A Generalized Language Model in Tensor Space

Tianjin University

Lipeng Zhang, Peng Zhang, Xindian Ma, Shuqin Gu,
Zhan Su, Dawei Song

AAAI 2019

Outline

- **Motivation**
- Background
- TSLM basic representation
- Generalization
- Recursive Language Modeling
- Experiment

Motivation

- Some classical work usually represent a sentence or document using vectors or matrices:
  - VSM (Vector Space Model)
  - LSI (Latent Semantic Index)
  - N-Gram (Bi-gram, Tri-gram)
  - Embeddings
  - …

Bi-Gram

——Count a co-occurrence matrix

	w_1	w_2	w_3	w_4	$\cdots$	
w_1		$p(w_1, w_2)$			$\cdots$	$p(w_1)$
w_2						
w_3						
w_4						
$\vdots$						

$$p(w_2|w_1) = \frac{p(w_1, w_2)}{p(w_1)}$$

Motivation

——A text representation method by tensors

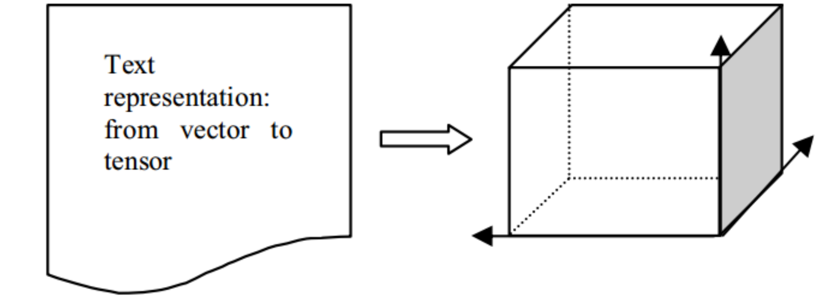


Figure 1. A document is represented as a character level 3-order tensor

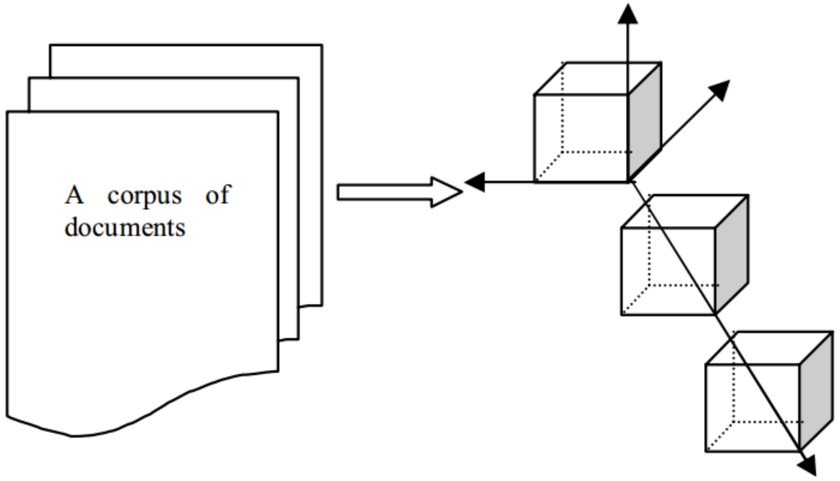
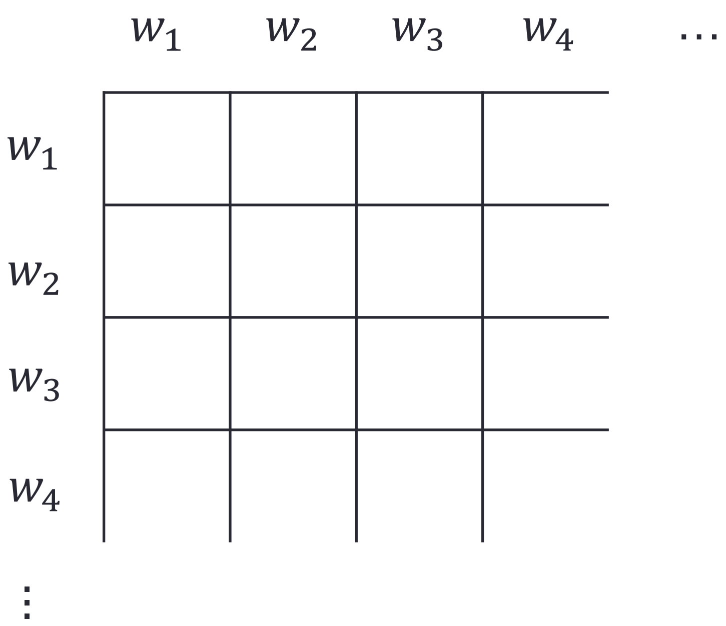


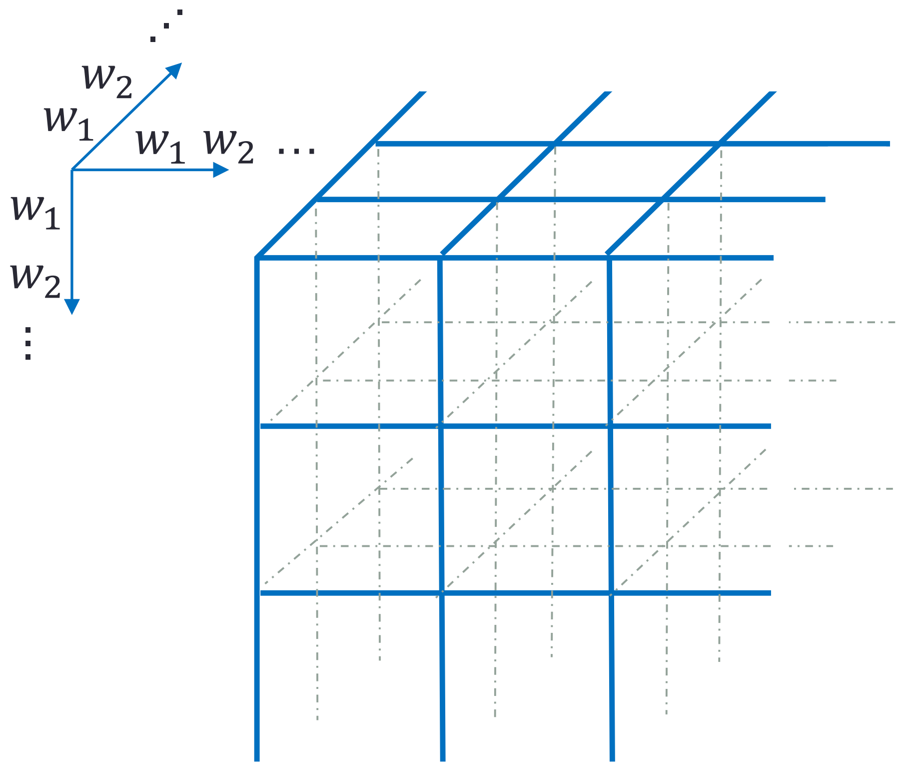
Figure 2. A corpus of documents is represented as a 4-order tensor

Representing the text as a 3-order tensor

Liu, N.; Zhang, B.; Yan, J.; and Chen, Z. 2005. Text representation: from vector to tensor. In IEEE International Conference on Data Mining, 725–728

Bi-Gram, Tri-Gram…





Motivation

- A vector can be considered as a 1-order tensor;
- A matrix can be considered as a 2-order tensor.

- The existing methods usually adopt relatively low-order tensors, which have limited expressive power in modeling language.

- We propose a language model based on relatively high-order tensor representation——Tensor Space Language Model (TSLM).

Challenges

- 1.To construct a high-order tensor representation;
- 2.To derive an effective solution for such representation;
- 3.To demonstrate such a solution is a general approach for language modeling;
- 4.To solve that such a high-order tensor contains exponential magnitude of parameters;
- …

How to solve these challenges

- We will introduce the Tensor Network (TN) for effectively representing the high-order tensors;
- Theoretically, we prove that TSLM is a generalization of the n-gram language model;
- With the help of tensor decomposition, the high dimensionality of parameters in tensor space can be reduced greatly.

Outline

- Motivation
- **Background**
- TSLM basic representation
- Generalization
- Recursive Language Modeling
- Experiment

Background

- Tensor

A tensor : a mutidimensional array

The order : the number of indexing entries

The dimension : the number of values in a particular order

$$\mathcal{T} \in \mathbb{R}^{m_1 \times \cdots \times m_n}, \qquad \mathcal{T}_{d_1 \ldots d_n} \in \mathbb{R}$$

Tensor product : denoted by $\otimes$, maps two low-order tensors to a high-order tensor.

$$\mathcal{A} \in \mathbb{R}^{m_1 \times \cdots \times m_j}, \qquad \mathcal{B} \in \mathbb{R}^{m_{j+1} \times \cdots \times m_{j+k}}$$

$$\mathcal{A} \otimes \mathcal{B} = \mathcal{T} \in \mathbb{R}^{m_1 \times \cdots \times m_{j+k}}, \qquad \mathcal{T}_{d_1 \ldots d_{j+k}} = \mathcal{A}_{d_1 \ldots d_j} \cdot \mathcal{B}_{d_{j+1} \ldots d_{j+k}}$$

Background

- Tensor

The tensor product of n vectors is a n-order rank-one tensor :

$$\mathcal{A} = \boldsymbol{a}_1 \otimes \boldsymbol{a}_2 \otimes \cdots \otimes \boldsymbol{a}_n$$

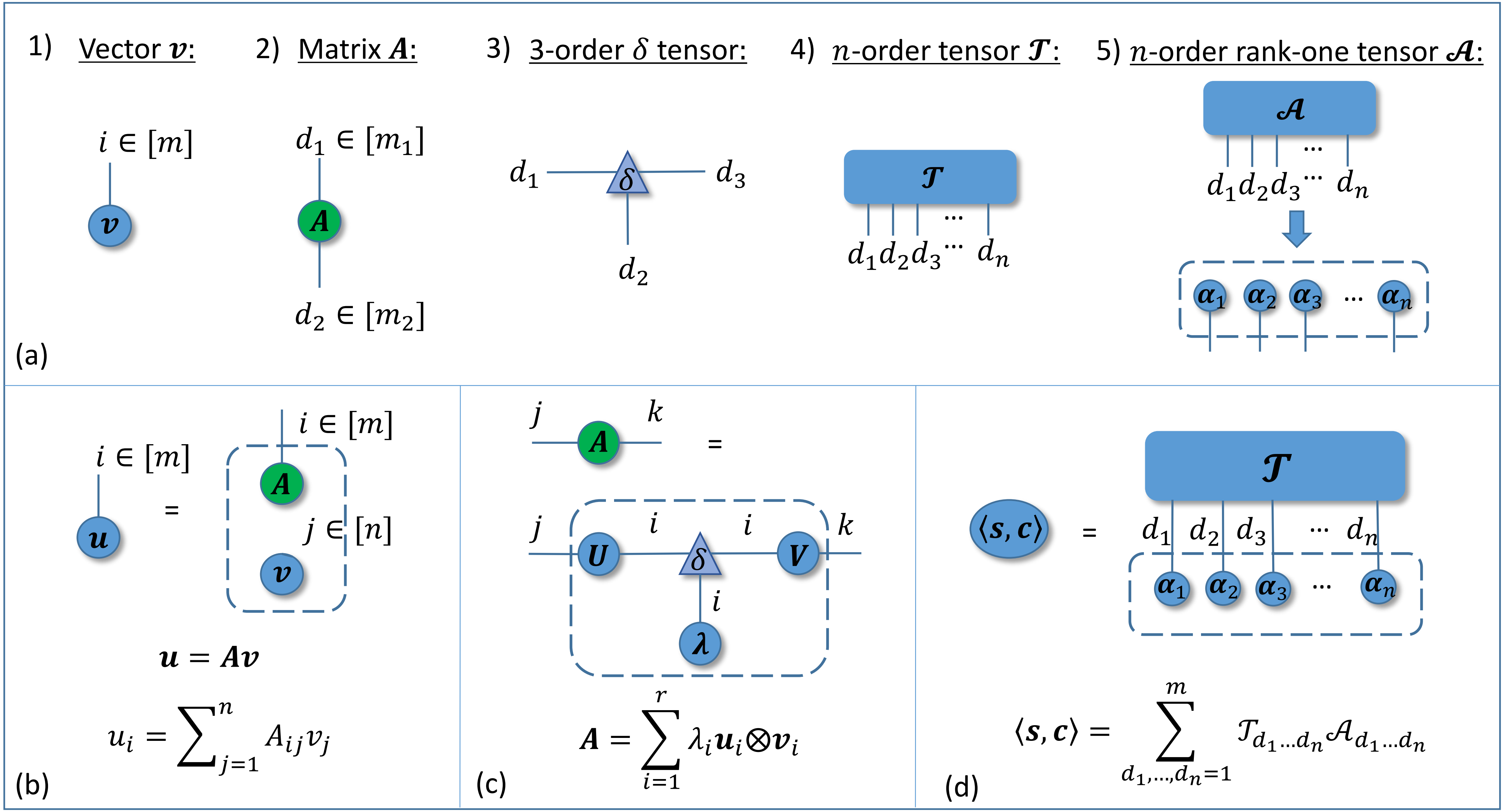
The rank of a tensor $\mathcal{T}$ is defined as the smallest number of rank-one tensors that generate $\mathcal{T}$ as their sum.

The inner product of two tensors returns a scalar value that is sum of the products of their entries.

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{d_1,\ldots,d_n=1}^{m} \mathcal{A}_{d_1 \ldots d_n} \mathcal{B}_{d_1 \ldots d_n}$$

Background

- Tensor and Tensor Networks
  - Tensor Network is formally represented an undirected and weight graph;
  - Tensor operations (e.g., multiplication, inner product, decomposition) can be represented intuitively in tensor networks.



1) Vector $\boldsymbol{v}$:
2) Matrix $\boldsymbol{A}$:
3) 3-order δ tensor:
4) n-order tensor $\boldsymbol{\mathcal{T}}$:
5) n-order rank-one tensor $\boldsymbol{\mathcal{A}}$:
$i \in [m]$
$d_1 \in [m_1]$
$d_2 \in [m_2]$
d_1 d_2 d_3
$d_1 d_2 d_3 \cdots d_n$
α_1 α_2 α_3 … α_n
(a)
$j \in [n]$
$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}$
$u_i = \sum_{j=1}^{n} A_{ij} v_j$
(b)
j k i
$\boldsymbol{A} = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i$
(c)
$\langle \boldsymbol{s}, \boldsymbol{c} \rangle$ =
$\langle \boldsymbol{s}, \boldsymbol{c} \rangle = \sum_{d_1,\ldots,d_n=1}^{m} \mathcal{T}_{d_1 \ldots d_n} \mathcal{A}_{d_1 \ldots d_n}$
(d)

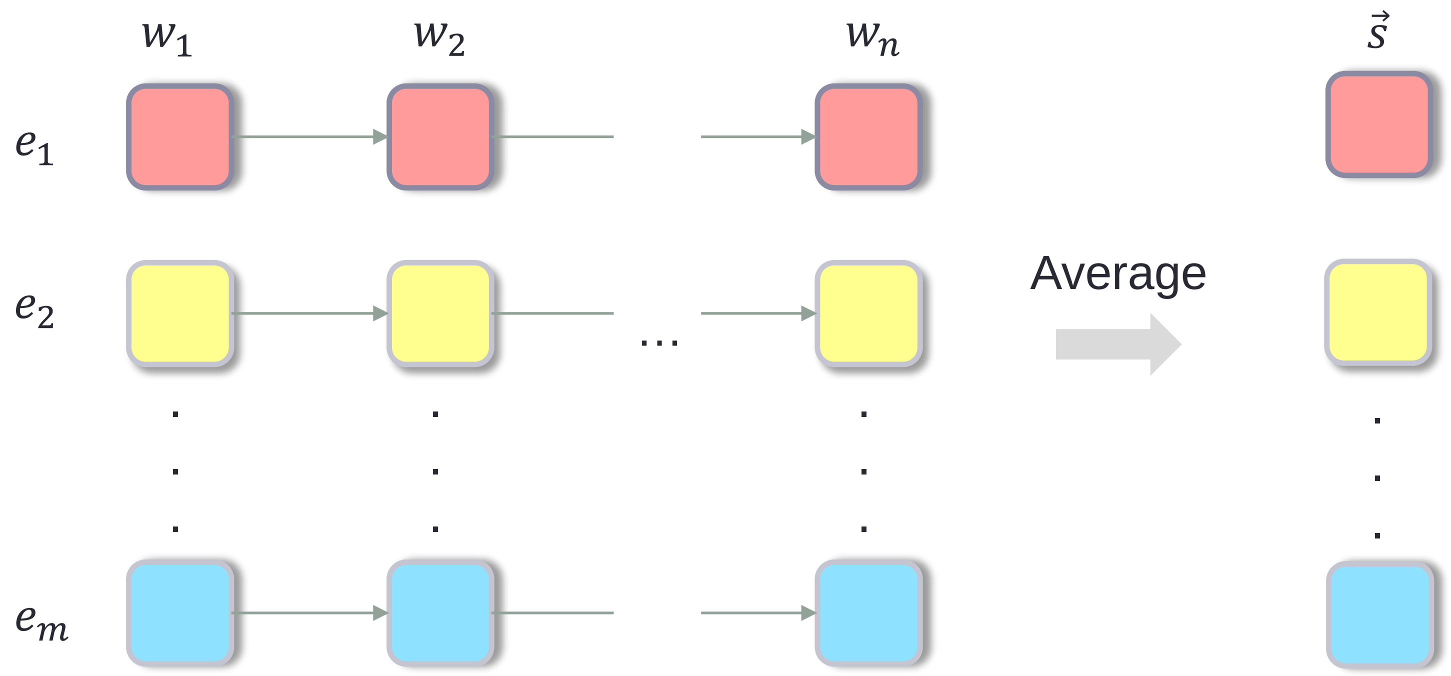
Outline

- Motivation
- Background
- **TSLM basic representation**
- Generalization
- Recursive Language Modeling
- Experiment

TSLM basic representation

——A basic text representation by words average

- Hypotheses: A sentence has n words. Each word has m semantic meanings.

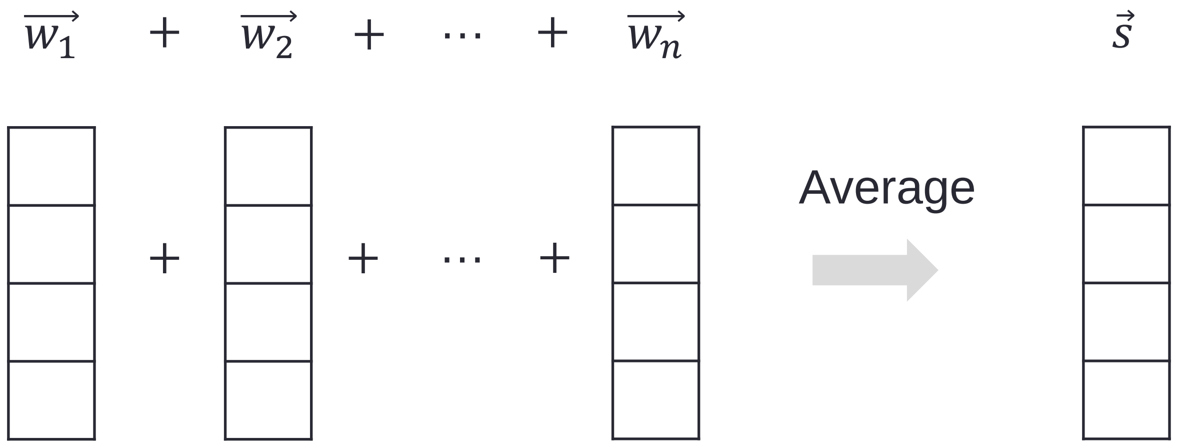


The sentence still has m semantic meanings.

TSLM basic representation

——A basic text representation by words average

- We can present such combination method as from vector to vector.



The sentence is still in vector space.

TSLM basic representation

- How to represent a single word

$$w_i = \sum_{d_i=1}^{m} \alpha_{i,d_i} e_{d_i}$$

- How to represent a sentence

Rank One $\boxed{s = w_1 \otimes \cdots \otimes w_n}$

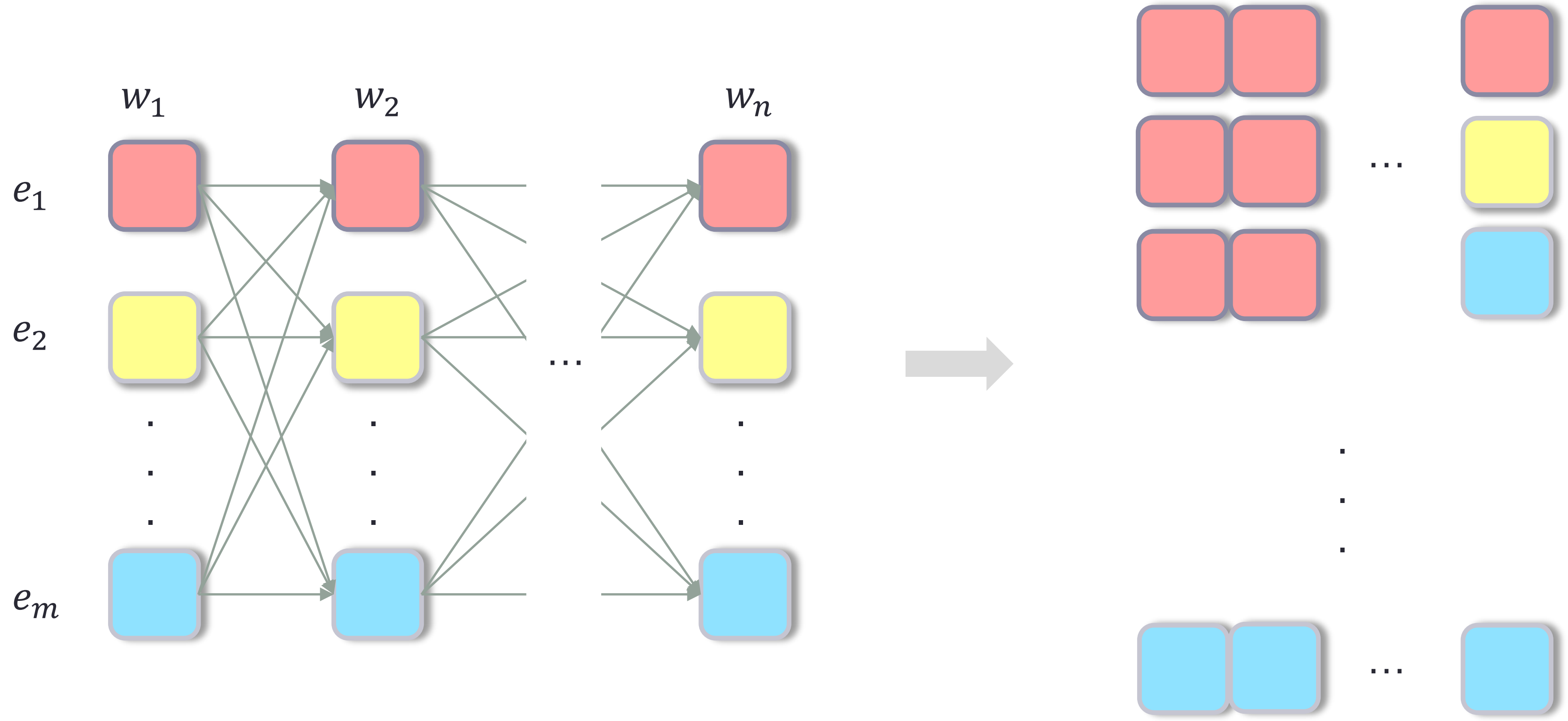
$$\boxed{\mathcal{A}_{d_1 \cdots d_n} = \prod_{i=1}^{n} \alpha_{i,d_i}}$$

$$s = \sum_{d_1, \cdots, d_n = 1}^{m} \boxed{\mathcal{A}_{d_1 \cdots d_n}} e_{d_1} \otimes \cdots \otimes e_{d_n}$$

TSLM basic representation

——How to construct a high-order tensor representation

- Hypotheses: A sentence has n words. Each word has m semantic meanings.



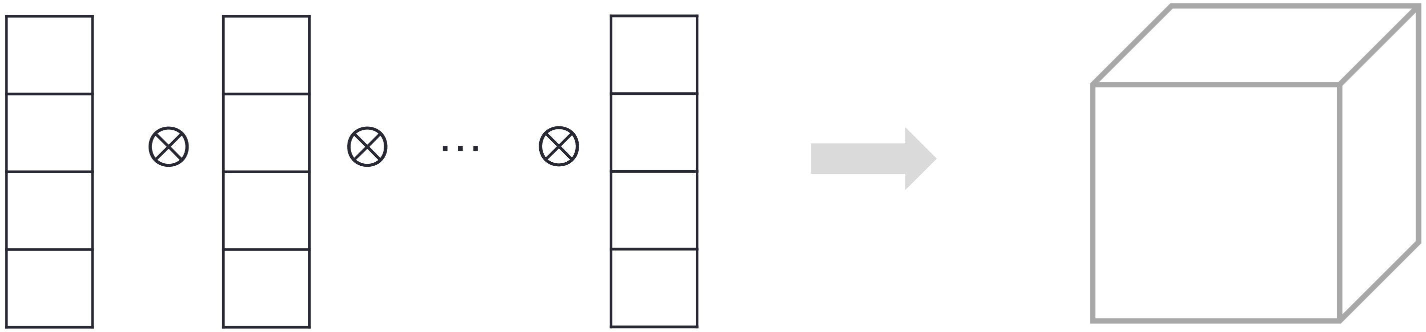
According to the fully arranged combination, we will get m^n semantic combinations.

TSLM basic representation

——How to construct a high-order tensor representation

- We can present such combination method as from vector to tensor using tensor product.

$$\overrightarrow{w_1} \otimes \overrightarrow{w_2} \otimes \cdots \otimes \overrightarrow{w_n}$$



However, it is difficult to represent a high-order tensor. (A cube can only represent a 3-order tensor)

TSLM basic representation

- Assume that each sentence s_i appears with a probability p_i.
- We can denote the corpus as:

$$c = \sum_i p_i s_i = \sum_{d_1 \ldots d_n = 1}^{m} \mathcal{T}_{d_1 \ldots d_n} e_{d_1} \otimes \cdots \otimes e_{d_n}$$

- The sentence probability:

$$p(s) = \langle s, c \rangle = \sum_{d_1 \ldots d_n = 1}^{m} \mathcal{T}_{d_1 \ldots d_n} \mathcal{A}_{d_1 \ldots d_n}$$

Outline

- Motivation
- Background
- TSLM basic representation
- **Generalization**
- Recursive Language Modeling
- Experiment

A Generalization of N-Gram Language Model

- N-gram Language Model
  - estimate the probability distribution of sentences

$$s = (w_1, \dots, w_n) := w_1^n$$

  - Compute a sentence's joint probability

$$p(s) = p(w_1^n) = p(w_1) \prod_{i=2}^{n} p(w_i) p(w_i | w_1^{i-1})$$

  - Compute the current word's conditional probability

$$p\left(w_i \middle| w_1^{i-1}\right) = \frac{p(w_1^i)}{p(w_1^{i-1})}$$

How to Prove TSLM as a Generalization of N-Gram

- Three hypotheses
  - The dimension of vector space $m = |\mathrm{V}|$
  - The represent of a word is an one-hot vector
  - The corpus:

$$c = \sum_i p_i s_i$$

Compute the joint probability

- N-gram language model
  - A sentence's joint probability

$$p(s) = p(w_1^n)$$

$$p(w_1^n) = p(w_1)\prod_{i=2}^{n} p(w_i|w_1^{i-1}) \quad ①$$

Compute the joint probability

- The sentence s will be represented as :

$$s = \sum_{d_1,\cdots,d_n=1}^{|V|} \mathcal{A}_{d_1\cdots d_n} w_{d_1} \otimes \cdots \otimes w_{d_n}$$

- Where

$$\mathcal{A}_{d_1\cdots d_n} = \begin{cases} 1, & d_k = \mathrm{index}(V, w_k) \\ 0, & \qquad\qquad otherwise \end{cases}$$

Compute the joint probability

- The corpus is $c = \sum p_i s_i$:

$$c = \sum_{d_1,\cdots,d_n=1}^{|\mathrm{V}|} \mathcal{T}_{d_1 \cdots d_n} w_{d_1} \otimes \cdots \otimes w_{d_n}$$

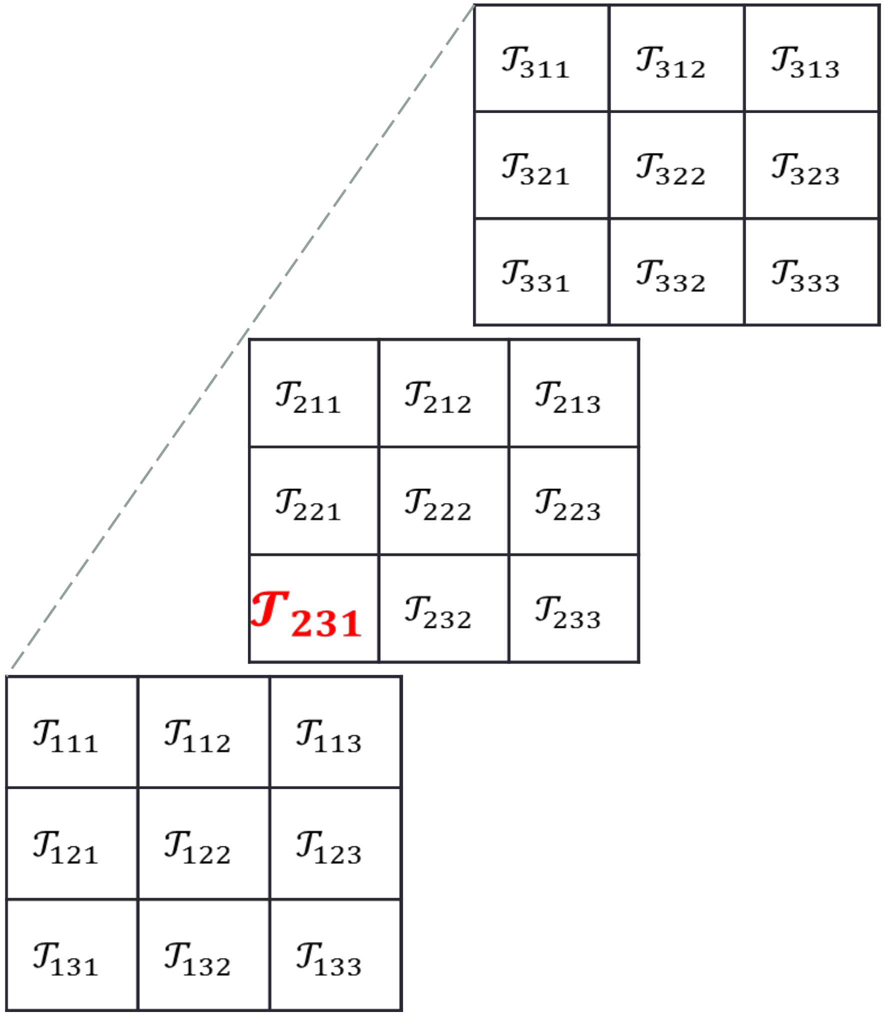
- Therefore, the probability of sentence

$$p_i = \langle s_i, c \rangle = \sum_{d_1,\cdots,d_n=1}^{|\mathrm{V}|} \mathcal{T}_{d_1 \cdots d_n} \mathcal{A}_{d_1 \cdots d_n}$$

An example

- The vocabulary :V $= \{A, B, C\}$
- The probability of each combination is one element in the right tensor
- If the sequence is $s_i = (B, C, A)$.
- The combination :

$$\mathrm{p}(s_i) = \mathcal{T}_{231}$$



An example

$$s_i = B \otimes C \otimes A$$

$$A = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$p(BCA) = \langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{A}} \rangle = \mathcal{T}_{231}$$

$\mathcal{T}_{311}$	$\mathcal{T}_{312}$	$\mathcal{T}_{313}$
$\mathcal{T}_{321}$	$\mathcal{T}_{322}$	$\mathcal{T}_{323}$
$\mathcal{T}_{331}$	$\mathcal{T}_{332}$	$\mathcal{T}_{333}$

$\mathcal{T}_{211}$	$\mathcal{T}_{212}$	$\mathcal{T}_{213}$
$\mathcal{T}_{221}$	$\mathcal{T}_{222}$	$\mathcal{T}_{223}$
$\boldsymbol{\mathcal{T}_{231}}$	$\mathcal{T}_{232}$	$\mathcal{T}_{233}$

$\mathcal{T}_{111}$	$\mathcal{T}_{112}$	$\mathcal{T}_{113}$
$\mathcal{T}_{121}$	$\mathcal{T}_{122}$	$\mathcal{T}_{123}$
$\mathcal{T}_{131}$	$\mathcal{T}_{132}$	$\mathcal{T}_{133}$

$\mathcal{T}$

0	0	0
0	0	0
0	0	0

0	0	0
0	0	0
1	0	0

0	0	0
0	0	0
0	0	0

$\mathcal{A}$

Compute the conditional probability

- N-Gram Language Model
  - The conditional probability can be calculated as:

$$p\left(w_i \middle| w_1^{i-1}\right) = \frac{p(w_1^i)}{p(w_1^{i-1})} \approx \frac{count(w_1^i)}{count(w_1^{i-1})}$$

- In TSLM

$$\frac{p(w_1^i)}{p(w_1^{i-1})} = \frac{\langle w_1^i, c\rangle}{\langle w_1^{i-1}, c\rangle}$$

②

Compute the conditional probability

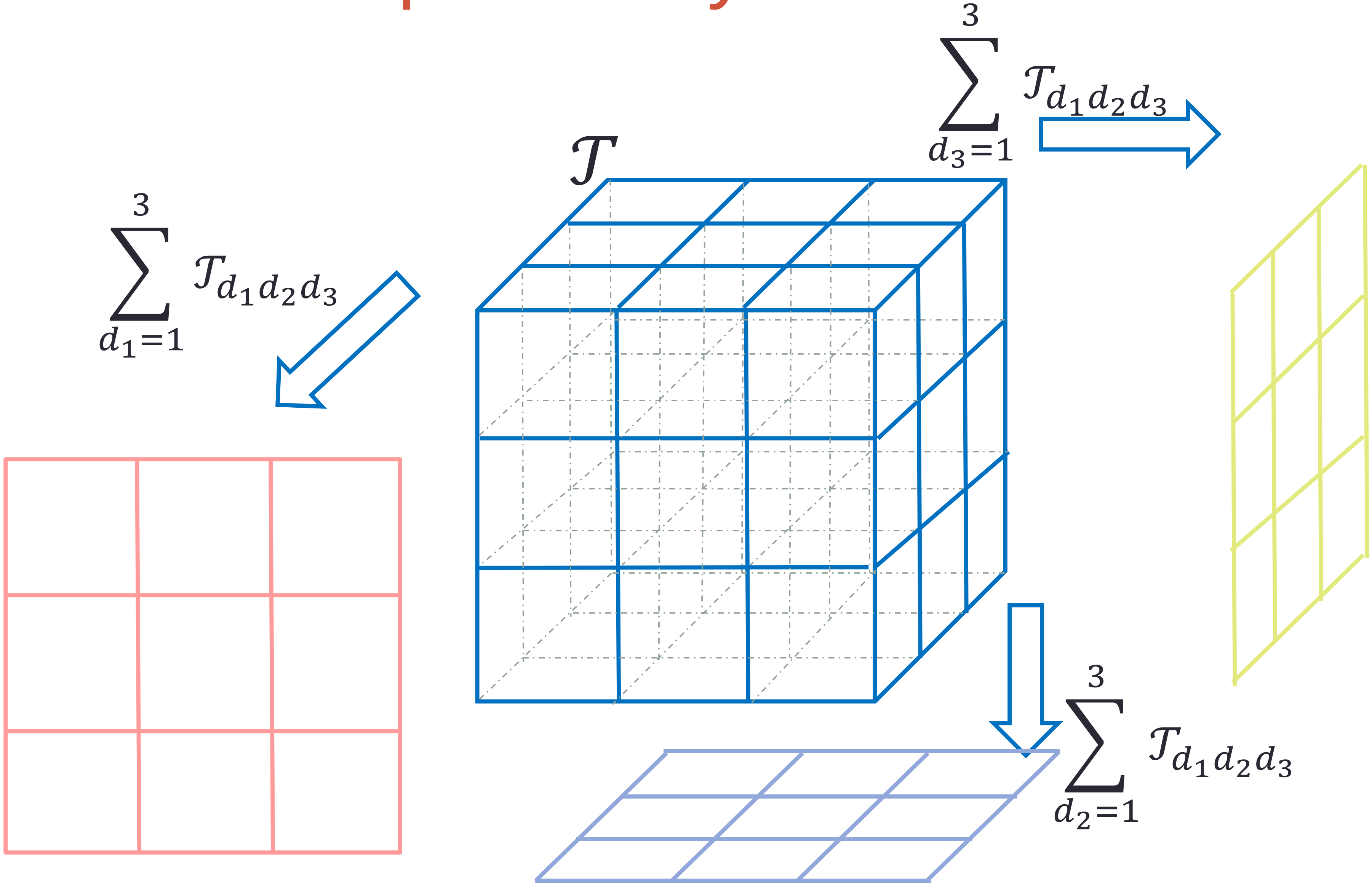
- In TSLM, we define marginal distribution as:

$$p(w_i) = \sum_{w_j \in \mathrm{V}} p(w_i, w_j)$$

$$p(w_1, \dots, w_{n-1}) = \sum_{w_n \in \mathrm{V}} p(w_1, \dots, w_{n-1}, w_n)$$

$$\begin{aligned} p(w_1^i) &= p(w_1, \cdots, w_i) \\ &= \sum_{w_{i+1,\dots,w_n} \in \mathrm{V}} p(w_1, \dots, w_i, w_{i+1}, \dots w_n) \\ &= \sum_{d_{i+1},\cdots,d_n=1}^{|\mathrm{V}|} \mathcal{T}_{d_1 \cdots d_n} \end{aligned}$$

The conditional probability in Tensor



Outline

- Motivation
- Background
- TSLM basic representation
- Generalization
- **Recursive Language Modeling**
- Experiment

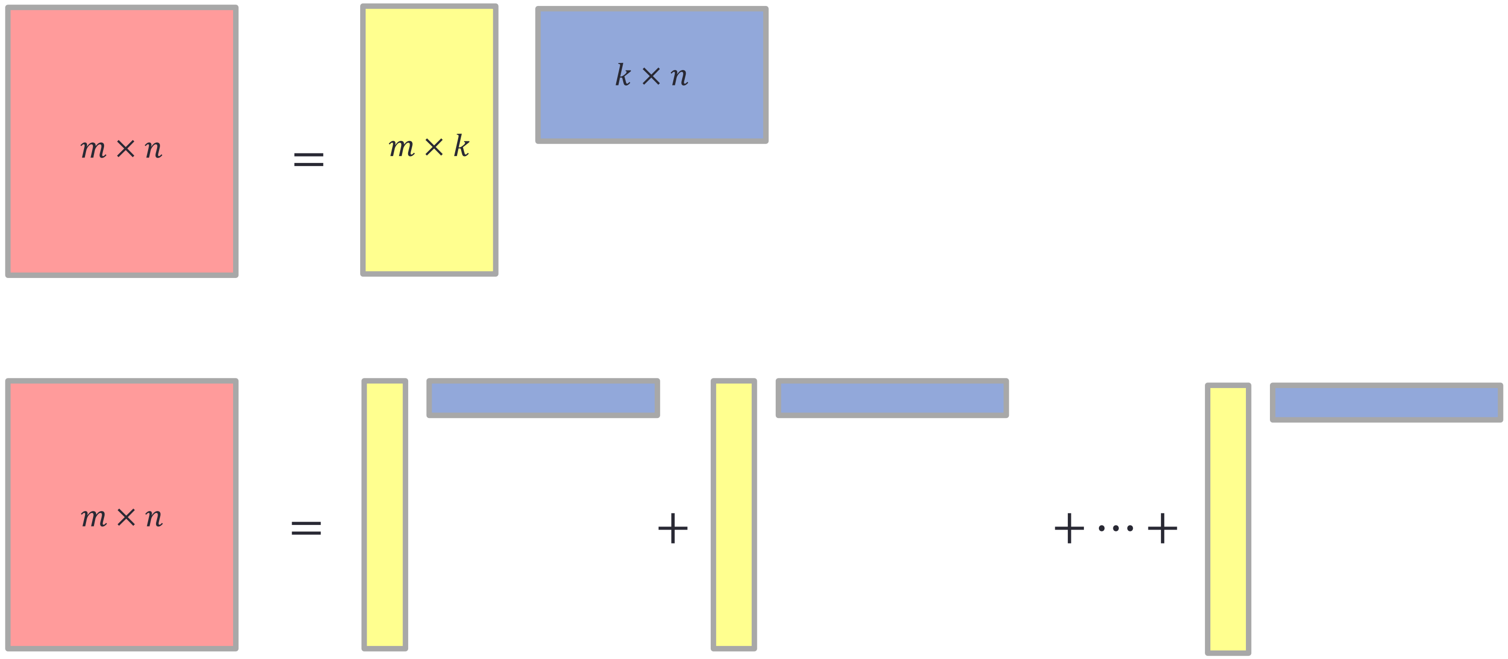
Recursive Language Modeling

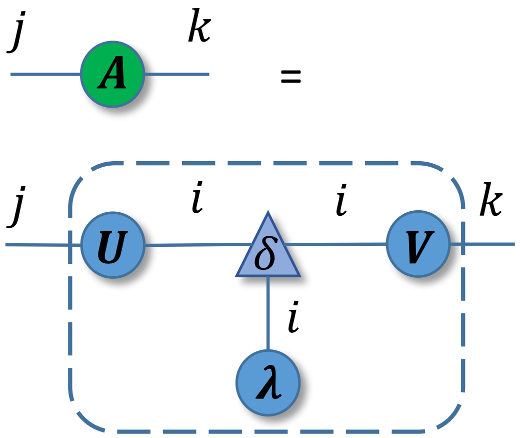
- Two hypotheses:
  - The dimensions of word vectors is $m \ll |\mathrm{V}|$
  - The corpus : $c = \sum_i p_i s_i$

- The high-order tensor $\mathcal{T}$ contains exponential magnitude of parameters (m^n) so that we can not effectively learn.

- We introduce the recursive tensor decomposition.

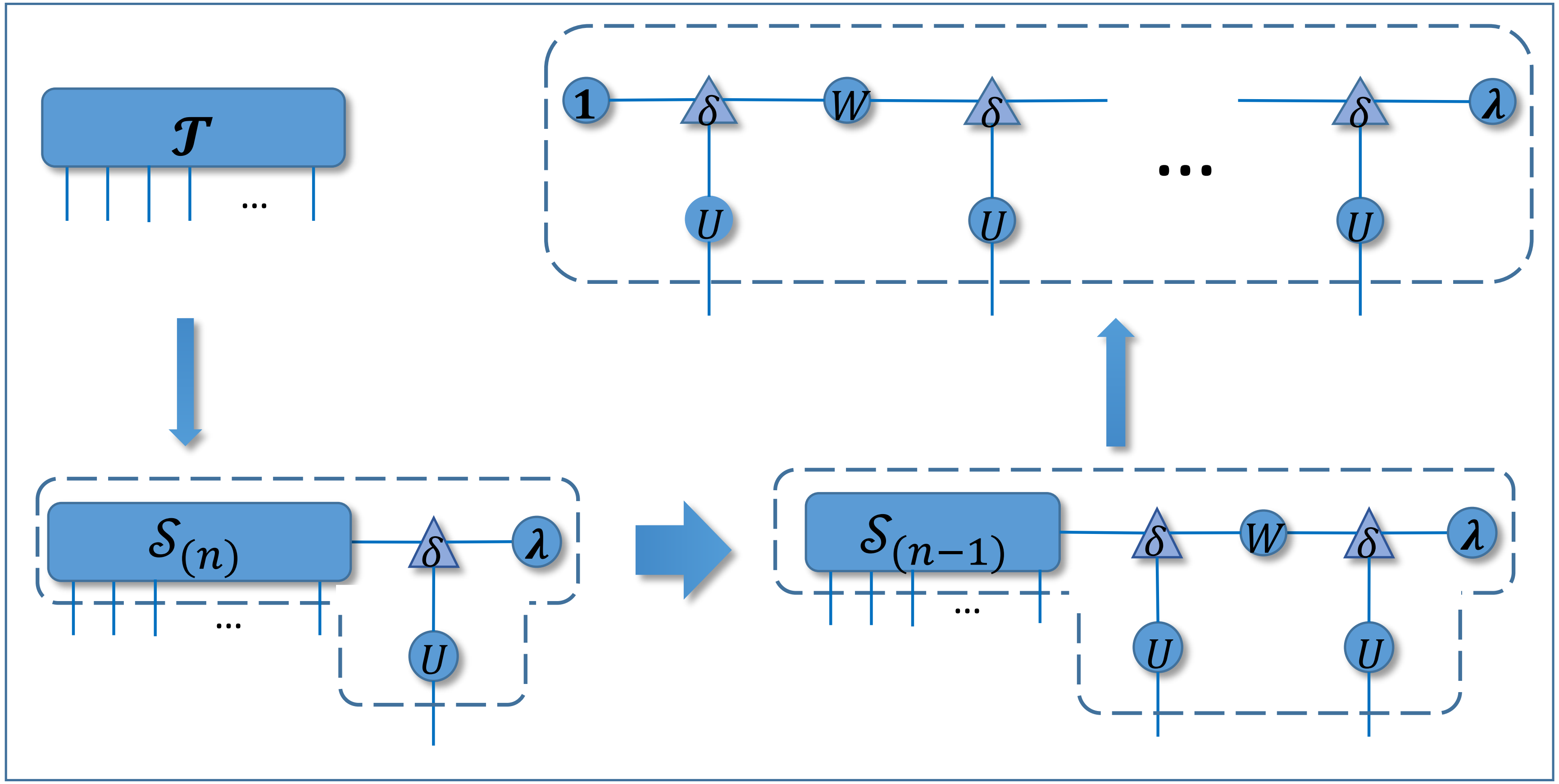
SVD(Single Value Decomposition)





$$A = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i$$

Recursive tensor decomposition



$$\mathcal{T} = \sum_{i=1}^{r} \lambda_i \mathcal{S}_{(n),i} \otimes \boldsymbol{u}_i$$

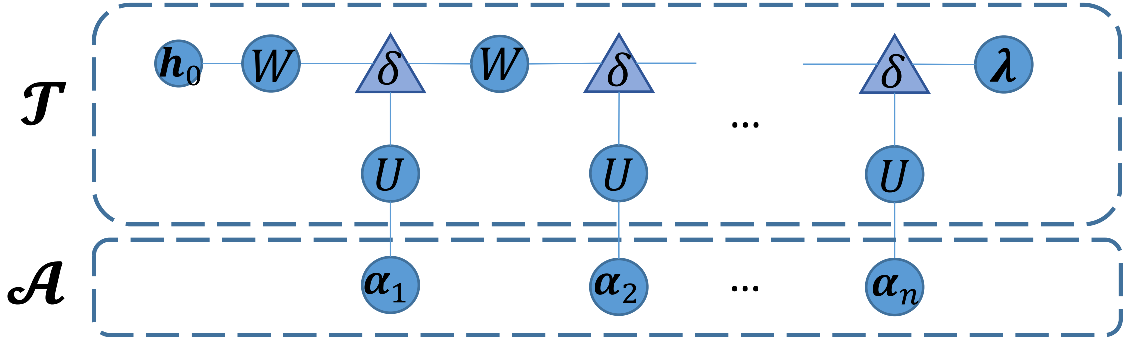
...

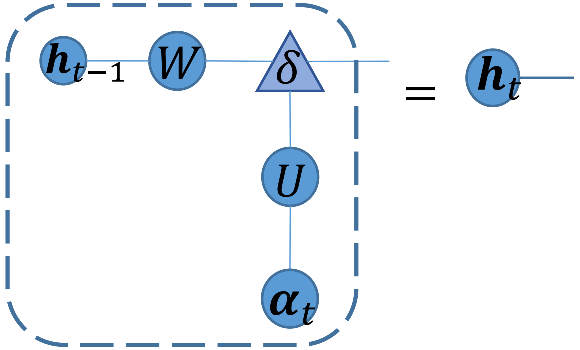
$$\mathcal{S}_{(n),k} = \sum_{i=1}^{r} W_{k,i} \mathcal{S}_{(n-1),i} \otimes \boldsymbol{u}_i$$

...

$$\mathcal{S}_{(1)} = \mathbf{1}$$

Recursive Language Modeling





$Denote: \ \boldsymbol{h}_0 = W^{-1}\mathbf{1}$

$Computing\ \boldsymbol{h}_t\ recursively:$

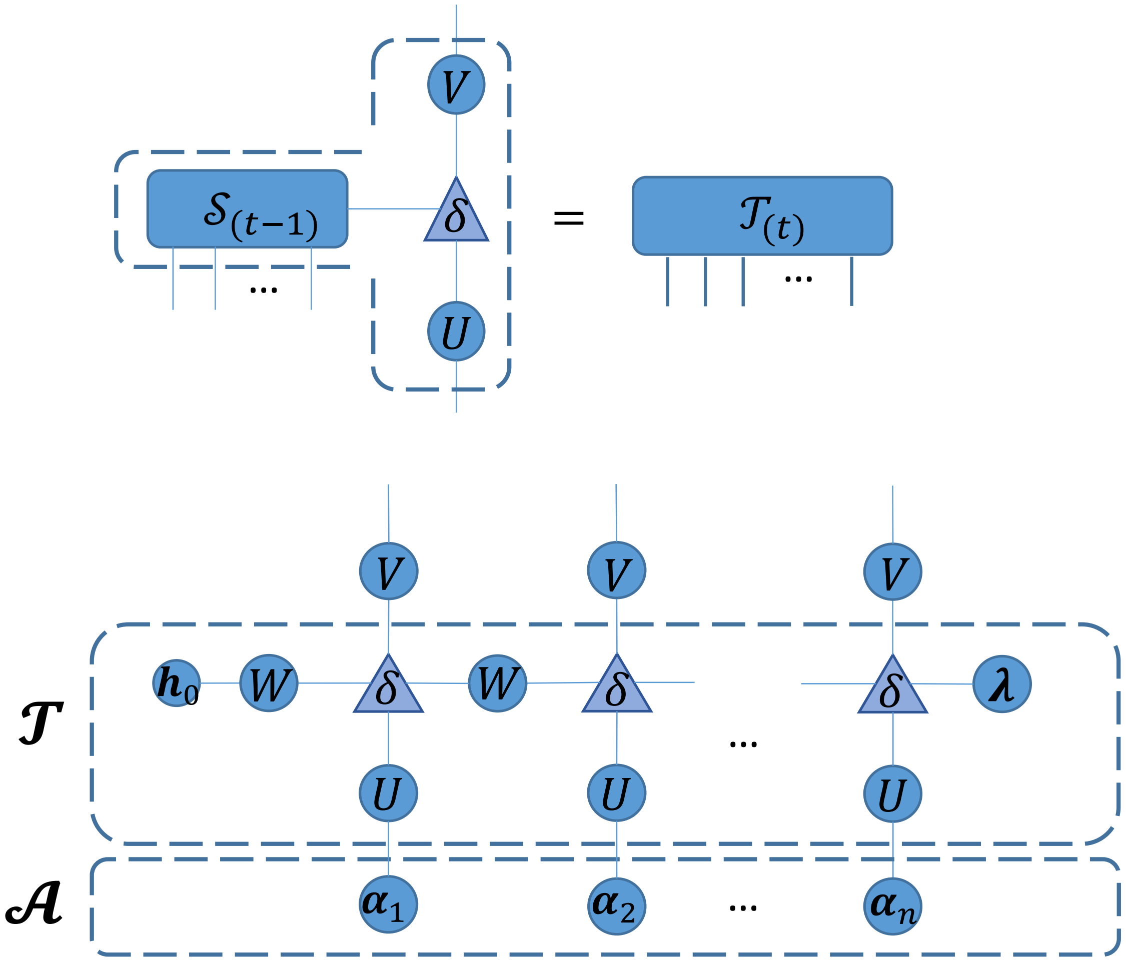
$$\boldsymbol{h}_1 = W\boldsymbol{h}_0 \odot U\boldsymbol{\alpha}_1$$

$$\ldots$$

$$\boldsymbol{h}_t = W\boldsymbol{h}_{t-1} \odot U\boldsymbol{\alpha}_t$$

$$\ldots$$

Recursive Language Modeling



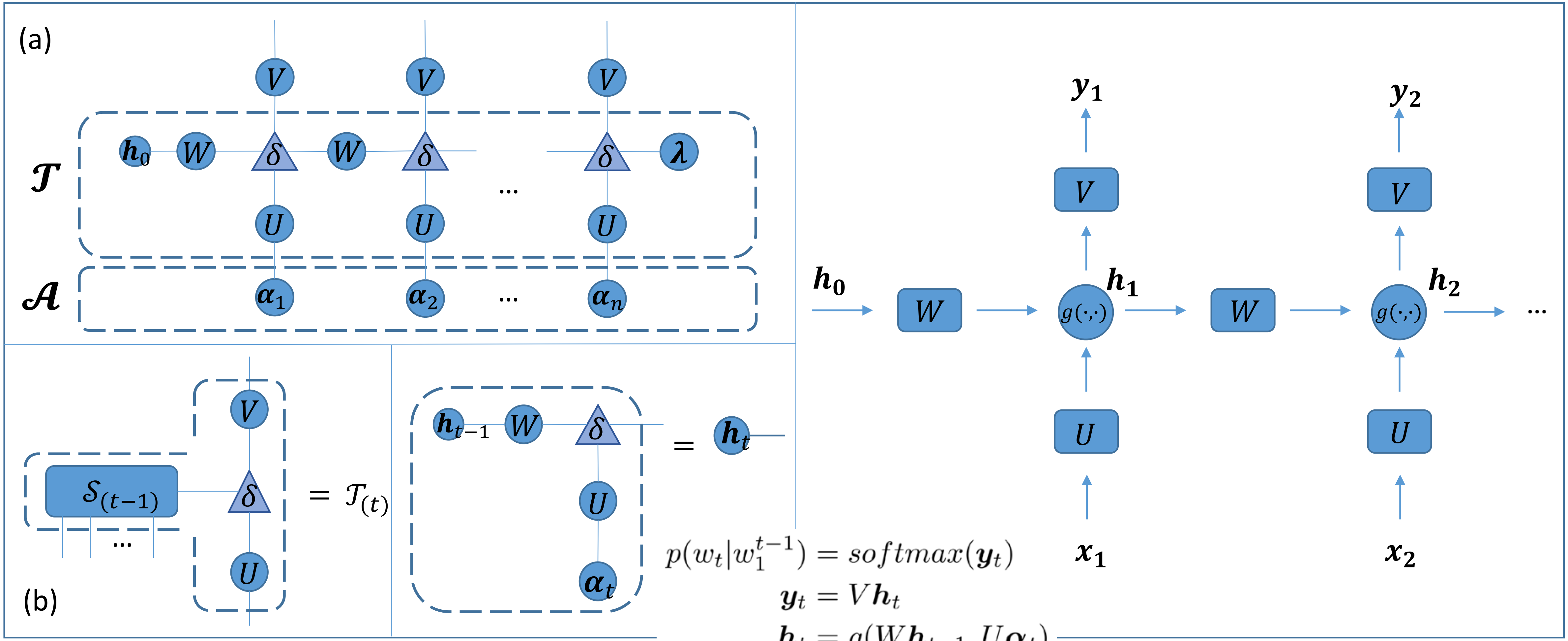
Contructing a tensor maping to vocabulary by a matrix $V \in \mathbb{R}^{r\times|\mathrm{V}|}$:

$$\mathcal{T}_{(t),k} = \sum_{i=1}^{r} V_{k,i}\mathcal{S}_{(t-1),i} \otimes \boldsymbol{u}_i$$

Computing coditional probability :

$$p(w_t|w_1^{t-1}) = softmax(\langle \mathcal{T}_{(t)}, \mathcal{A}_{(t-1)} \rangle)$$

Recursive Language Modeling



$$p(w_t|w_1^{t-1}) = softmax(\boldsymbol{y}_t)$$
$$\boldsymbol{y}_t = V\boldsymbol{h}_t$$
$$\boldsymbol{h}_t = g(W\boldsymbol{h}_{t-1}, U\boldsymbol{\alpha}_t)$$
$$g(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a} \odot \boldsymbol{b}$$

Outline

- Motivation
- Background
- TSLM basic representation
- Generalization
- Recursive Language Modeling
- **Experiment**

Experimental Result

	PTB				WikiText-2			
Model	**Hidden size**	**Layers**	**Valid**	**Test**	**Hidden size**	**Layers**	**Valid**	**Test**
KN-5(Mikolov and Zweig 2012)	-	-	-	141.2	-	-	-	-
RNN(Mikolov and Zweig 2012)	300	1	-	124.7	-	-	-	-
LSTM(Zaremba, Sutskever, and Vinyals 2014)	200	2	120.7	114.5	-	-	-	-
LSTM(Grave, Joulin, and Usunier 2016)	1024	1	-	82.3	1024	1	-	99.3
LSTM(Merity et al. 2017)	650	2	84.4	80.6	650	2	108.7	100.9
RNN†	256	1	130.3	124.1	512	1	126.0	120.4
LSTM†	256	1	118.6	110.3	512	1	105.6	101.4
TSLM	256	1	**117.2**	**108.1**	512	1	**104.9**	**100.4**
RNN+MoS†(Yang et al. 2018)	256	1	88.7	84.3	512	1	85.6	81.8
TSLM+MoS	256	1	**86.4**	**83.6**	512	1	**83.9**	**81.0**

Table 2: Best perplexity of models on the PTB and WikiText-2 dataset. Models tagged with † indicate that they are reimplemented by ourselves.

Experience

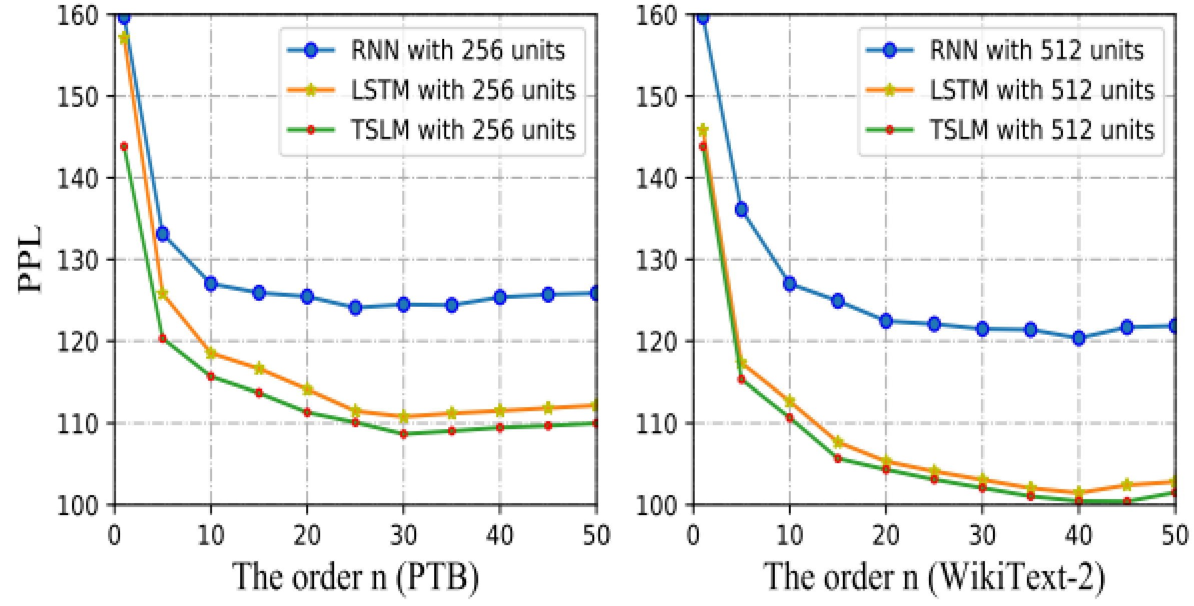


Figure 4: Perplexity (PPL) with different max length of sentences in corpus.

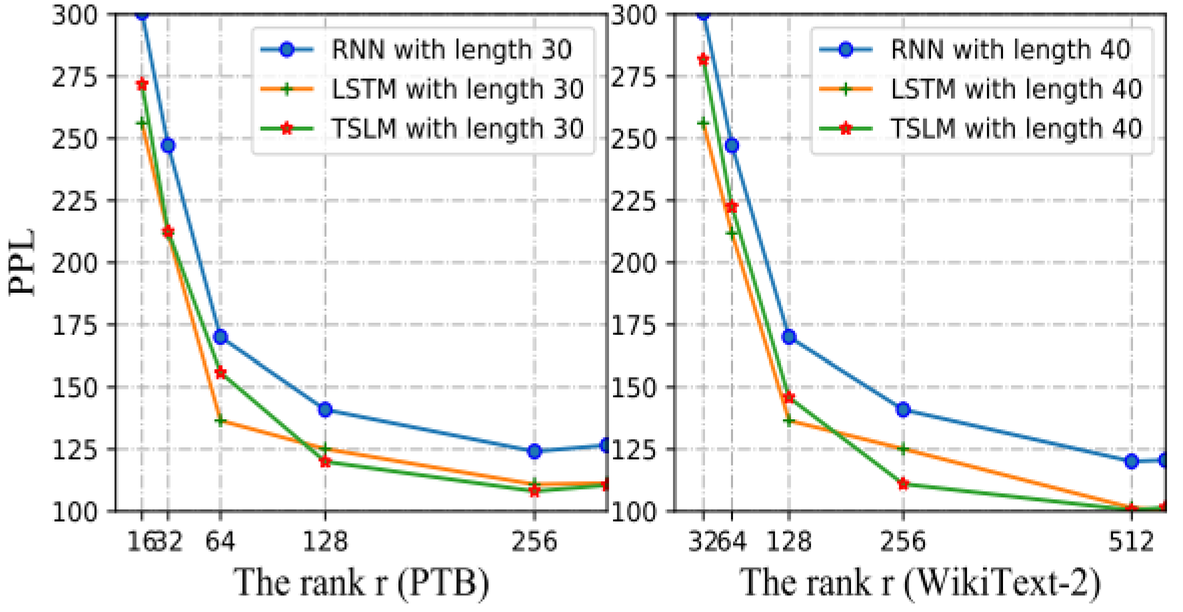


Figure 5: Perplexity (PPL) with different hidden sizes.

Future Work

- Achieve text generation by using TSLM
- Further interpreted in the neural network by tensor network
- Further explore the potential of tensor network for language model