A Generalized Language Model in Tensor Space

Tianjin University

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

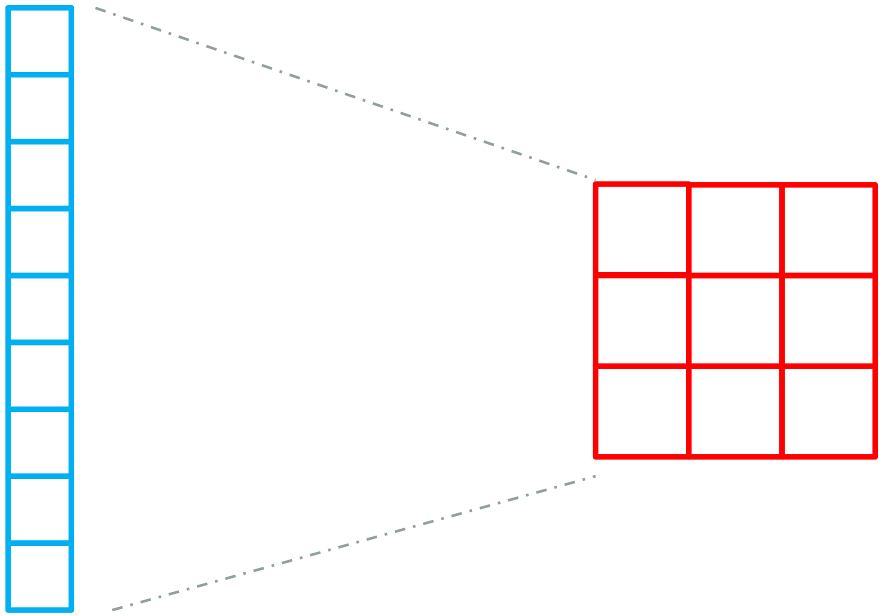
AAAI 2019

Outline

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

Motivation

- Represent the documents as the two order tensors:



For 10000-dimensional vector can be converted into 100×100

Cai, D.; He, X.; and Han, J. 2006. Tensor space model for document analysis. The 29th SIGIR conference on Research and development in information retrieval, 625–626

Motivation

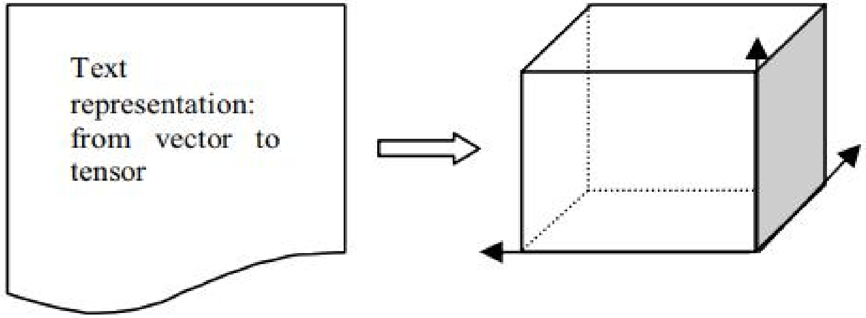


Figure 1. A document is represented as a character level 3-order tensor

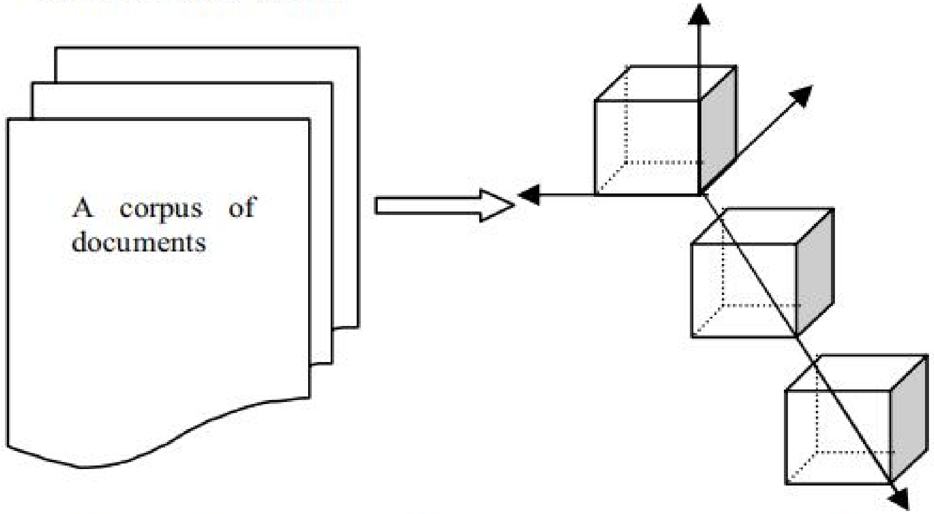


Figure 2. A corpus of documents is represented as a

Represent 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

We need a high-order tensor

- To construct a high-order tensor based language model

(Not limited to two/three consecutive words in 2/3-order tensor)

High tensor can consider all the combinatorial relations among words through the interaction among all the dimensions of word vectors.

We need a high-order tensor

- To derive an effective solution and demonstrate such a solution is a general approach for language modeling

(A high-order tensor contains exponential magnitude of parameters)

The Changing is that a high-order tensor contains exponential magnitude of parameters

Outline

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

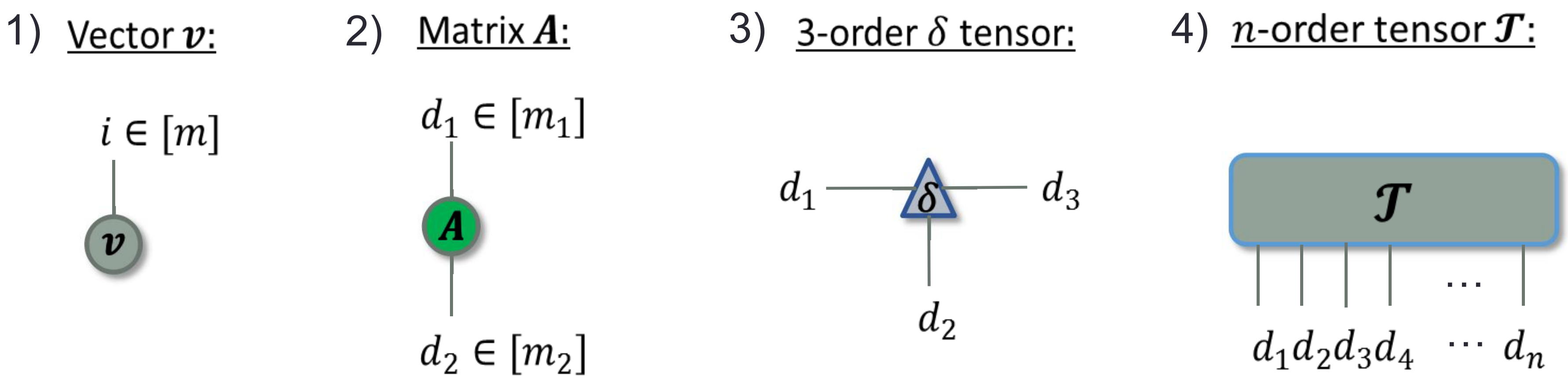
Background

- Tensor and Tensor Representation

  A tensor : a mutidimensional array

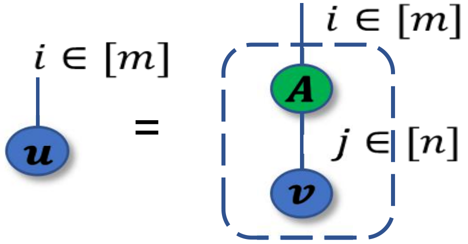
  The order : the number of indexing entries

  Tensor product : a fundamental operator



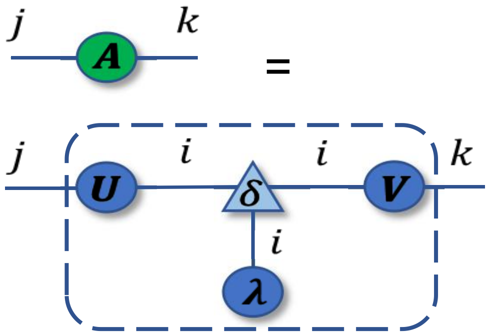
Tensor and Tensor Networks

- Tensor Network is formally represented an undirected and weight graph

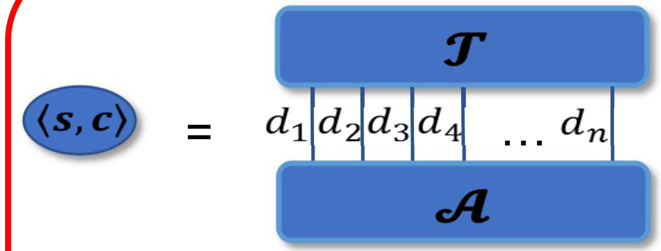


$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}$$

$$u_i = \sum_{j=1}^{n} A_{ij} v_j$$



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



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

Outline

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

TSLM basic representation

- How to represent a single word

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

- How to represent a original sentence

Rank One

$$\boxed{s = w_1 \otimes \cdots \otimes w_n} \qquad \mathcal{A}_{d_1,\cdots,d_n} = \prod_{i=1}^{n} \alpha_{i,d_i}$$

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

TSLM basic representation

- Assume that each sentence s_i appears with a probability p_i.

  We can denoted the corpus as:

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

  The sentence probability:

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

Outline

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

A Generation of N-Gram Language Mode

- N-gram Language Model
  - N-gram language model: estimate the probability distribution of sentences
  - Compute a sentence’s joint probability
  - Compute the current word’s conditional probability

How to Prove TSLM as a Generalization of N-Gram

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

$$c = \sum p_i s_i$$

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

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

Compute the joint probability

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

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

- Therefore, the probability of sentence

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

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 :

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

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

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

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

An example

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

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

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

$\mathcal{T}$

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

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

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

$\mathcal{A}$

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

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

- We define $p(w_1^i) = p(w_1, \cdots, w_i)$, in TSLM,
- $p(w_1^i) = \langle w_1^i, c \rangle =$
$\left\langle w_1 \otimes \cdots \otimes \mathbf{1}, \sum_{d_1 \cdots d_n = 1}^{|V|} \mathcal{T}_{d_1 \cdots d_n} w_{d_1} \otimes \cdots \otimes w_{d_n} \right\rangle$
- $= \sum_{d_{i+1} \cdots d_n = 1}^{|V|} \mathcal{T}_{d_1 \cdots d_n}, d_k = V.index(w_k)$
- We can compute the $p(w_1^{i-1}) = \langle w_1^{i-1}, c \rangle$, using the same approach

Outline

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

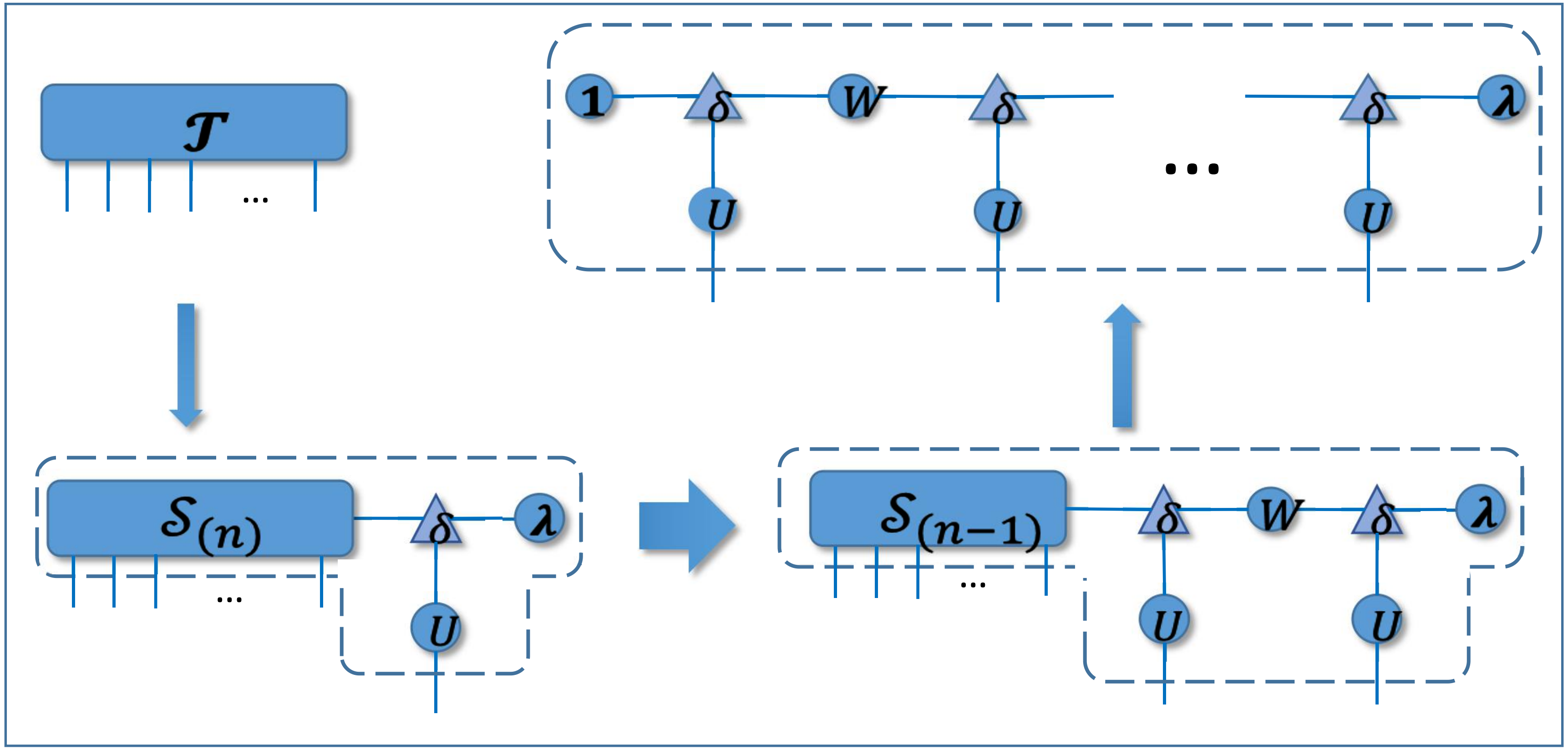
Recursive Language Modeling

- Two hypotheses
  - The dimensions of word vectors is $m \ll |V|$
  - The parameters are the same after each recursive SVD decomposition
  - The corpus : $c = \sum p_i |s_i\rangle$
- The formula of the recursive decomposition about tensor $\mathcal{T}$ is :

$$\mathcal{T} = \sum_{i=1}^{r} \lambda_i S_{(n),i} \otimes u_i$$

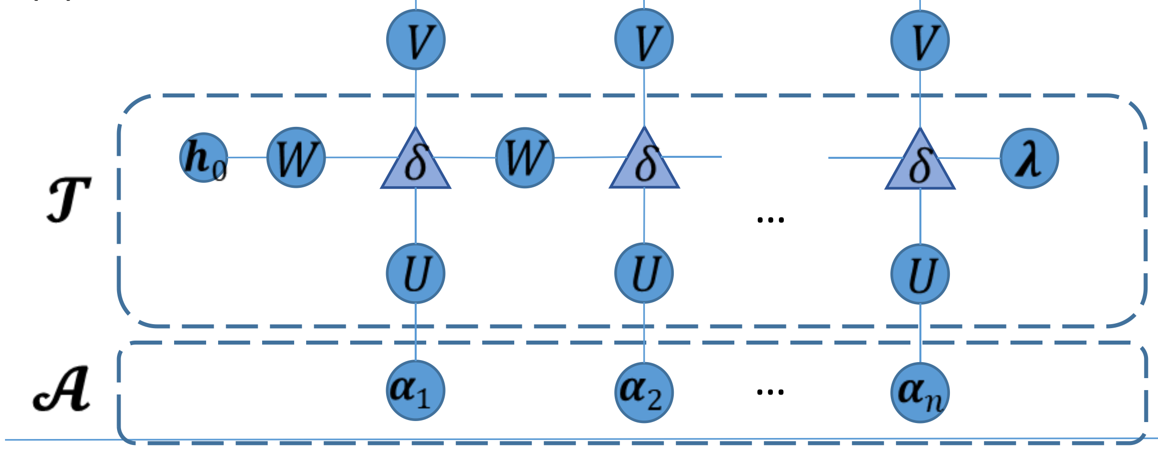
$$S_{(n),k} = \sum_{i=1}^{r} w_{k,i} S_{(n-1),i} \otimes u_i$$

Tensor recursive decomposition

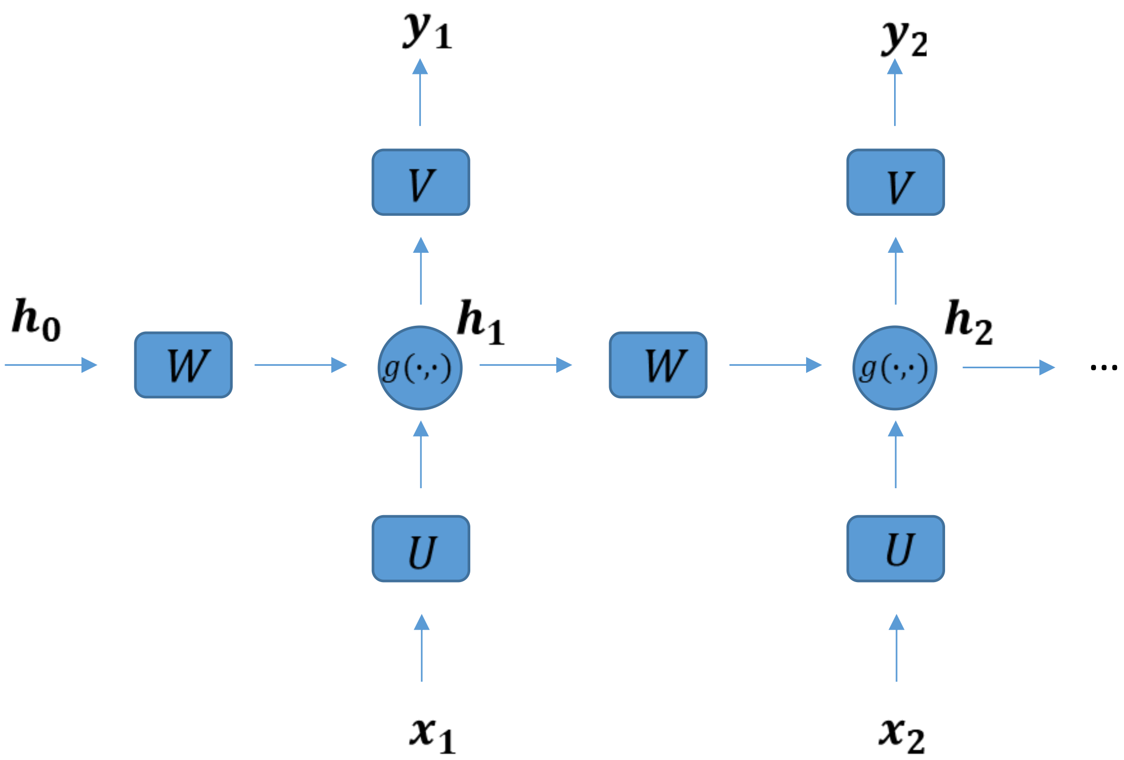


Recursive Language Modeling

(a)



$$h_t = g(\boldsymbol{a}, \boldsymbol{b})$$
$$\boldsymbol{a} = W h_{t-1}$$
$$\boldsymbol{b} = U x_t$$
$$g(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a} \odot \boldsymbol{b}$$



(d)

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

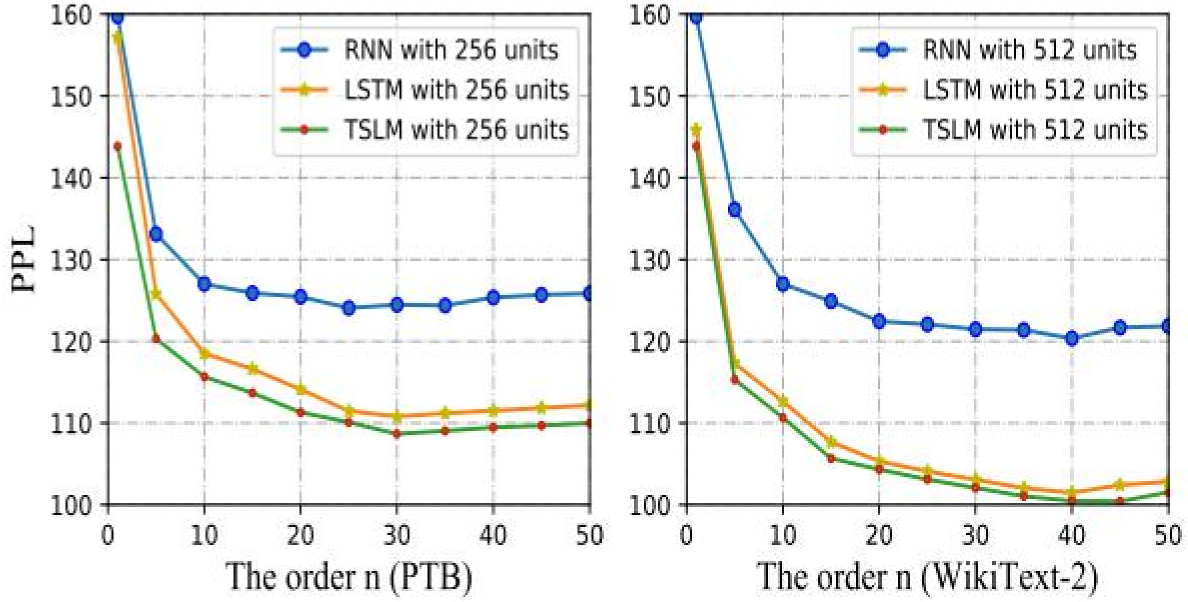


Figure 4: Perplexity (PPL) with different max length of sentences in corpus.

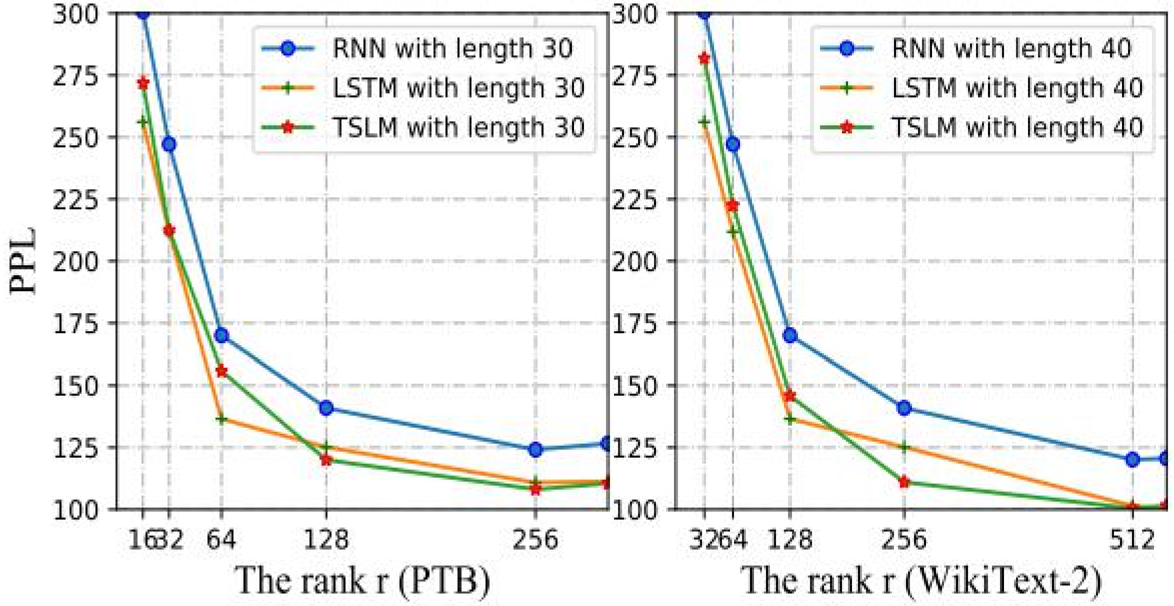


Figure 5: Perplexity (PPL) with different hidden sizes.

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