CCKS 2020

Pushing the Frontier of Research on Language Models

Hang Li

Bytedance Technology

Language Model

- Probability distribution over word sequences
- Language probability calculation and language generation
- Language models can be specified by neural networks
- Pre-trained language models are state-of-the-art technologies

Talk Outline

- *Past*
  - *n-gram Language Model*
- Present
  - Neural Language Model
  - Pretrained Language Model
- Our Work
  - Soft Masked BERT
  - AMBERT
- Future
  - Brain-Inspired Language Model

Markov and Language Model

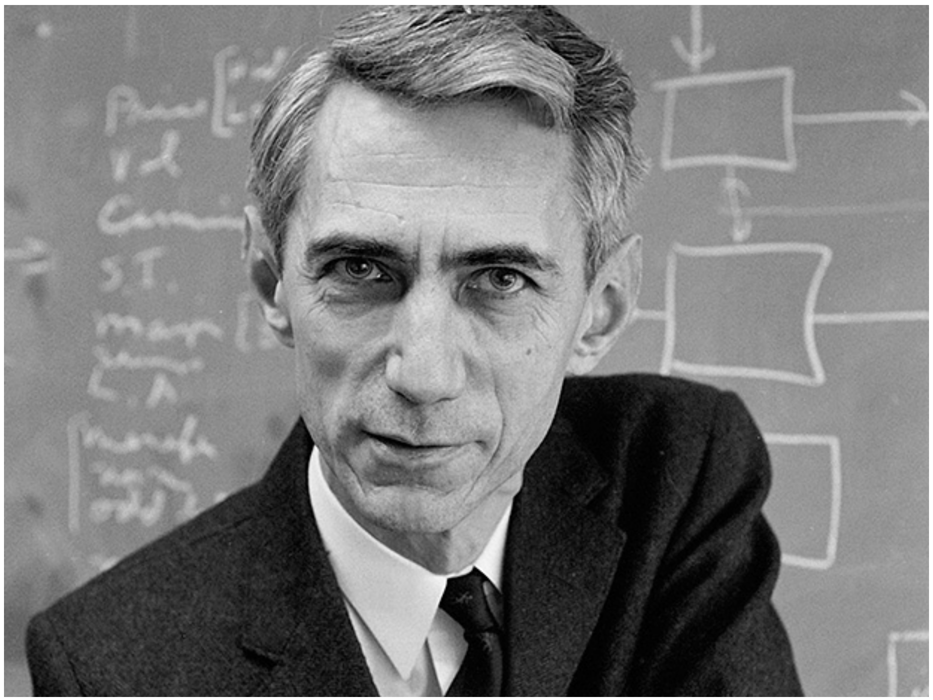
Andrey Markov

- In 1906, gave definition of Markov chain
- Simple case: only two states, proved ergodic theorem
- Later, expanded to more general cases
- In 1913, applied to Pushkin's Eugene Onegin

n-gram Language Model

- Language model is probability distribution
- To determine probability of word sequence $w_1, w_2, \cdots, w_N$
- $p(w_1, w_2, \cdots, w_N) = \prod_{i=1}^{N} p(w_i | w_1, w_2, \cdots, w_{i-1})$
- n-gram language model
- $p(w_1, w_2, \cdots, w_N) \approx \prod_{i=1}^{N} p(w_i | w_{i-n+1}, w_{i-n+2}, \cdots, w_{i-1})$
- $n - 1$ order Markov chain

Shannon and Language Model



Claude Shannon

- In 1948, laid down foundation of information theory
- Studied n-gram model
- Defined entropy and cross entropy of language

Entropy and Cross Entropy

- Entropy and cross entropy of n-gram model
- $H_n(p) = \sum_{w_1, w_2, \cdots w_n} -p(w_1, w_2, \cdots w_n) \cdot \log_2 p(w_1, w_2, \cdots w_n)$
- $H_n(p, q) = \sum_{w_1, w_2, \cdots w_n} -p(w_1, w_2, \cdots w_n) \cdot \log_2 q(w_1, w_2, \cdots w_n)$
- $H_n(p) \leq H_n(p, q)$
- Entropy and cross entropy of language
- $H(p) = \lim_{n \to \infty} \frac{1}{n} H_n(p) = \lim_{n \to \infty} -\frac{1}{n} \log_2 p(w_1, w_2, \cdots w_n)$
- $H(p, q) = \lim_{n \to \infty} \frac{1}{n} H_n(p, q) = \lim_{n \to \infty} -\frac{1}{n} \log_2 q(w_1, w_2, \cdots w_n)$
- $H(p) \leq H(p, q)$

Chomsky and Language Model

Noam Chomsky

- In 1956, introduced Chomsky hierarchy
- Sentences are generated according to grammar
- Finite state grammar (including n-gram model) is not suitable for language generation

Fine State Grammar

- Finite state grammar (also n-gram model) is not suitable for language generation
- (i) If S1, then S2.
- (ii) Either S3, or S4.
- (iii) Either if S5, then S6, or if S7, then S8

Talk Outline

- Past
  - n-gram Language Model
- *Present*
  - *Neural Language Model*
  - *Pretrained Language Model*
- Our Work
  - Soft Masked BERT
  - AMBERT
- Future
  - Brain-Inspired Language Model

Neural Language Model

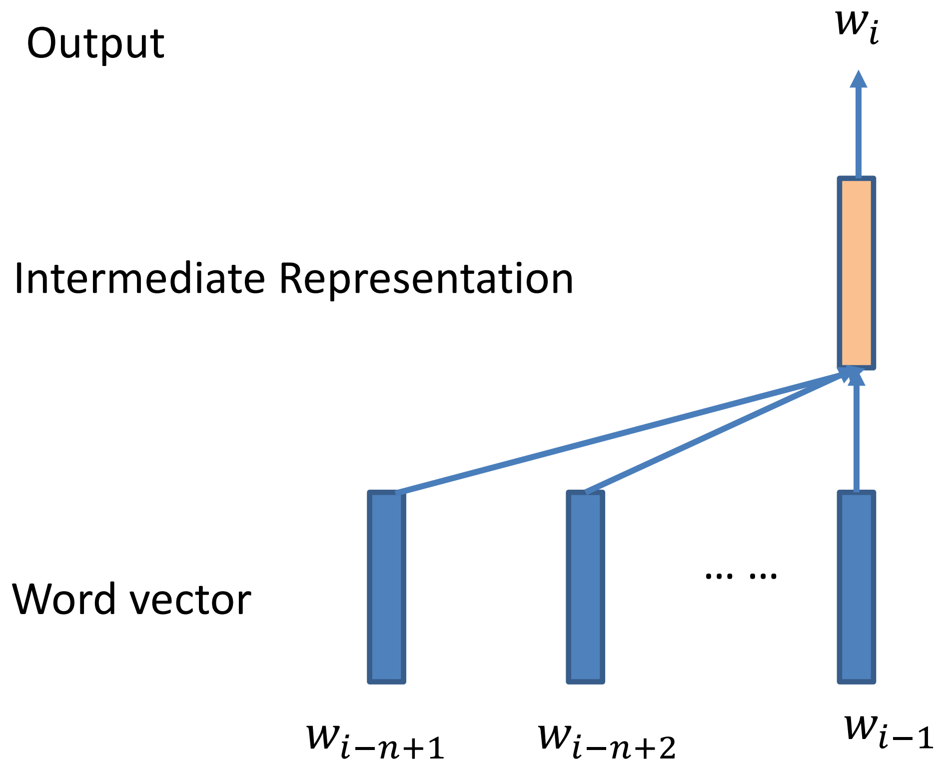
Yoshua Bengio

- n-gram model is difficult to learn when n is large
- In 2003, proposed first neural language model
- Language model parameterized by neural network
- Word is represented by word embedding (real-valued vector)

Neural Language Model

- Conditional probability is determined by neural network
- $p(w_i|w_{i-n+1}, w_{i-n+2}, \cdots, w_{i-1}) = f_\theta(\boldsymbol{w}_{i-n+1}, \boldsymbol{w}_{i-n+2}, \cdots, \boldsymbol{w}_{i-1})$
- Word embedding v.s. one-hot vector
- More compact, generalizable (similarity calculation), robust, extensible
- Neural network: learnable non-linear function

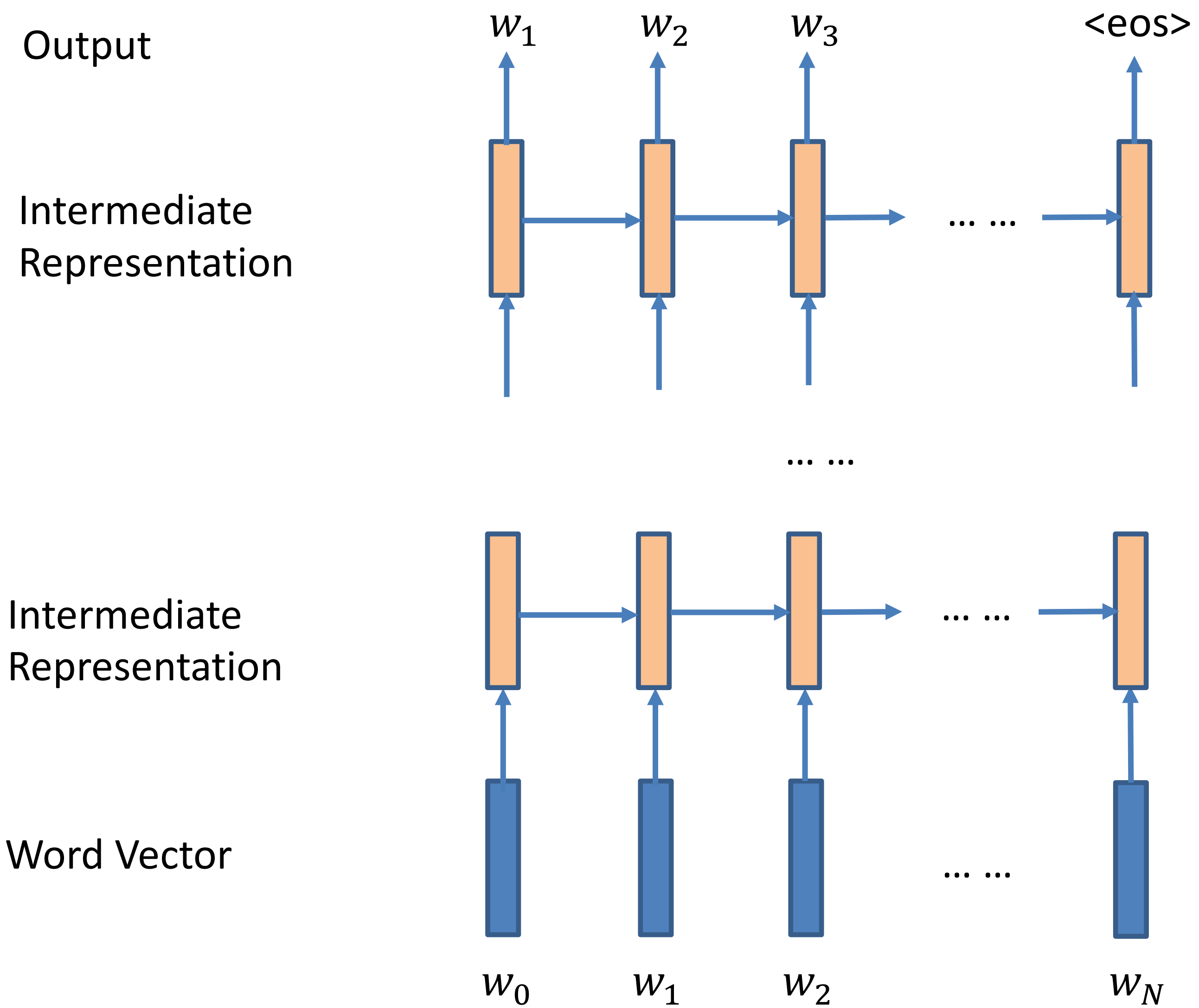
Representations in Bengio's Model



RNN Language Model

- Conditional probability is determined by RNN (Recurrent Neural Network)
- $p(w_i|w_1, w_2, \cdots, w_{i-1}) = f_\theta(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{i-1})$
- $\boldsymbol{h}_i = \tanh(\boldsymbol{U} \cdot \boldsymbol{h}_{i-1} + \boldsymbol{W} \cdot \boldsymbol{w}_i + \boldsymbol{b})$
- $p(w_i|w_1, w_2, \cdots, w_{i-1}) = \mathrm{softmax}(\boldsymbol{V} \cdot \boldsymbol{h}_i)$
- No Markov assumption

Representations in RNN Language Model



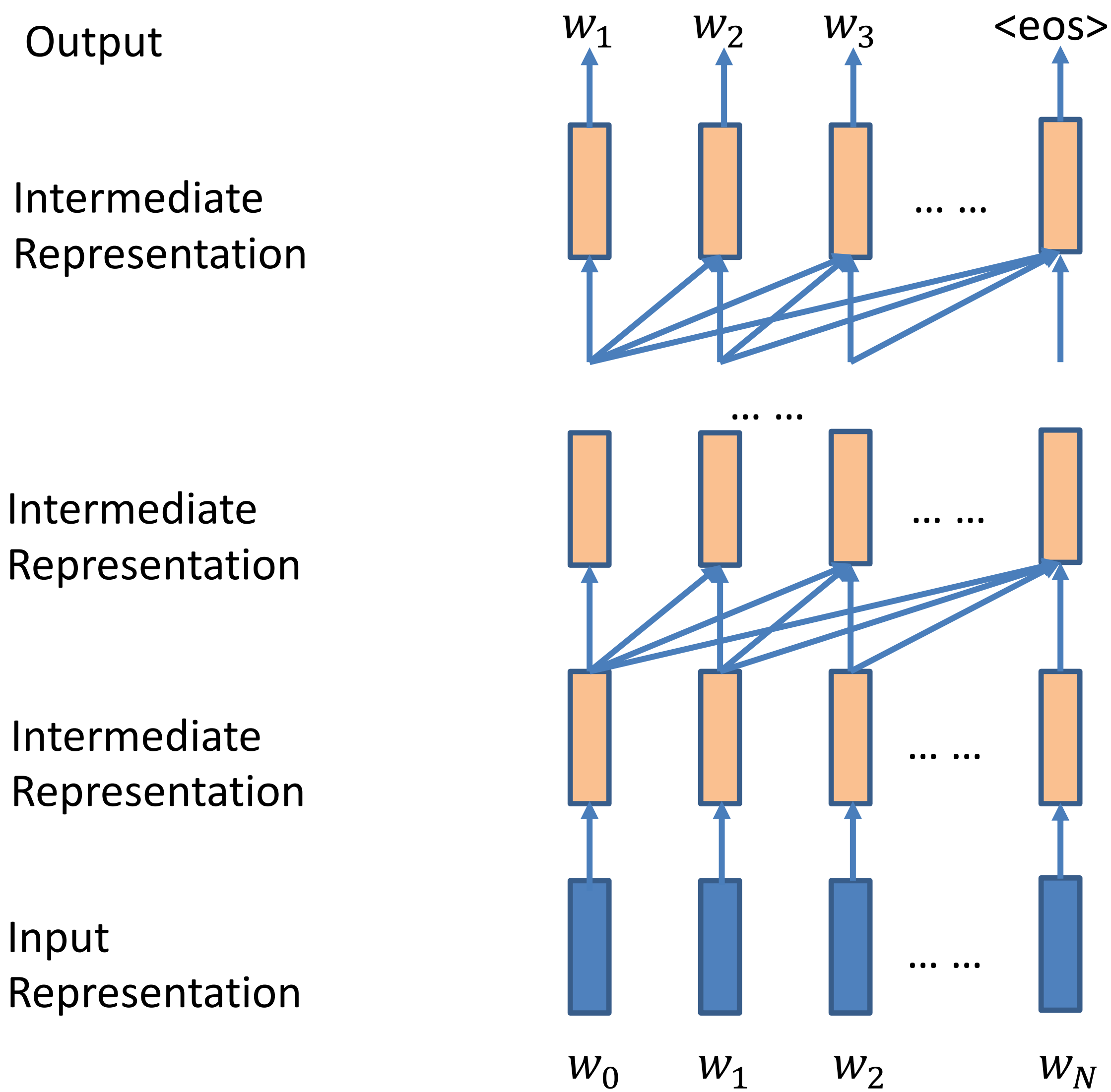
Pre-Trained Language Model

- Pre-training: learning of neural language model (Transformer) using large amount of data in unsupervised learning manner
- Fine-tuning: learning of neural language model for downstream task in supervised learning manner
- Key ingredients: big data, powerful representation, pre-training techniques

Pretrained Language Model: GPT

- Input: sequence of words
- Output: sequence of representations of words
- Model: Transformer decoder
- $\boldsymbol{H}^{(L)} = \text{transformer\_decoder}(\boldsymbol{H}^{(0)})$
- Unidirectional language model (auto regressive)
- Pre-training: maximum likelihood estimation of sequence (minimum cross entropy)
- $-\log p(\boldsymbol{w}) = -\sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(w_i | w_1, \cdots, w_{i-1})$

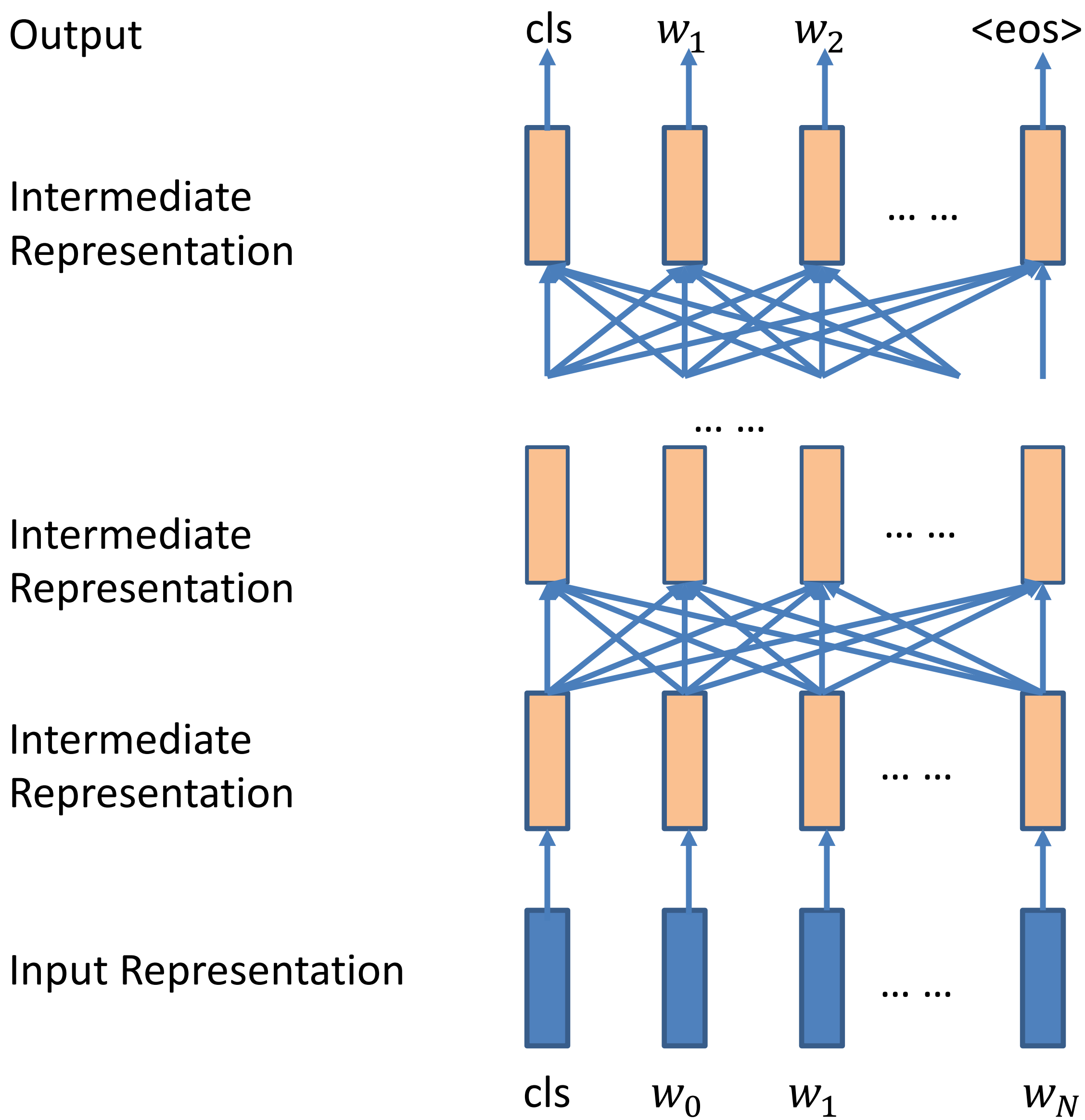
Representations in GPT



Pretrained Language Model: BERT

- Input: sequence of words
- Output: sequence of representations of words
- Model: Transformer encoder
- $\boldsymbol{H}^{(L)} = \text{transformer\_encoder}(\boldsymbol{H}^{(0)})$
- Bidirectional language model
- Pre-training: mask language model, sequence-to-sequence denoising
- $-\log p(\bar{\boldsymbol{w}}|\tilde{\boldsymbol{w}}) \approx -\sum_{i=1}^{N} \delta_i \log p_{\boldsymbol{\theta}}(w_i|\tilde{\boldsymbol{w}})$

Representations in BERT



Talk Outline

- Past
  - n-gram Language Model
- Present
  - Neural Language Model
  - Pretrained Language Model
- *Our Work*
  - *Soft Masked BERT*
  - *AMBERT*
- Future
  - Brain-Inspired Language Model

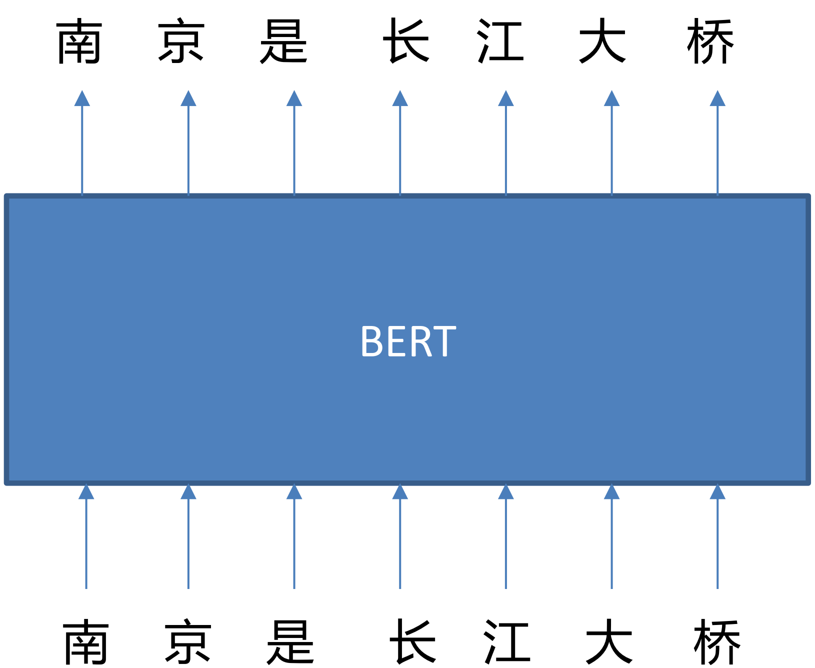
Soft-Masked BERT

- A model for spelling error correction
- Using BERT as correction network
- Using Bidirectional GRU as dection network
- Soft masking is performed on BERT
- State-of-the-art method for Chinese spelling error correction

Soft-Masked BERT

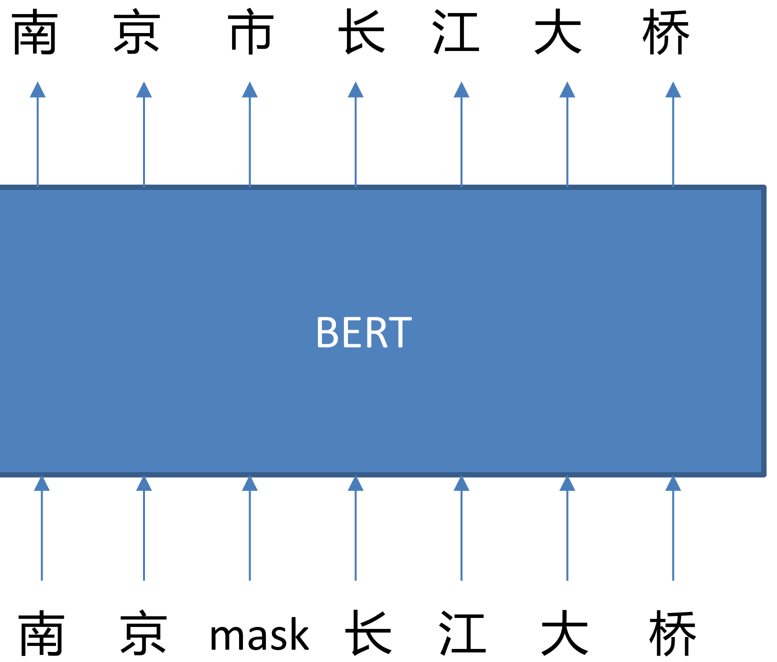
A Naïve Approach

- Directly using BERT
- Tends to choose not to make correction
- Due to way of pre-training, i.e., masking language modeing



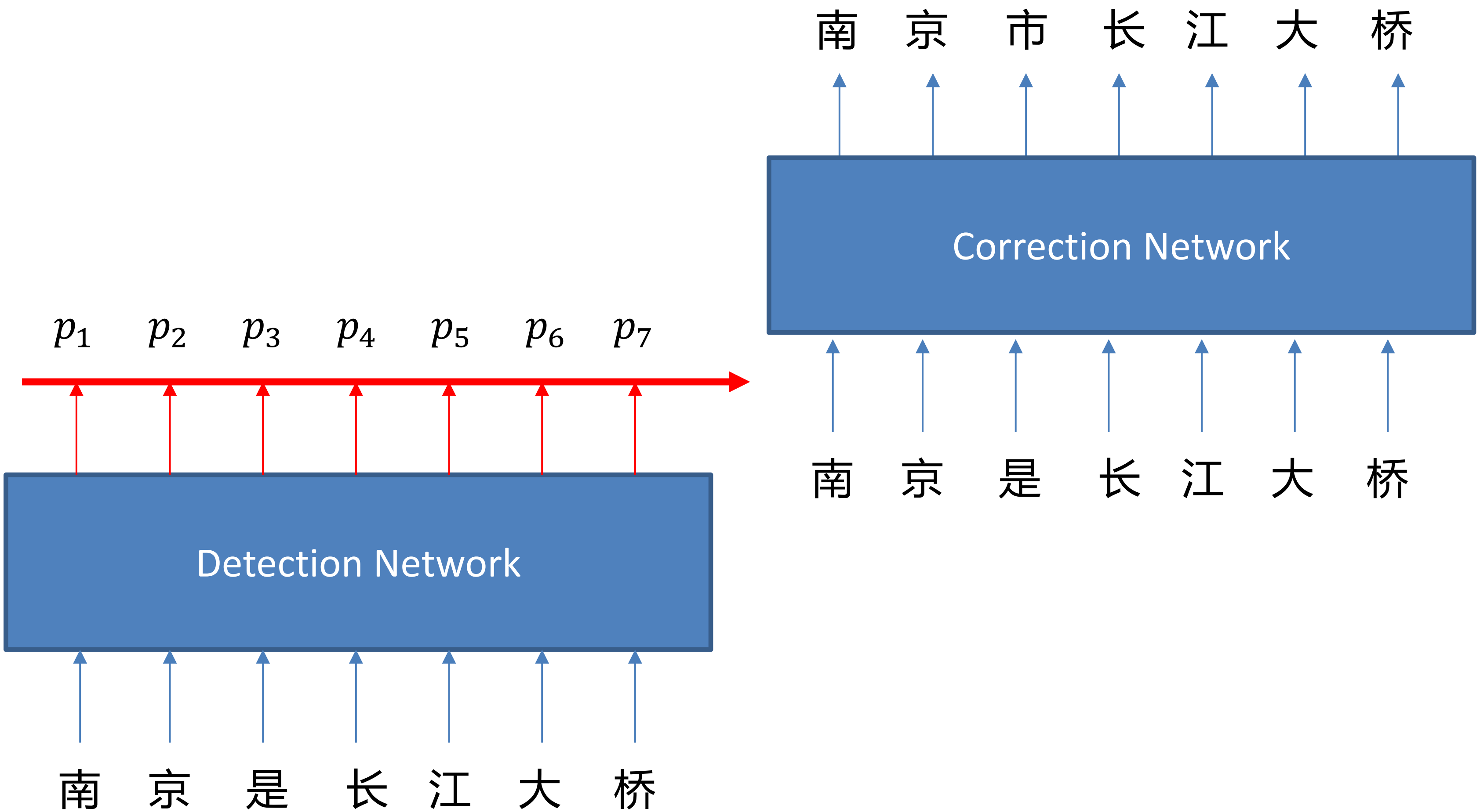
If Candidate is Masked

- Directly using BERT
- Mask incorrect word and make prediction
- Can work very well in error correction
- However, candidate for masking is not known in advance



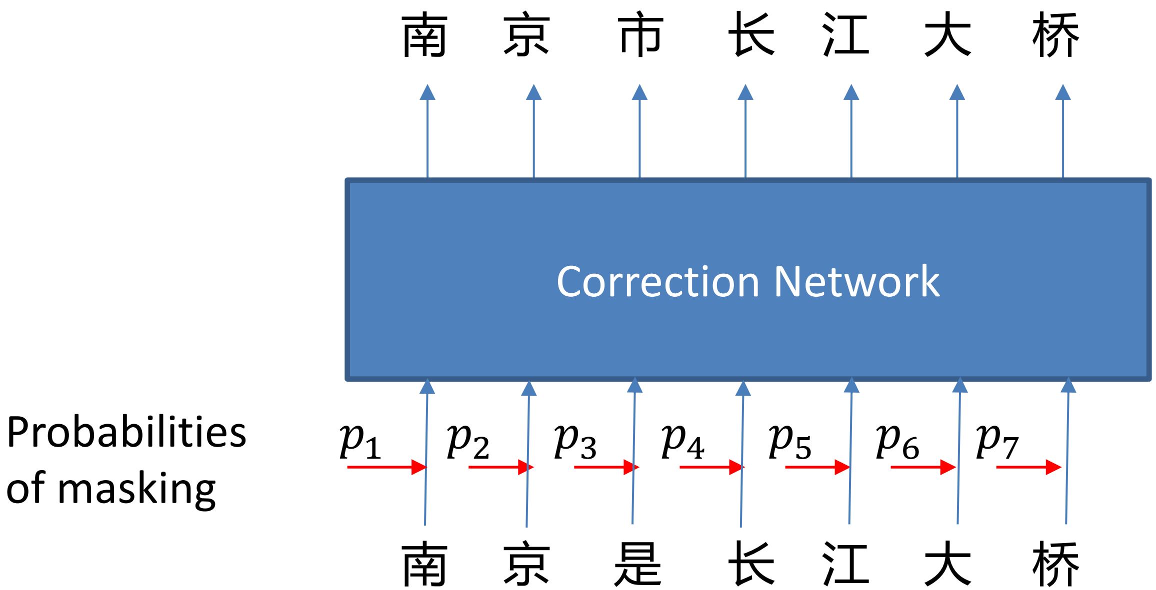
Architecture of Soft-Masked BERT

- Consists of detection network and correction network
- Detection network: bi-directional GRU
- Correction network: BERT



Correction Network

- Soft masking at each position
- Embeddings with soft masking are input



Experimental Results

Test Set	Method	Detection				Correction			
		Acc.	**Prec.**	**Rec.**	**F1.**	**Acc.**	**Prec.**	**Rec.**	**F1.**
SIGHAN	NTOU (2015)	42.2	42.2	41.8	42.0	39.0	38.1	35.2	36.6
	NCTU-NTUT (2015)	60.1	71.7	33.6	45.7	56.4	66.3	26.1	37.5
	HanSpeller++ (2015)	70.1	**80.3**	53.3	64.0	69.2	**79.7**	51.5	62.5
	Hybird (2018b)	-	56.6	69.4	62.3	-	-	-	57.1
	FASPell (2019)	74.2	67.6	60.0	63.5	73.7	66.6	59.1	62.6
	Confusionset (2019)	-	66.8	73.1	69.8	-	71.5	59.5	64.9
	BERT-Pretrain	6.8	3.6	7.0	4.7	5.2	2.0	3.8	2.6
	BERT-Finetune	80.0	73.0	70.8	71.9	76.6	65.9	64.0	64.9
	Soft-Masked BERT	**80.9**	73.7	**73.2**	**73.5**	**77.4**	66.7	**66.2**	**66.4**
News Title	BERT-Pretrain	7.1	1.3	3.6	1.9	0.6	0.6	1.6	0.8
	BERT-Finetune	80.0	65.0	61.5	63.2	76.8	55.3	52.3	53.8
	Soft-Masked BERT	**80.8**	**65.5**	**64.0**	**64.8**	**77.6**	**55.8**	**54.5**	**55.2**

AMBERT (A Multi-Grained BERT)

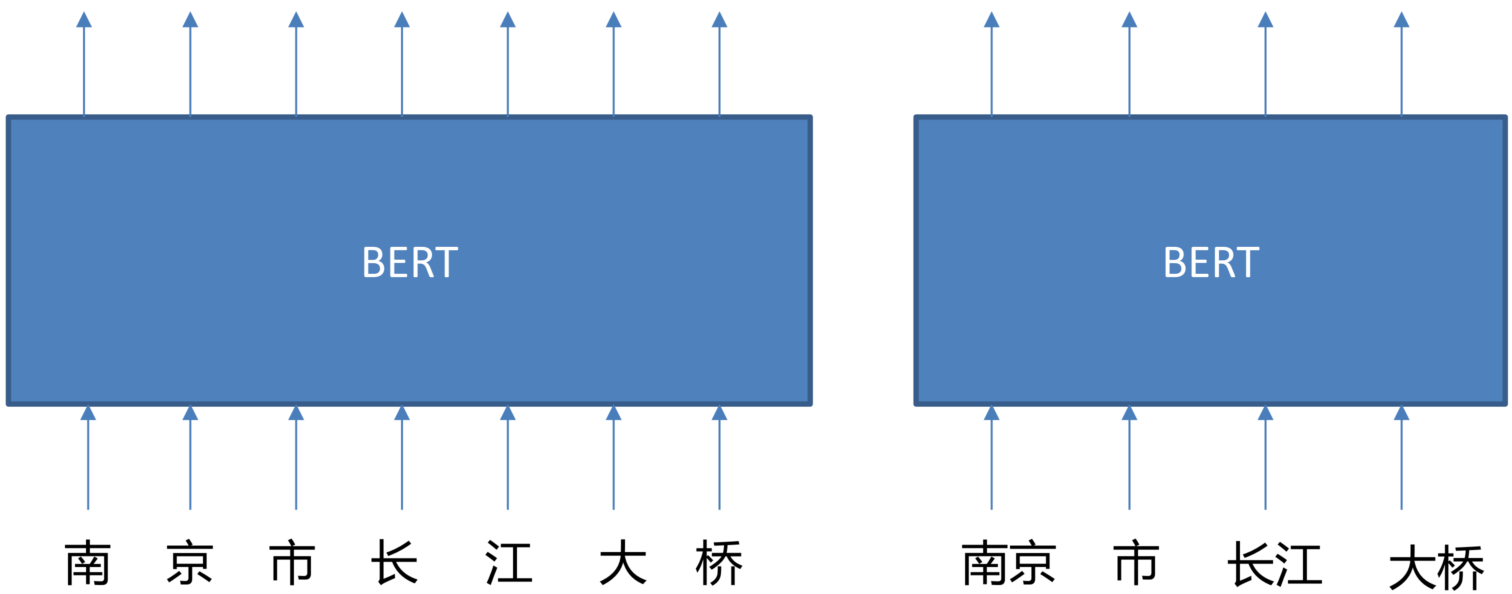
- A new technique for pre-trained language modeling
- Using both multi-grained tokens, e.g., characters and words in Chinese
- Taking BERT as example
- State of the art performances on Chinese and English language understanding tasks

Fine-Grained vs Coarse-Grained Language Processing

- Fine-grained tokens are less complete as lexical units but their representations are easier to train
- Coarse-grained tokens are more complete as lexical units but their representations are harder to train
- Tokenization can be incorrect
- Sometimes it is better to retain both fine-grained and coarse-grained tokens

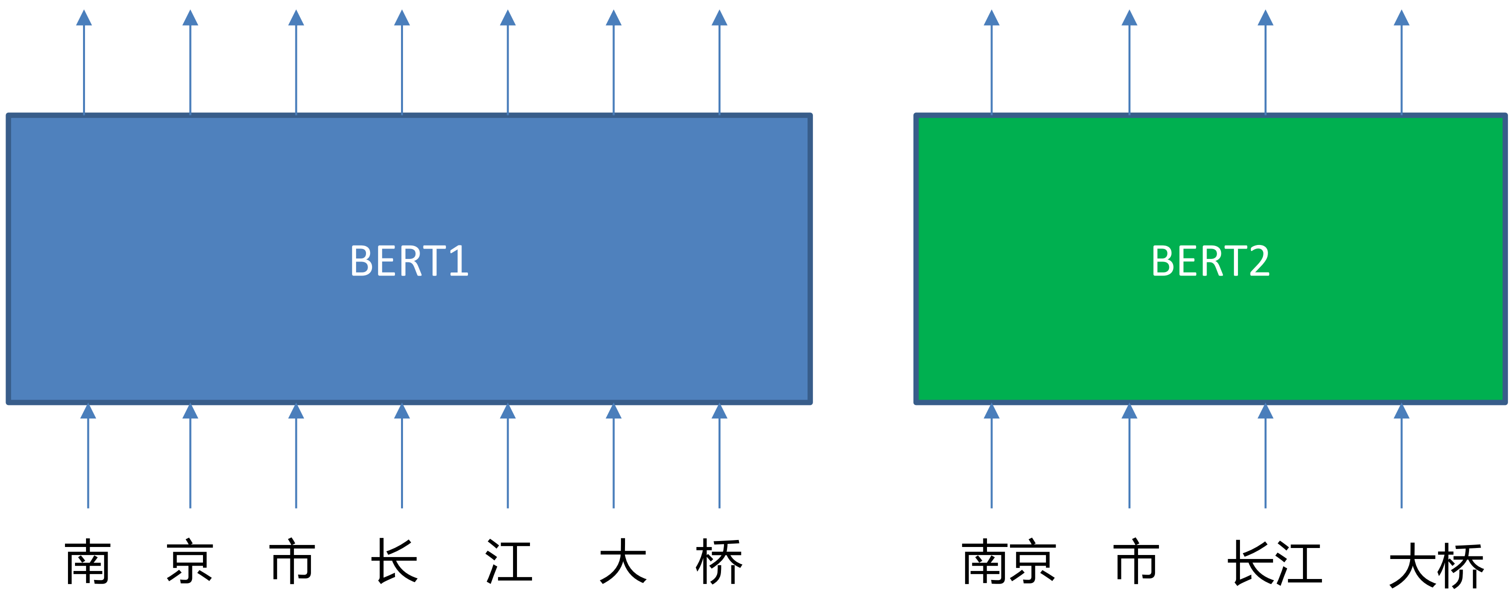
Architecture of AMBERT

- Two BERT models for multi-grained inputs
- Two models work in parallel and share parameters
- Perform best among existing pre-trained models



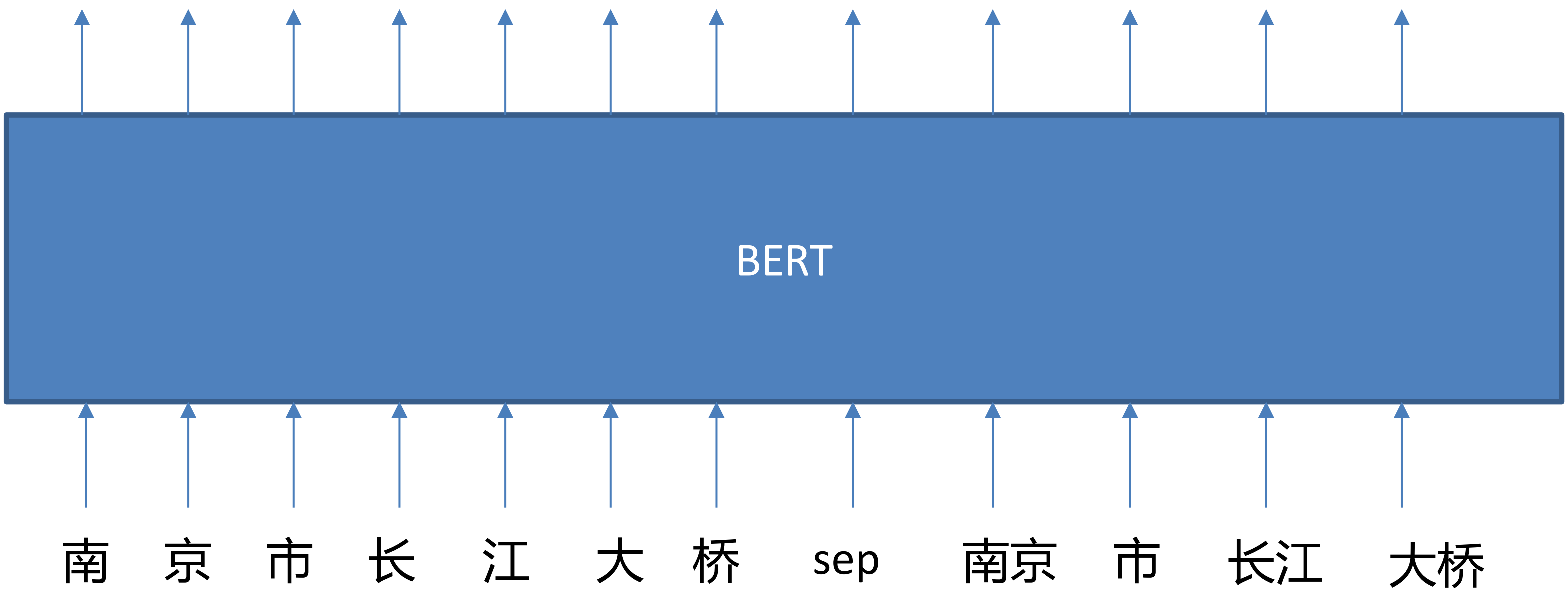
An Alternative Architecture

- Two BERT models for multi-grained inputs
- Two models work in parallel, but do not share parameters
- Does not work better than AMBERT



Another Alternative Architecture

- One BERT model for multi-grained inputs
- Model share parameters
- Does not work bette than AMBERT



Experimental Results

Chinese CLUE Data Sets

Model	Params	Avg.	TNEWS†	IFLYTEK	WSC.†	AFQMC	CSL†	CMNLI	CMRC.	ChID	C^3
Google BERT	108M	72.59	66.99	60.29	71.03	73.70	83.50	79.69	71.60	82.04	64.50
XLNet-mid	200M	73.00	66.28	57.85	78.28	70.50	84.70	81.25	66.95	83.47	67.68
ALBERT-xlarge	60M	73.05	66.00	59.50	69.31	69.96	84.40	81.13	**76.30**	80.57	**70.32**
ERNIE	108M	74.20	68.15	58.96	**80.00**	73.83	85.50	80.29	74.70	82.28	64.10
RoBERTa	108M	74.38	67.63	**60.31**	76.90	**74.04**	84.70	80.51	75.20	83.62	66.50
AMBERT	176M	**75.28**	**68.58**	59.73	78.28	73.87	**85.70**	**81.87**	73.25	**86.62**	69.63

English GLUE Data Sets

Model	Params	Avg.	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	SQuAD	RACE
Google BERT	110M	78.7	52.1*	93.5*	84.8*	85.8*	89.2*	84.6*	90.5*	66.4*	75.5	64.3*
XLNet	110M	78.6	47.9	94.3	83.3	84.1	89.2	86.8	91.7	61.9	79.9*	66.7*
SpanBERT	110M	79.1	51.2	93.5	87.0	82.9	89.2	85.1	92.7	69.7	81.8	57.4
ELECTRA	110M	81.3	59.7*	93.4*	86.7*	87.7*	89.1*	85.8*	92.7*	73.1*	74.8	69.9
ALBERT	12M	80.1	53.2	93.2	87.5	87.2	87.8	85.0	91.2	71.1	78.7	65.8
RoBERTa	135M	82.7	**61.5**	**95.8**	88.7	**88.9**	89.4	**87.4**	**93.1**	**74.0**	78.6	69.9
AMBERT‡	194M	**82.8**	60.0	95.2	**88.9**	88.2	**89.5**	87.2	92.6	72.6	**82.5**	**71.2**

Talk Outline

- Past
  - n-gram Language Model
- Present
  - Neural Language Model
  - Pretrained Language Model
- Our Work
  - Soft Masked BERT
  - AMBERT
- *Future*
  - *Brain-Inspired Language Model*

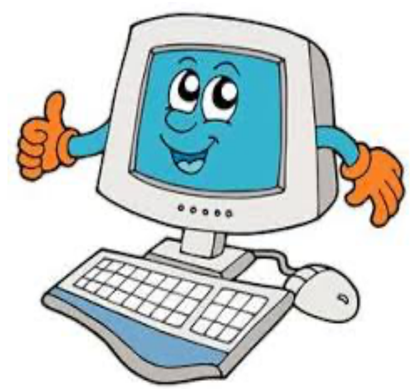
Power and Limitation of Pre-trained Language Model

- Is close to or on par with humans in many language processing tasks
- Mimic human language behaviors with pre-trained language models
- Not the same as human language processing

Mimic Human Language Generation

$w_1, w_2, \cdots, w_N$

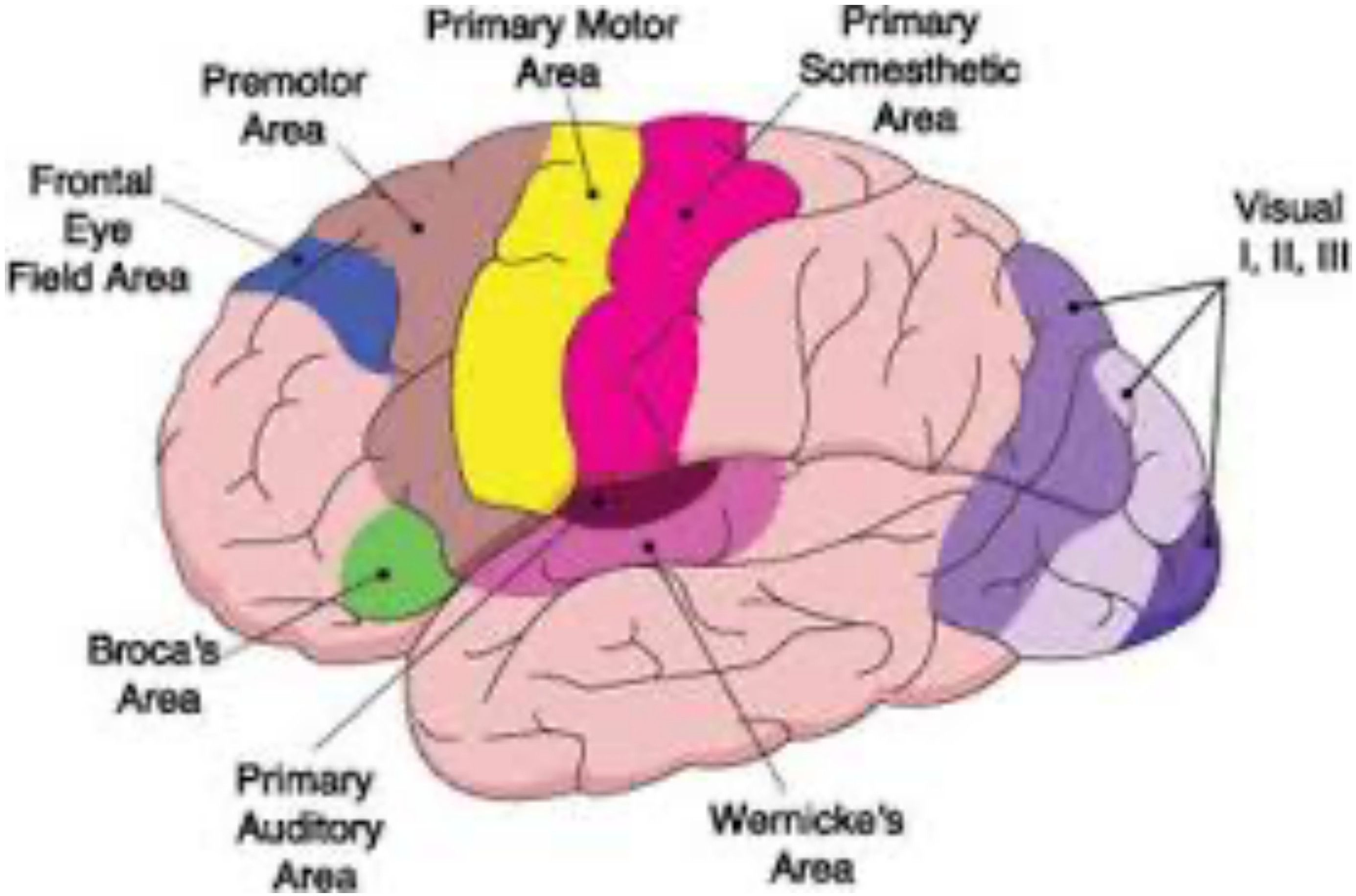
$w_1, w_2, \cdots, w_N$



Brain-Inspired Language Model: Grammar

- Human language processing
- Broca’s area: responsible for syntax
- Wernick’s area: responsible for lexicon
- Hypothesis: language processing is parallel processing
- Question: can grammar be incorporated into language model?

Areas in Cerebral Cortex



Brain-Inspired Language Model: Multimodality

- Human language processing
- Language understanding: related to visual, auditory, motion processing
- Multimodal processing
- Question: can multimodal language model be built with multimodal data?

Language Model and Knowlege

- Language model implicitly contains certain knowledge (linguistic, world knowledge, etc)
- Store simple knowledge as patterns
- Does not store complex knowledge

Talk Outline

- Past
  - n-gram Language Model
- Present
  - Neural Language Model
  - Pretrained Language Model
- Our Work
  - Soft Masked BERT
  - AMBERT
- Future
  - Brain-Inspired Language Model

Take-away Messages

- Language model has history of over one-hundred years
- From n-gram language model to neural language model and pre-trained language model
- Present: pre-trained language model approach is powerful, although having limitation
- We proposed Soft Masked BERT and AMBERT
- Future: grammar-incorporated and multimodal models

References

- Zhang, X. and Li, H., 2020. AMBERT: A Pre-trained Language Model with Multi-Grained Tokenization. arXiv preprint arXiv:2008.11869.
- Zhang, S., Huang, H., Liu, J. and Li, H., 2020. Spelling Error Correction with Soft-Masked BERT. arXiv preprint arXiv:2005.07421.
- 李航，语言模型：过去、现在、未来，计算机学会通讯，2020

Thank you!

lihang.lh@bytedance.com