Language Representation: from Language model to BERT

Zhenya Huang, USTC

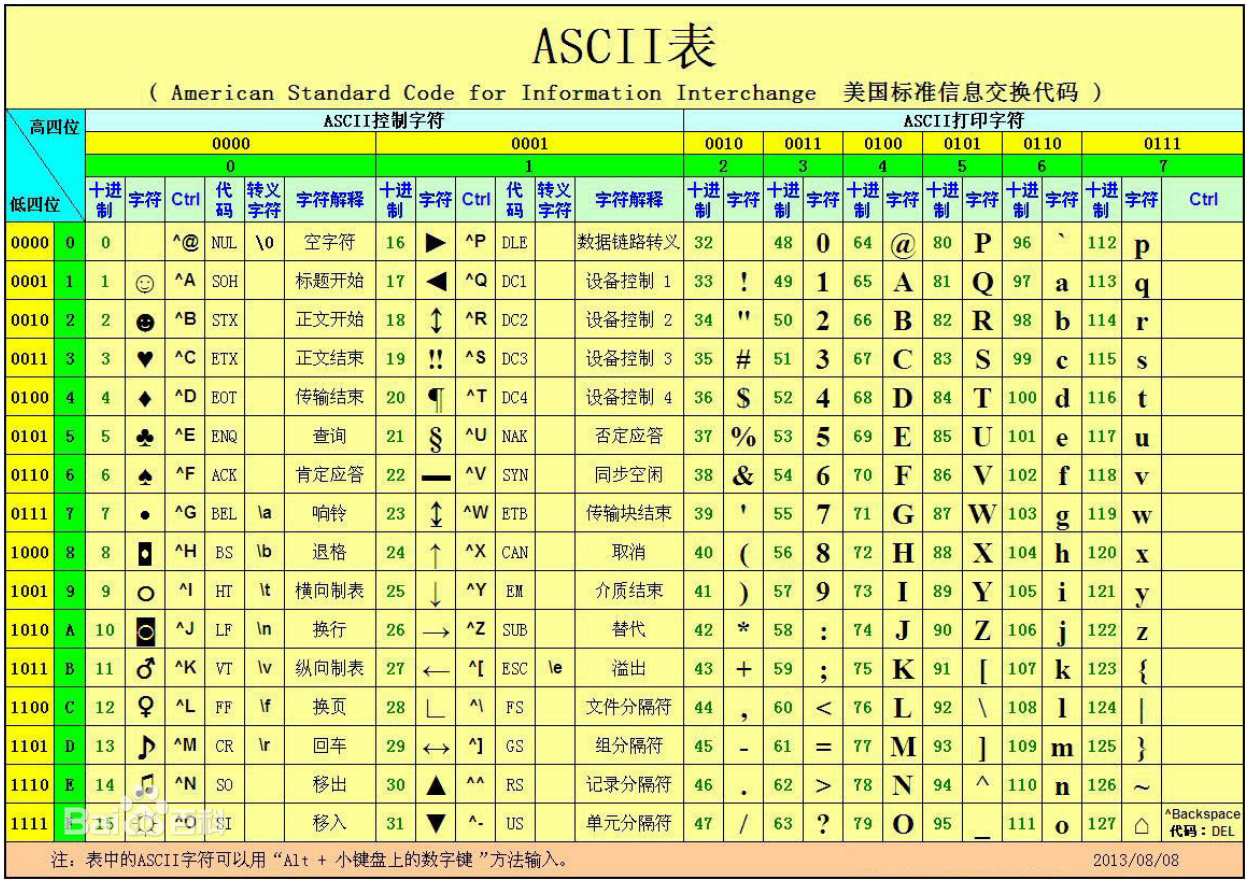
Oct 6th, 2019

Language Representation

- Basic task in NLP
- To represent human language in computer language/coding
  - Input: character/ word/ document/ sentence
  - Output: code/feature

Language Representation

- Character encoding



ASCII表

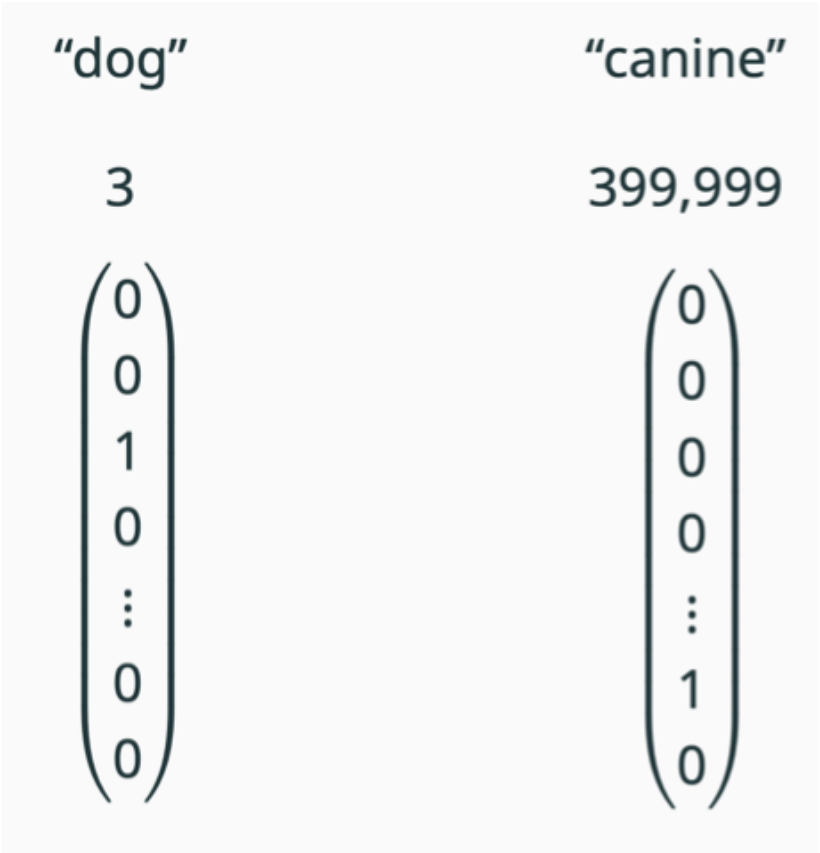
(American Standard Code for Information Interchange 美国标准信息交换代码)

高四位 / 低四位		ASCII控制字符 0000 0						ASCII控制字符 0001 1						ASCII打印字符 0010 2		ASCII打印字符 0011 3		ASCII打印字符 0100 4		ASCII打印字符 0101 5		ASCII打印字符 0110 6		ASCII打印字符 0111 7		
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0	空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	^Backspace 代码：DEL

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。 2013/08/08

Language Representation

- Word representation
  - One-hot representation
  - Problem
    - Sparsity
    - Semantic
    - Frequency (document)



Language Representation

- Document: Bag of words

John likes to watch movies. Mary likes too.

John also likes to watch football games.

  - Dictionary

{"John": 1, "likes": 2, "to": 3, "watch": 4, "movies": 5, "also": 6, "football": 7, "games": 8, "Mary": 9, "too": 10}

  - Document Representation

[1, 2, 1, 1, 1, 0, 0, 0, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

Language Representation

- Document: TF-IDF
  - Term Frequency

$$TF_{w,D_i} = \frac{count(w)}{|D_i|}$$

  - Inverse Document Frequency

$$IDF_w = \log \frac{N}{1 + \sum_{i=1}^{N} I(w, D_i)}$$

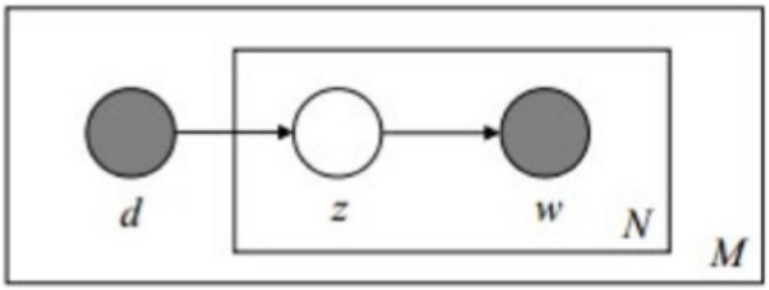
  - TF-IDF:

$$TF - IDF_{w,D_i} = TF_{w,D_i} * IDF_w$$

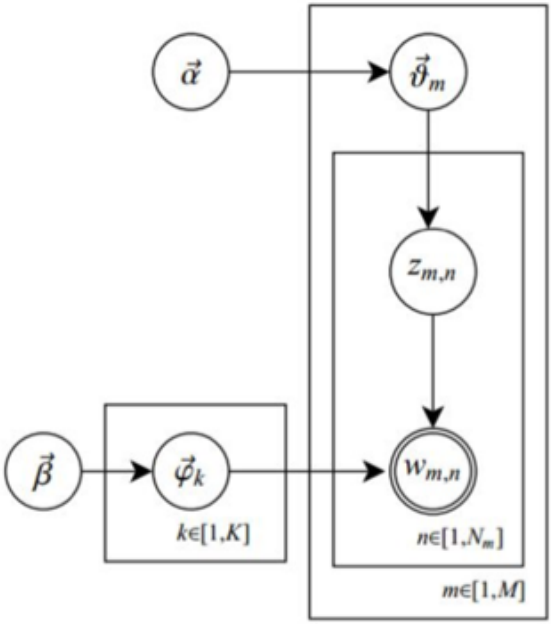
Language Representation

- Document: Topic Model
  - Document—topic / Topic—word
  - PLSA

- 1. 按照概率 $p(d_i)$ 选择一篇文档 d_i
- 2. 根据选择的文档 d_i，从从主题分布中按照概率 $p(\zeta_k \mid d_i)$ 选择一个隐含的主题类别 $\$\zeta_k$
- 3. 根据选择的主题 ζ_k，从词分布中按照概率 $p(\omega_j \mid \zeta_k)$ 选择一个词 ω_j



Language Representation



- Document: Topic Model
  - Document—topic / Topic—word
  - LDA
  - Latent Dirichlet allocation, JMLR, 2013

- 1. 按照先验概率 $p(d_i)$ 选择一篇文档 d_i
- 2. 从Dirichlet分布 α 中取样生成文档 d_i 的主题分布 θ_i，主题分布 θ_i 由超参数为 α 的Dirichlet分布生成
- 3. 从主题的多项式分布 θ_i 中取样生成文档 d_i 第 j 个词的主题 $z_{i,j}$
- 4. 从Dirichlet分布 β 中取样生成主题 $z_{i,j}$ 对应的词语分布 $\phi_{z_{i,j}}$，词语分布 $\phi_{z_{i,j}}$ 由参数为 β 的Dirichlet分布生成
- 5. 从词语的多项式分布 $\phi_{z_{i,j}}$ 中采样最终生成词语 $\omega_{i,j}$

- Essence: word co-occurrence (semantic ?)

Word Embedding

- Language Modeling
  - a probability distribution over word sequences

S1: 语言模型的本质是对一段自然语言的文本进行预测概率的大小
S2: 语言模型的本质是对自然一段语言的文本进行预测概率的大小
S3: 语言模型的本质是对自然语言一段的文本进行预测概率的大小

  - Which one is more like a sentence?

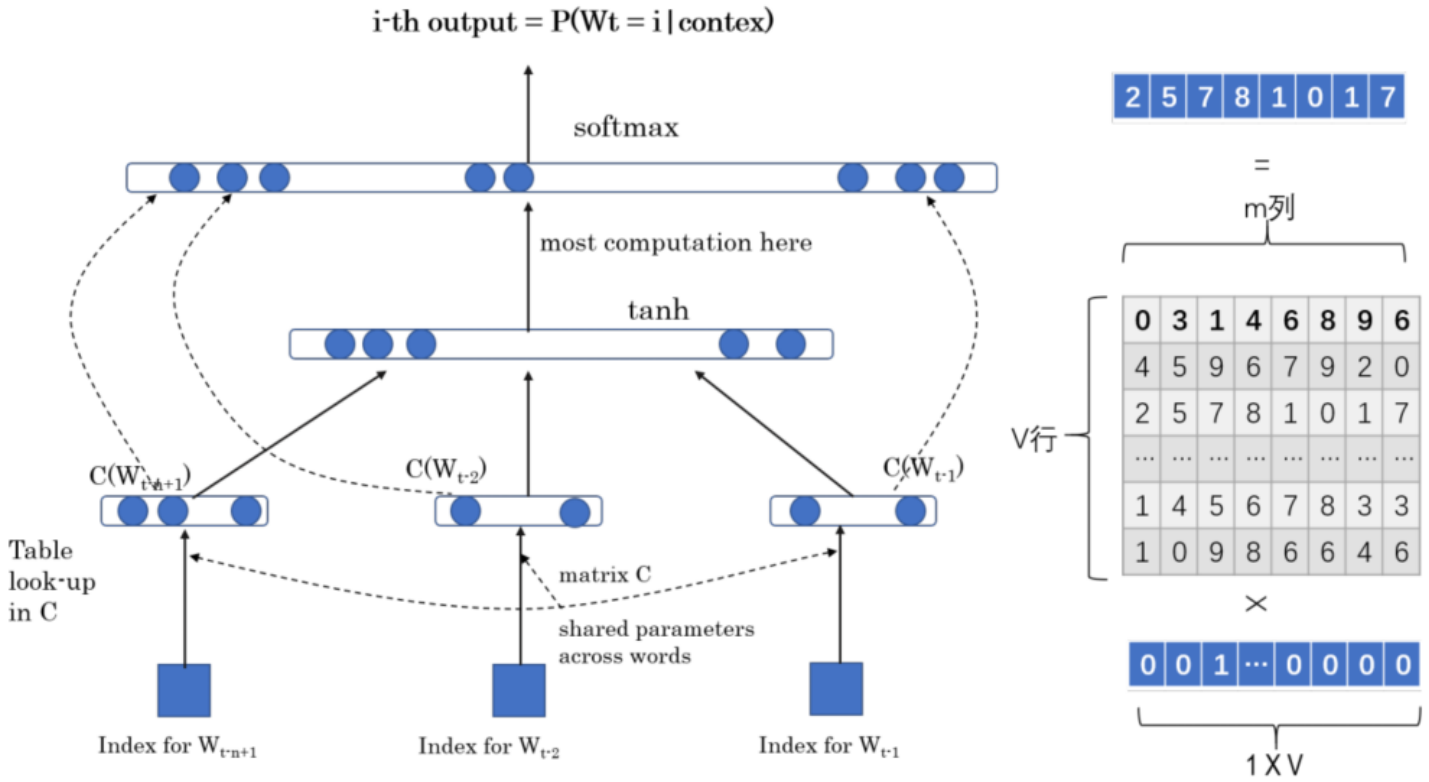
$$P(w_1, w_2, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_n|w_1, \dots, w_{n-1})$$

$$L = \sum_{w \in C} \log P(w|context(w))$$

  - N-gram model

Word Embedding

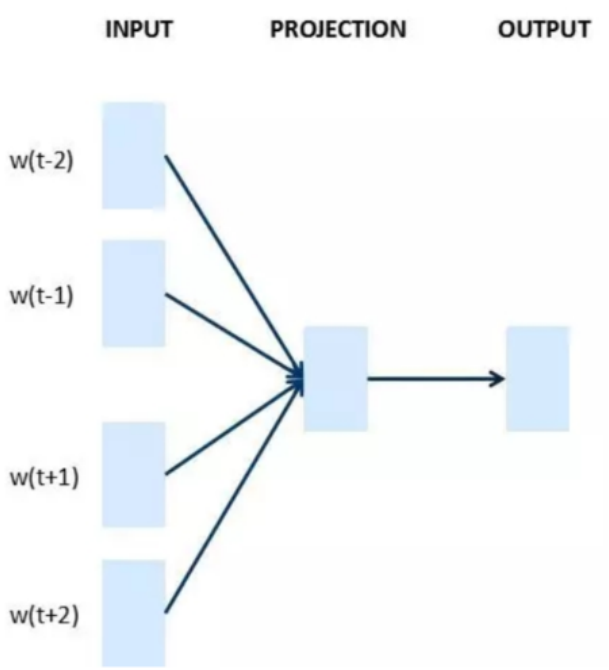
- Neural Network Language Model (NNLM) (2003)



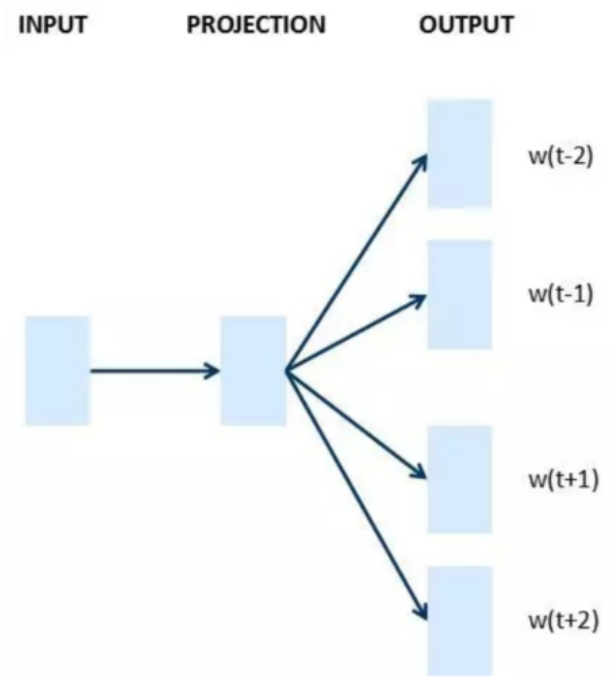
- Given word from 1 to t-1, predict word t

Word2Vec

- Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. ICLR Workshop (2013).
  - Distributional Hypothesis
    - a word is characterized by the context it keeps



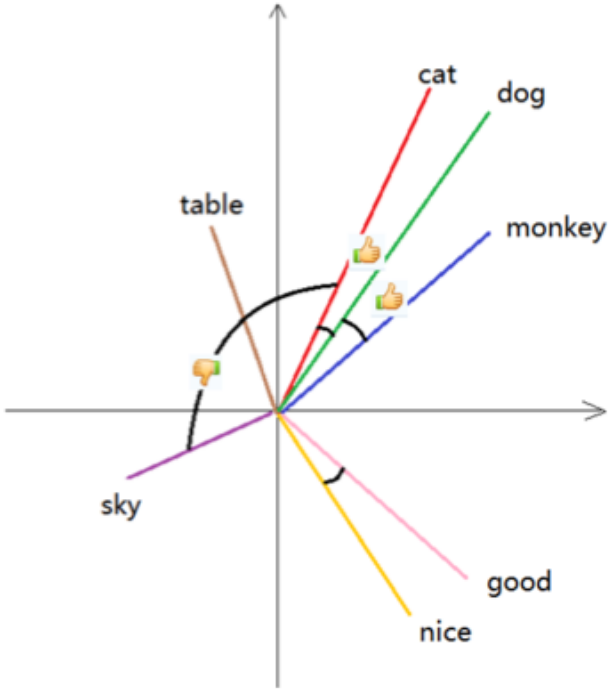
CBOW (Continuous Bag-of-Words Model)



Skip-gram (Continuous Skip-gram Model)

Word Embedding

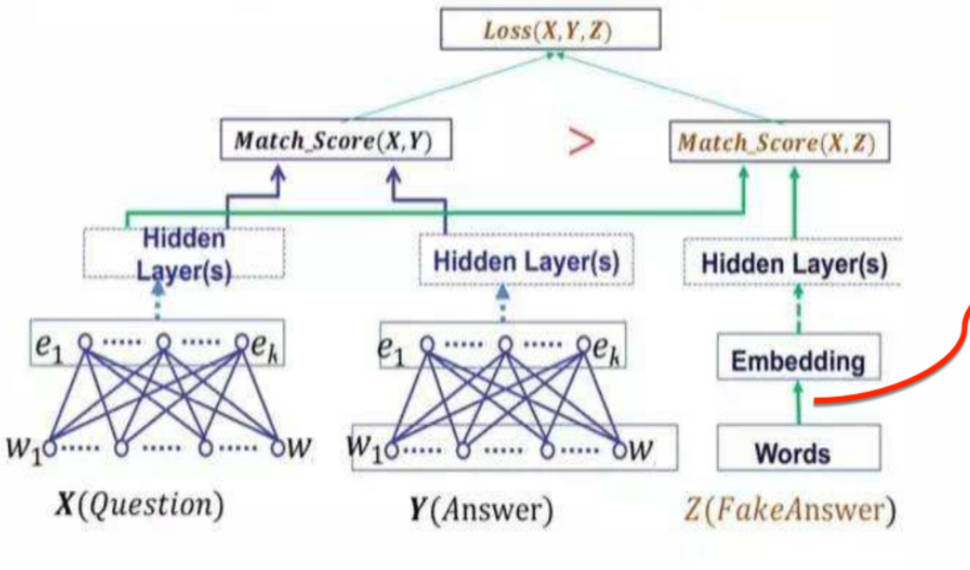
- Advantage： semantic
  - Word similarity



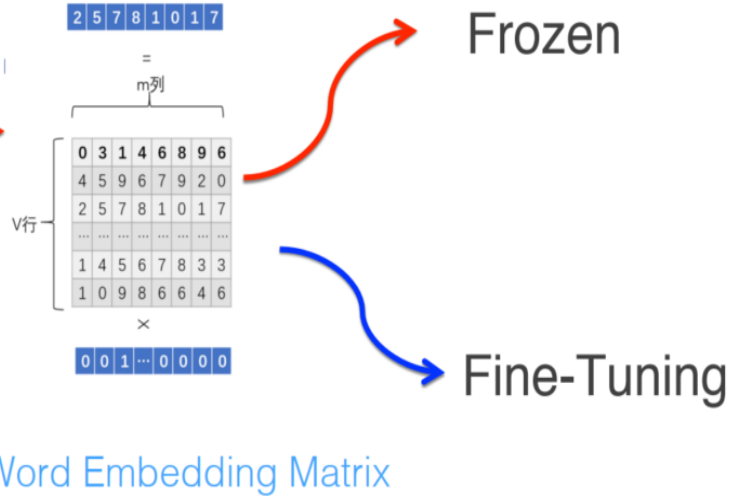
  - Word Analogy
    - King – Queen ~= Man – Woman
    - China – Beijing ~= UK – London ~= Capital

Word Embedding

- How to use



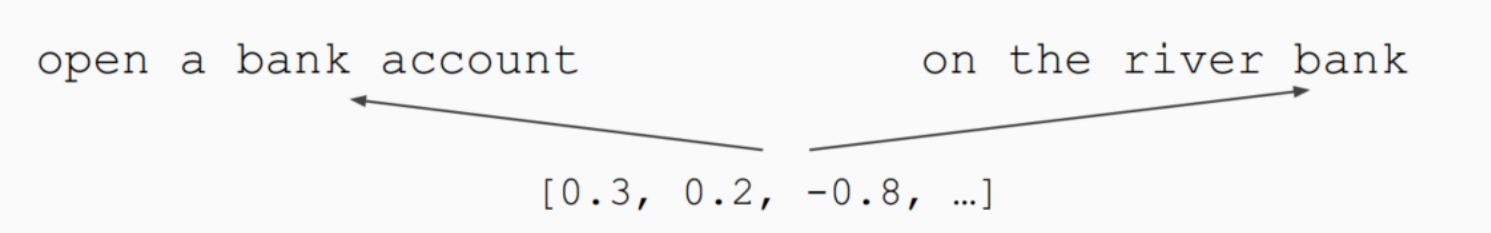
QA



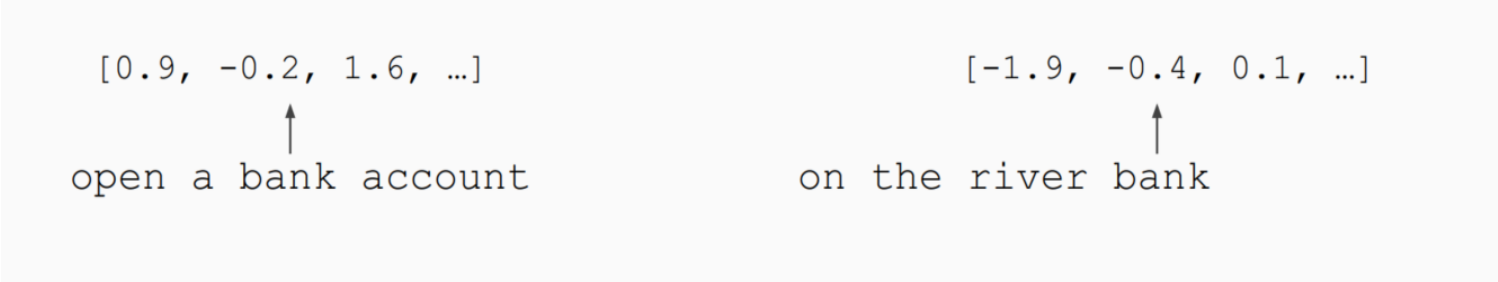
Word Embedding Matrix

Word Embedding

- Problem: 多义词
  - Static
  - No context



- Context-aware

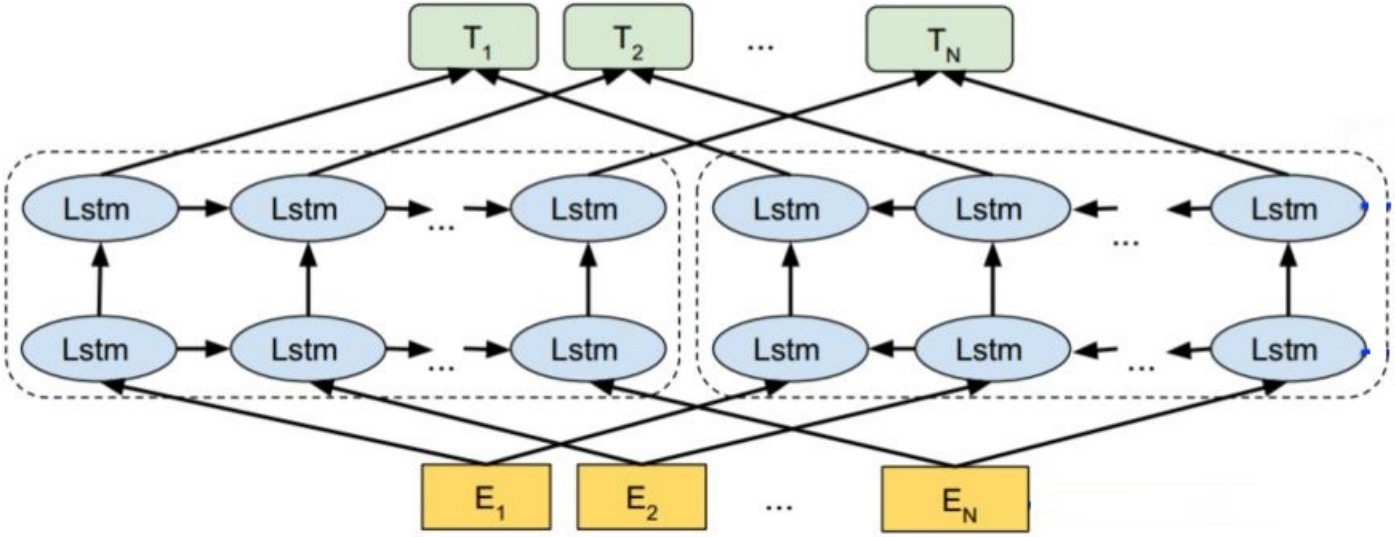


Pre-training

- ELMo
  - Deep contextualized word representation (NACCL 18 Best)
- ULMFit
  - Universal Language Model Fine-tuning model (ACL 2018)
- GPT—open AI
  - Improving Language Understanding by Generative Pre-Training
- BERT—google research
  - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (ACL 2019)

Pre-training

- ELMo: Feature-based pre-training
  - Step 1: Pre-training ELMo on large corpus
    - Bi-LSTM

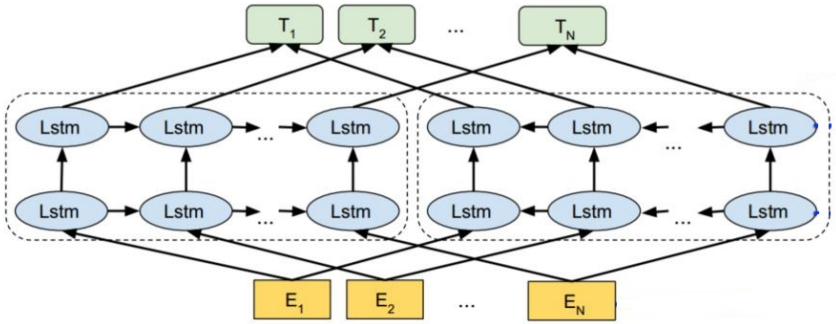


    - Language model for training objective

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

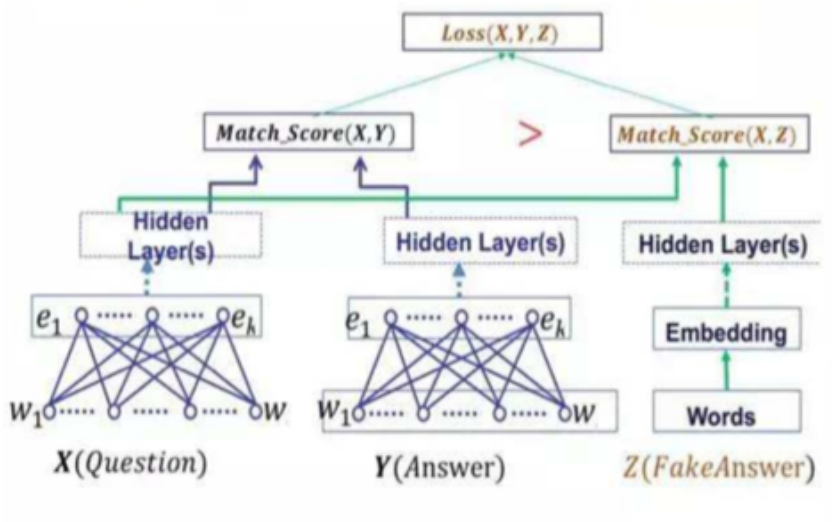
Pre-training

- ELMo: Feature-based pre-training
  - Step 2: Fine-tuning for downstream tasks
    - Extract the features of all layers in ELMo
    - Concatenation: Weighted-average in downstream tasks



$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.$$

pre-trained embeddings



QA

Pre-training

- ELMo: Experiments

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

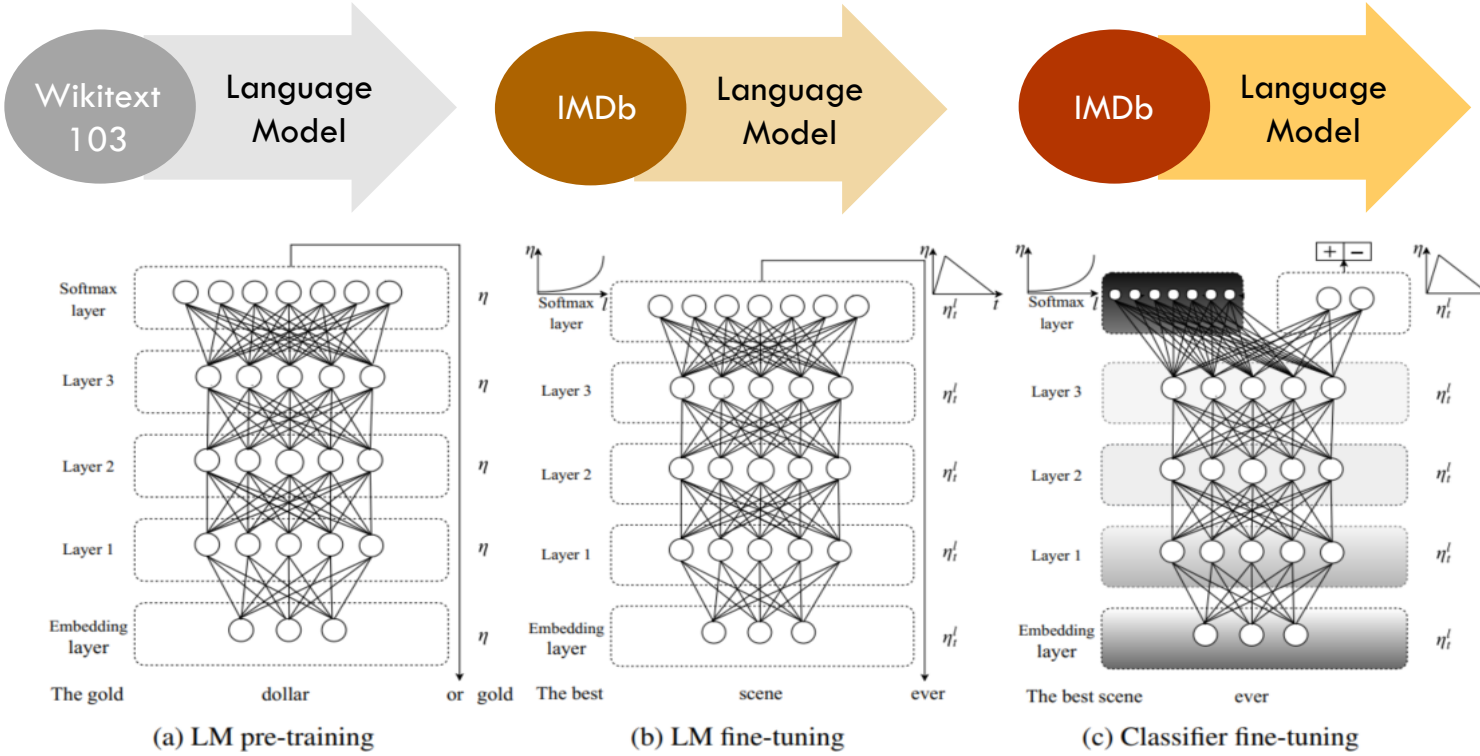
(多义词)Play:
1.运动
2.音乐

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

Pre-training

- ULMFit
  - Step 1: LM pre-training on large corpus (wiki 103)
  - Step 2: LM fine-tuning on specific corpus (LM funing)
  - Step 3: fine-tuning with tasks and models (Classification)

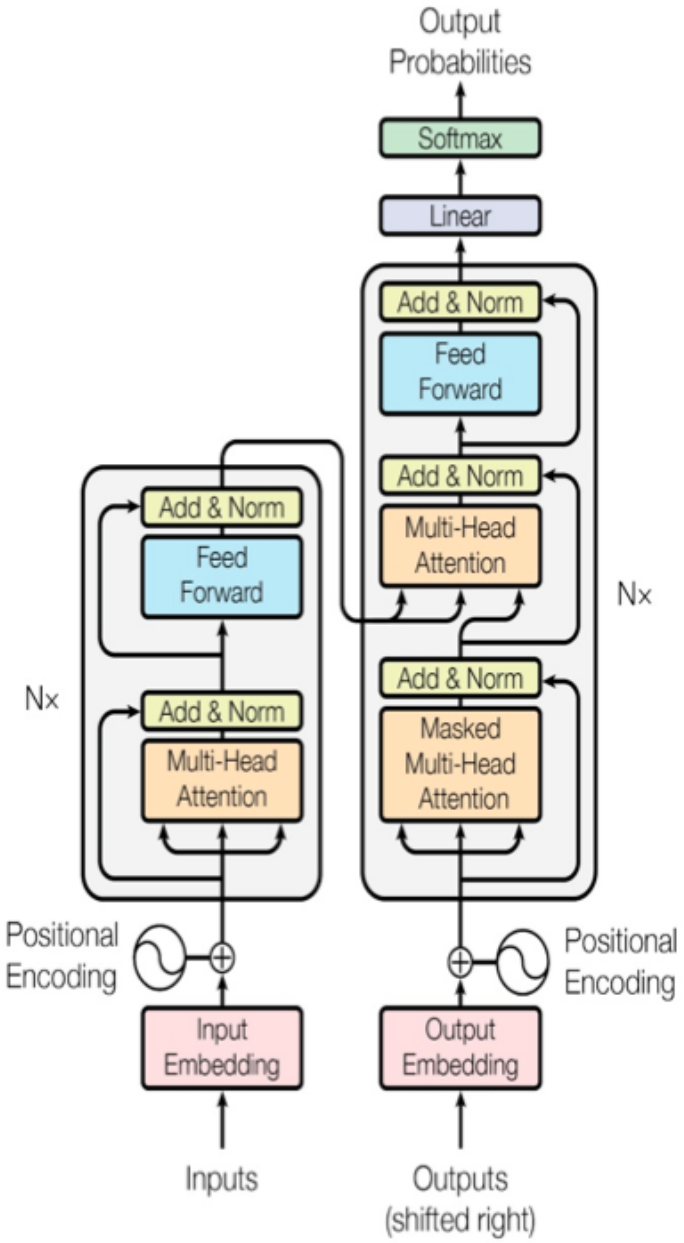


Pre-training

- Problem
  - LSTM is not good at extracting features
  - Long-term dependency
  - LM: Parallelization
  - Concatenation vs. real Context ?

Transformer

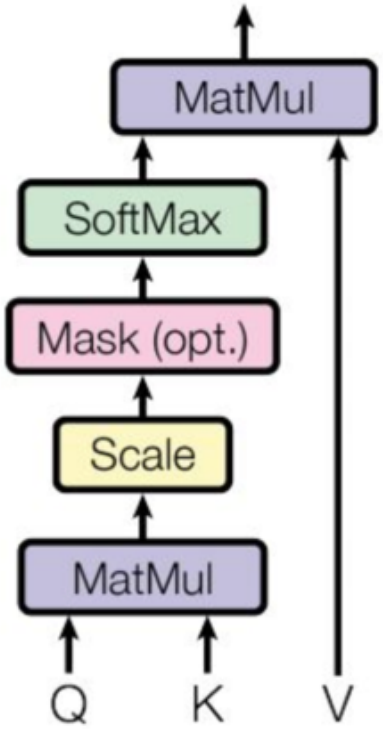
- Transformer
  - Self-attention
  - No sequence input
  - Attention is all you need, NIPS, 2017



Transformer

- Self-attention
  - All context

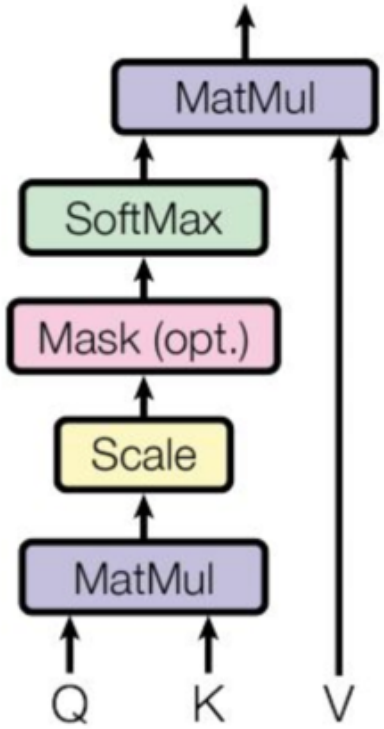
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

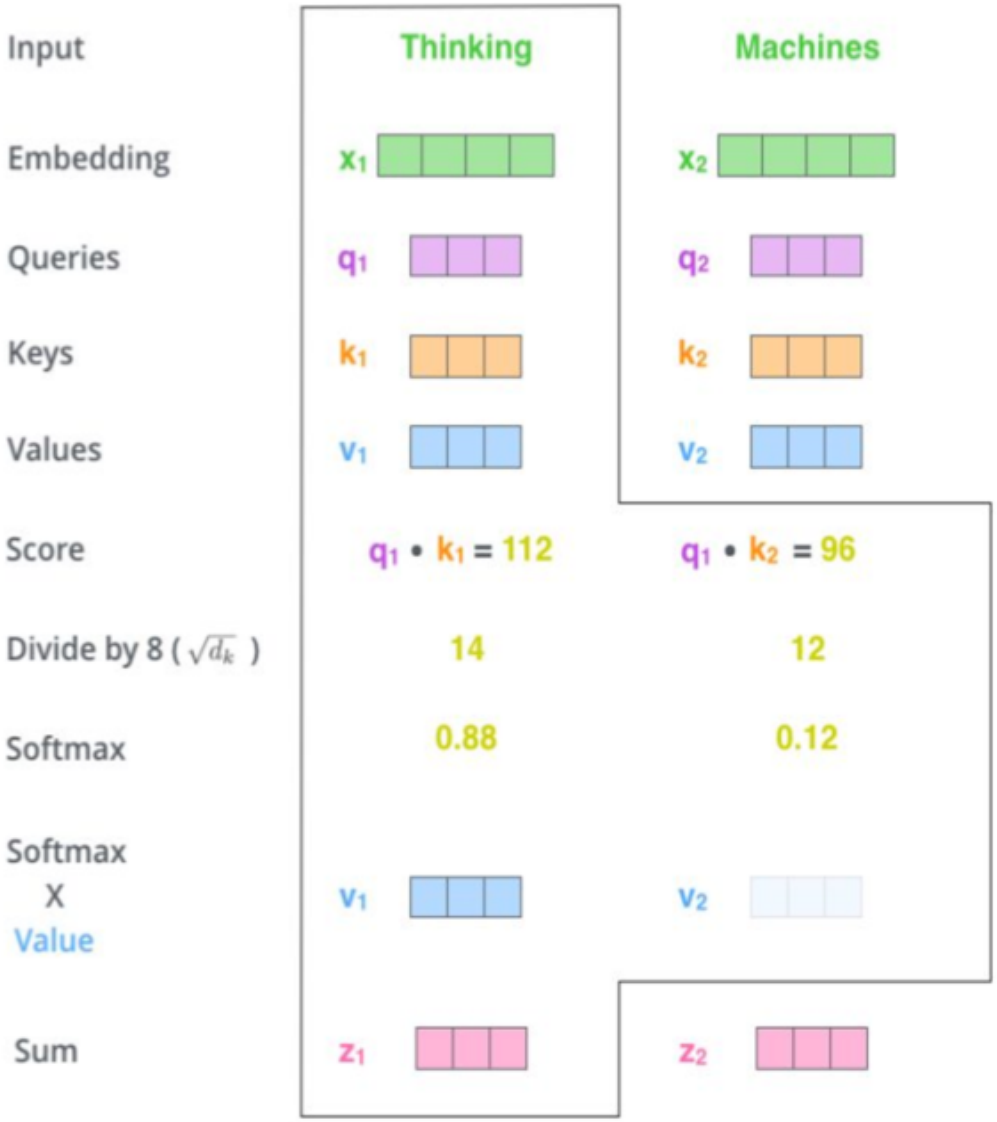


Transformer

- Self-attention
  - All context

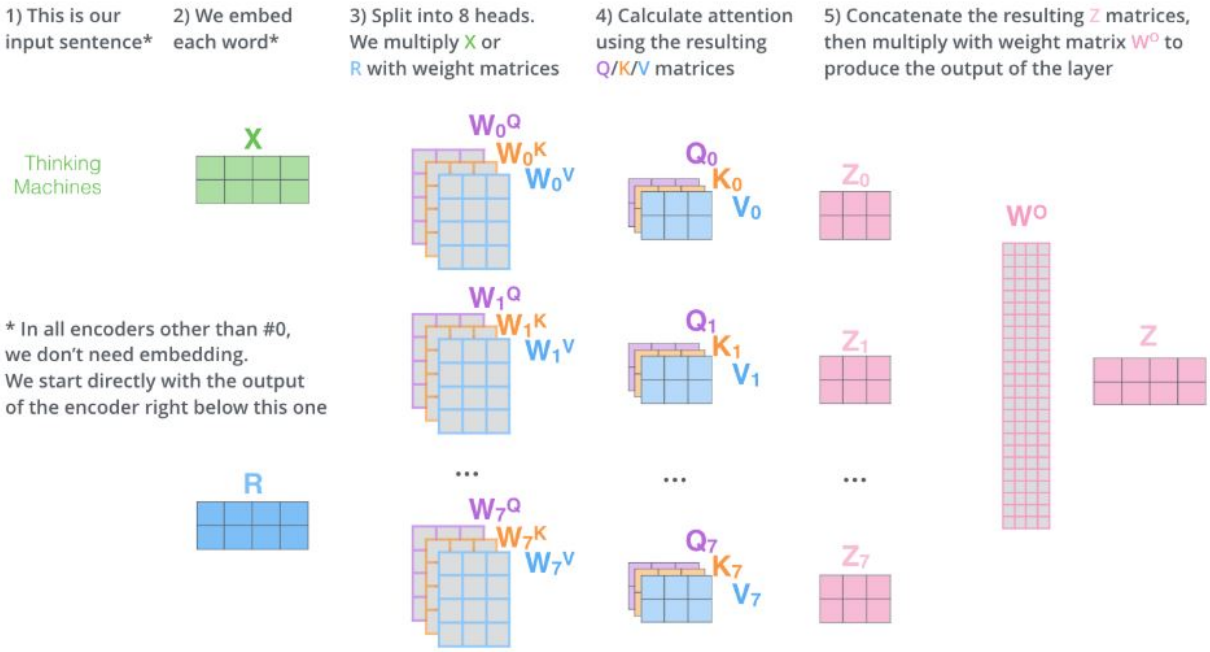
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$





Transformer

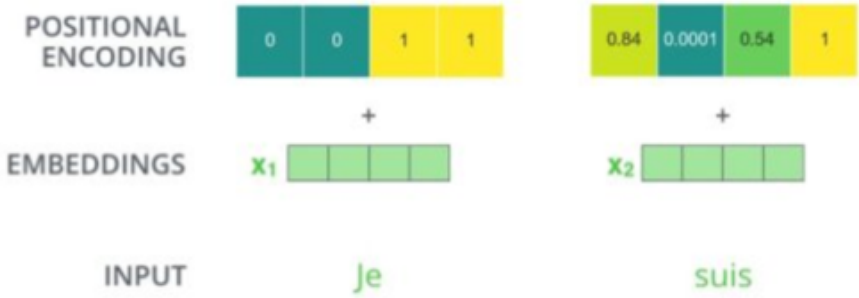
- Multi-head Attention



- Position encoding

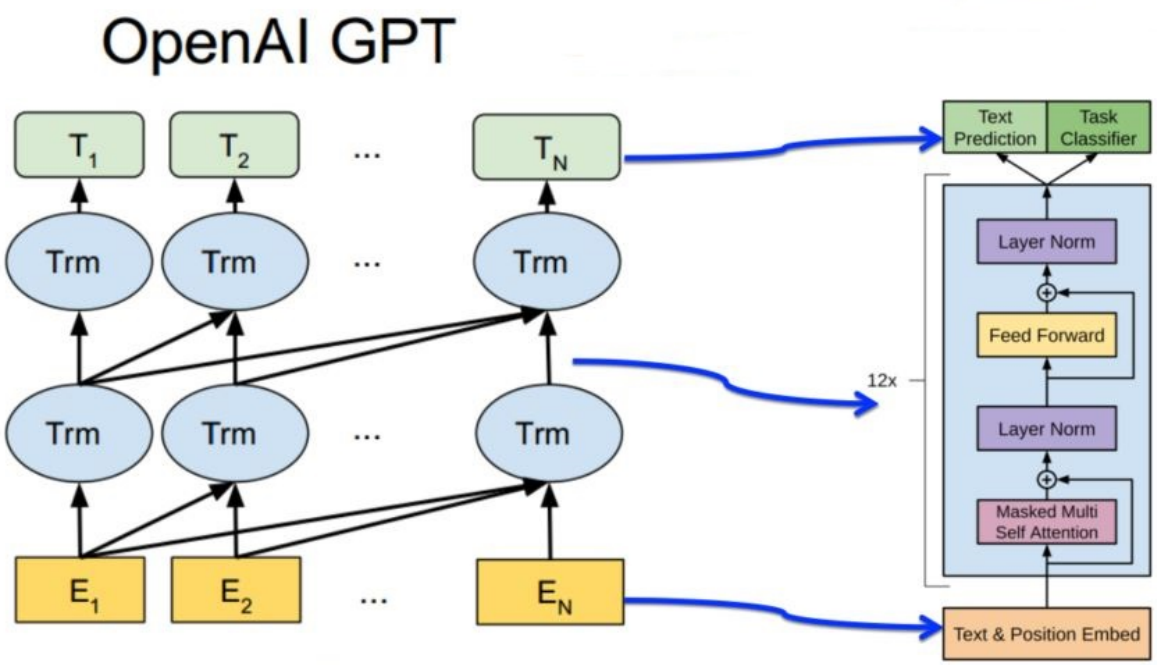
$$PE(pos, 2i) = \sin\left(\text{pos} / 10000^{2i} / d_m \text{ odel}\right)$$

$$PE(pos, 2i+1) = \cos\left(pos / 10000^{2i} / d_m \text{odel}\right)$$



Pre-training

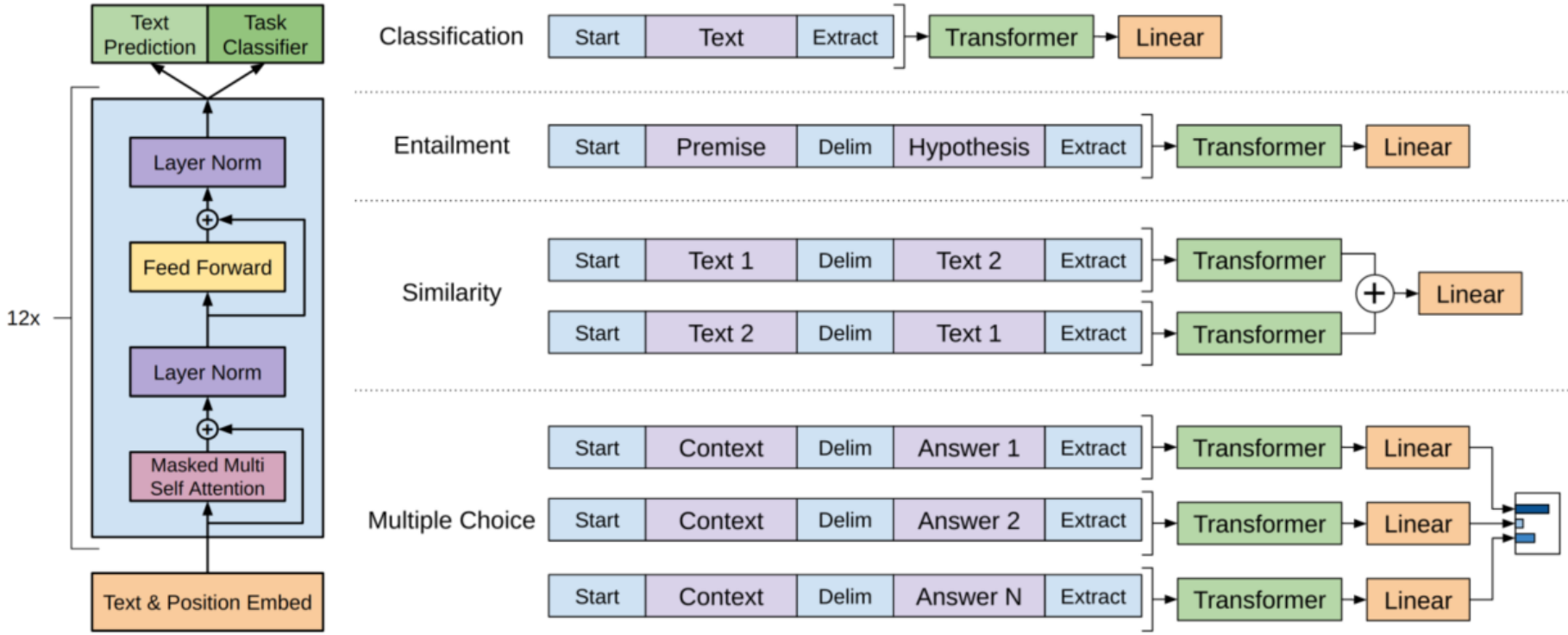
- GPT: Model-based pre-training
  - Step 1: pre-training
    - Single direction
    - Transformer



预训练

Pre-training

- GPT: Model-based pre-training
  - Step 2: fine-tuning
    - Reconstruct GPT with downstream tasks
    - Downstream models must adapt to Transformer



Pre-training

- GPT: Experiments

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

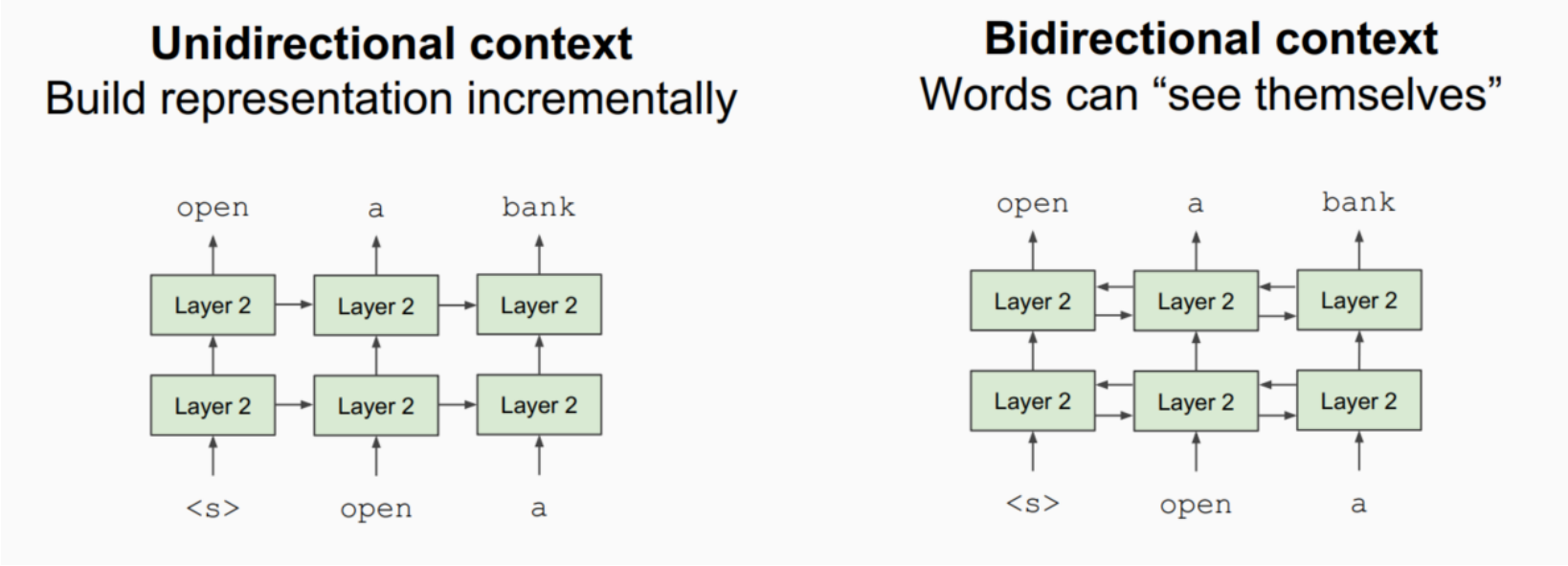
Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	**Classification**		**Semantic Similarity**			**GLUE**
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

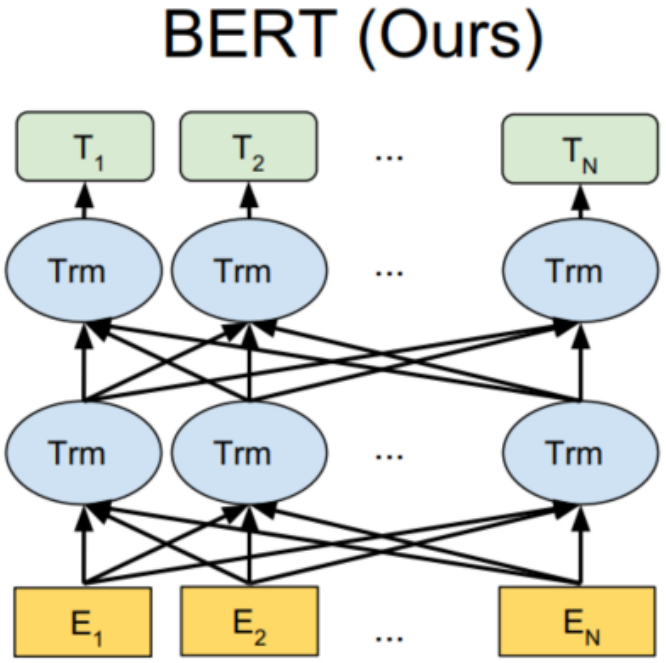
- 12 NLP tasks : 9 best !

Pre-training

- GPT: Problem
  - Unidirectional vs. bidirectional
  - 宣传不到位



Pre-training



- BERT: Model-based
  - Step 1: pre-training
  - Bidirectional transformer
  - Objective: multi-task
    - Mask LM (word-level) (15%)
    - Problem: Seen
      - 80% [mask] `went to the store → went to the [MASK]`
      - 10% original `went to the store → went to the store`
      - 10% random `went to the store → went to the running`
    - Next sentence prediction

Input = `[CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]`

Label = `IsNext`

Input = `[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]`

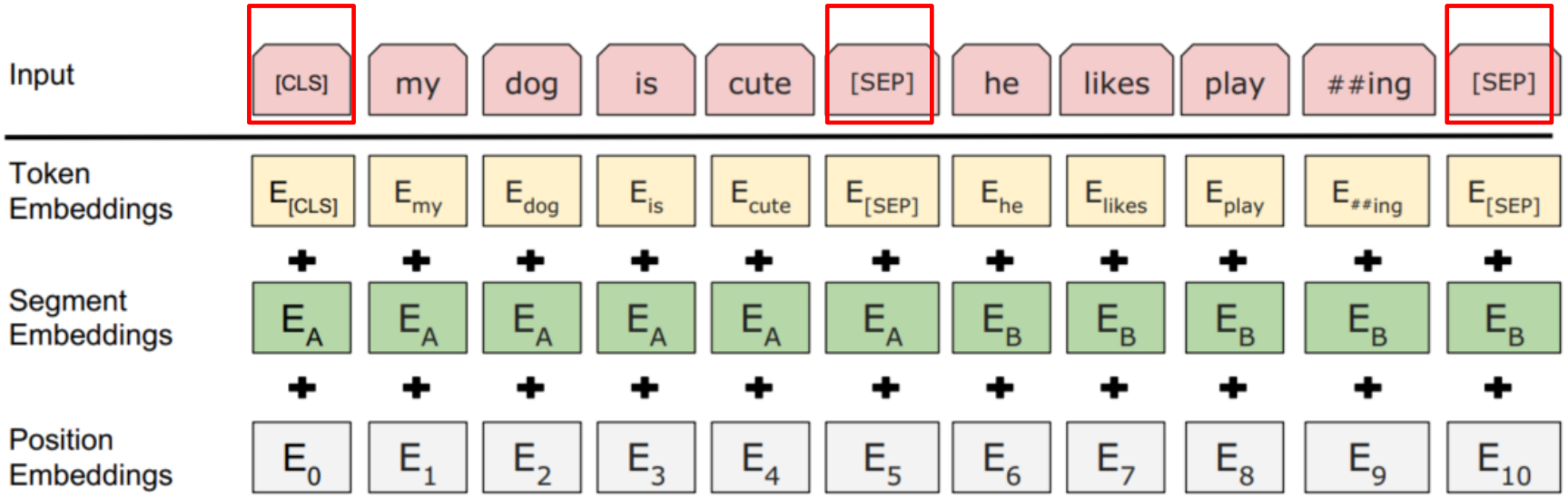
Label = `NotNext`

Pre-training

- BERT
  - Step 1: pre-training
    - Model Details
    - data : Wikipedia (2.5B words) + BookCorpus (800M words)
    - Batch size : 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
    - Training : 1M steps (~40 epochs)
    - Optimization : Adam, 1e-4 learning rate, linear decay
    - BERT-base: 12-layer, 768-hidden, 12-head
    - BERT-large: 24-layer, 1024-hidden, 16-head
    - Time : 4*4 or 8*8 TPU slice for 4 days

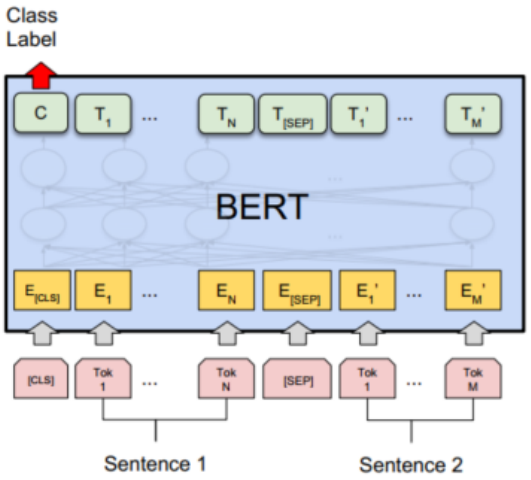
Pre-training

- BERT
  - Step 2: fine-tuning
  - Similar to GPT
    - On the top: add downstream tasks' objective

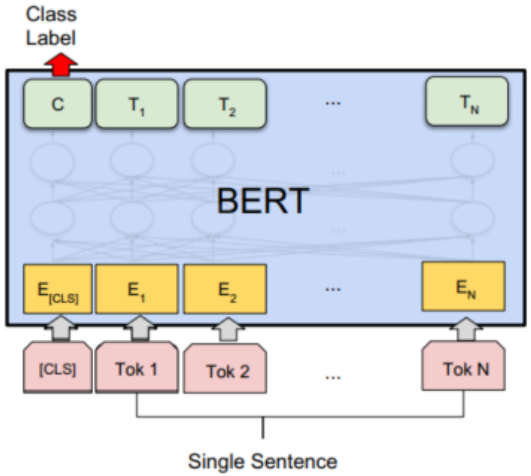


Pre-training

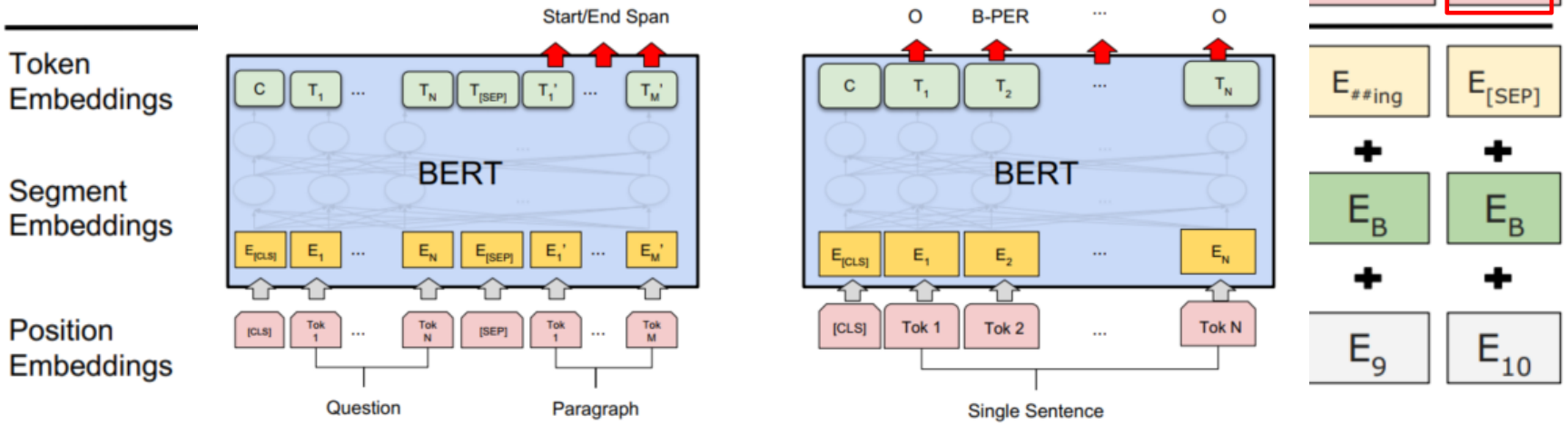
- BERT
  - Step
  - Sim
    - O



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Pre-training

- BERT: Experiments

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

System	Dev F1	Test F1
ELMo+BiLSTM+CRF	95.7	92.2
CVT+Multi (Clark et al., 2018)	-	92.6
$\text{BERT}_{\text{BASE}}$	96.4	92.4
$\text{BERT}_{\text{LARGE}}$	**96.6**	**92.8**

Table 3: CoNLL-2003 Named Entity Recognition results. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. Test results were scored against the hidden labels by the SWAG authors. †Human performance is measure with 100 samples, as reported in the SWAG paper.

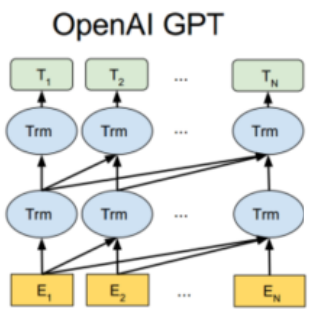
System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

11 NLP tasks：all best

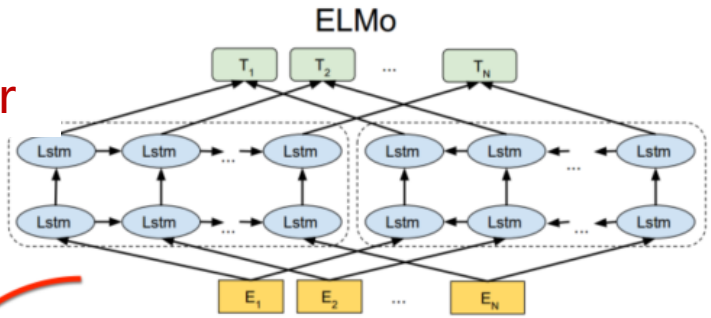
Pre-training

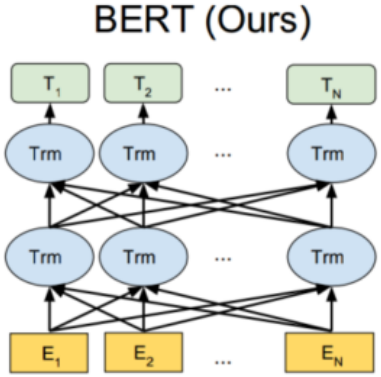
- BERT: novel?

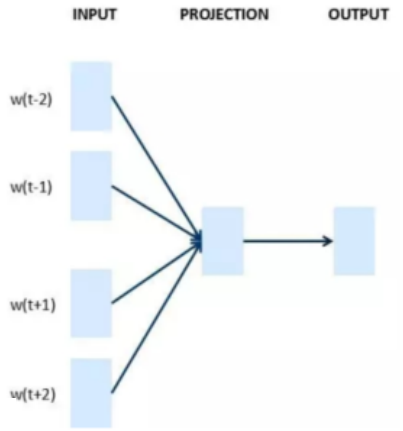


Unidirectional —> Bidirectional

LSTM —> Transformer







Mask language model

Next sentence prediction

Word2Vec:CBOW

After BERT

- **MT-DNN**: Multi-Task Deep Neural Networks for Natural Language Understanding (ACL 2019)
  - Multi-task learning
- **XLNet**: Generalized Autoregressive Pretraining for Language Understanding (preprint)
  - permutations with factorization order
- **ALBERT**: A Lite BERT for Self-supervised Learning of Language Representations.
  - Parameter factorization and sharing
- **ERNIE 2.0**: A Continual Pre-training Framework for Language Understanding
- **TinyBERT**: Distilling BERT for Natural Language Understanding
  - Distillation
- Probing Neural Network Comprehension of Natural Language Arguments, (BERT fails?:) Acl 2019
  - Overfitting ?

Discussion

- Data
- Speed
- Model
- Adversarial
- Interpretability
- Transfer
- Knowledge distillation
- Federated Learning

Thank you for listening!