们在NER和纠错任务表现较差 => 没有捕获到这方面信息
  - 在获得CWRs编码后, 再针对任务增加MLP(relu)或者LSTM会提升效果
  - 引出了问题: 什么时候直接fine-tune编码器? 什么时候freeze编码器, 增加task-specific layer?
- 编码器中不同层的transferability是怎样变化的?
  - 对于ELMo(LSTM)来说, 靠前的层更transferable, 靠后的层更task-specific
  - 对于transformer来说, 靠中间的层更transferable, 但是把各个层加权起来的效果会更好
  - 模型是有trade off的, 在任务上表现越好, 迁移性越差
- 预训练任务会对任务和transferability有怎样的影响?
  - 双向语言模型预训练出来平均效果越好
  - 预训练任务越接近特定任务, 在特定任务的表现越好
  - 预训练数据越多, 表现越好

Thanks!