

# SHTM: A Neocortex-inspired Algorithm for One-shot Text Generation

Yuwei Wang, Yi Zeng, Bo Xu

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China

wangyuwei2014@ia.ac.cn, yi.zeng@ia.ac.cn, xubo@ia.ac.cn

**Abstract**—Text generation is a typical nature language processing task, and is the basis of machine translation and question answering. Deep learning techniques can get good performance on this task under the condition that huge number of parameters and mass of data are available for training. However, human beings do not learn in this way. People combine knowledge learned before and something new with only few samples. This process is called one-shot learning. In this paper, we propose a neocortex based computational model, Semantic Hierarchical Temporal Memory model (SHTM), for one-shot text generation. The model is refined from Hierarchical Temporal Memory model. LSTM is used for comparative study. Results on three public datasets show that SHTM performs much better than LSTM on the measures of mean precision and BLEU score. In addition, we utilize SHTM model to do question answering in the fashion of text generation and verifying its superiority.

## I. INTRODUCTION

Text generation is a special task of sequence generation which is widely used in machine translation and question answering. Recent advances show that recurrent neural networks (RNNs) form an expressive model family for sequence tasks. They are powerful because of their high-dimensional hidden state with nonlinear transformation. Sutskever *et al.* spent five days on 8 high-end Graphics Processing Units for text generation task, because they have thousands of parameters to fine-tune and complex networks structure [1]. Zaremba *et al.* demonstrated the power and expressiveness of sequence-to-sequence Long-Short Term Memory (LSTM) and it still contains a lot of parameters [2].

Many of the leading approaches in machine learning are also data-hungry [3]. However, children can make meaningful generalizations via “one-shot learning”. Recent advances on one-shot learning are most on image processing, and they mainly consider the bayesian approaches. In this paper, we explore one-shot learning in the context of nature language processing task from the perspective of brain-inspired computational model.

Hierarchical Temporal Memory (HTM) is a brain-inspired model which is based on the architecture of neocortex and trying to model the process of how human brain handles

the information about vision, audio, behavior, thus leading to memory and prediction. Recently, HTM incorporates many recently discovered properties of pyramidal cells and active dendrites [4], [5]. The algorithms have been applied to many practical problems, including speech learning [6], anomaly detection [7], and online sequence learning [8].

The goal of this paper is to demonstrate the power of brain-inspired models such as HTM in one-shot learning. More specifically, we refine the original HTM model to the Semantic Hierarchical Temporal Memory (SHTM) model, and we validate this model in one-shot text generation task.

## II. ONE-SHOT LEARNING

There are many data-hungry machine learning algorithms, such as deep learning networks, which perform well in various pattern recognition tasks. They mainly learn from large-scale data iteration by iteration, to reduce the value of cost function. When the model need to learn something new, the algorithm has to learn new data and old data together from scratch. However, human beings do not learn in this way.

When children learns to name an object, speak or read, they always don't need to repeat new things over and over again. And with the accumulation of knowledge, they learn more efficiently [9]. Similar to the human learning process, one-shot learning focus on learning from few pattern samples once. During this procedure, one-shot learning models can take advantage of knowledge coming from previously learned information.

Many previous efforts focus on one-shot learning algorithm with traditional machine learning tasks. Instead of learning from the scratch, Li *et al.* proposed a Bayesian based algorithm taking advantage of knowledge coming from previously learned [10], [11]. The model is well used in object classifying task when only few training samples exist. Based on Bayesian Program Learning (BPL) framework, one-shot learning have also been explored for handwritten characters classification task. Lake *et al.* develops a Hierarchical Bayesian acoustic model on one-shot classification and generation tasks [12].

In this paper, we propose a brain-inspired model, Semantic Hierarchical Temporal Memory (SHTM), that can deal with one-shot text generating task. Our model learns from a few

examples, and then can generate text almost the same as the original version. In addition, the learning process in the model is an online learning process, which is human-like and will reduce computing resource a lot.

### III. HIERARCHICAL TEMPORAL MEMORY

Hierarchical Temporal Memory (HTM) is a brain-inspired machine learning framework which was firstly proposed by Jeff Hawkins [13]. On the macro level, HTM imitates the neocortex to recognize patterns, and on the micro level, it is inspired by the activity of synapses and active dendrites.

Neocortex can deal with different kinds of information at the same time. When people learn something new, they often combine knowledge in memory learned before and the prediction based on the current pattern. HTM is a kind of neocortex-inspired model which makes use of the Memory-Prediction framework and imitates activities of cells in the neocortex. In this section, we will show several biological background and HTM model in detail.

#### A. Biological background

The human neocortex is a sheet of neural tissue approximately 2mm thick and its size is around 1,000 cm<sup>2</sup> [13]. Almost all the high level cognitive functions are associated with the neocortex. In this section, we will discuss several biological mechanisms which are from the neocortex to support the theory of HTM.

1) *Cell columns in cortex*: Based on the optical and extracellular recordings of macaque inferotemporal (IT) cortex, Tsunoda *et al.* proposed that a complex object is represented by combinations of active and inactive cell columns in anterior part of the IT cortex [14]. Each cell in a column shares the same receptive field, however, the state of cells can be different in the same column. Patterns will cause highly sparse responses in a cell column. Experiments show these phenomena in cat cortex [15], [16].

In HTM, the state of cell columns are used to represent the pattern roughly, and the states of cells in the same column are used to represent different temporal contents of the same pattern. For example, the same word *express* in sentence *It is hard for him to express himself in English.* and *He sent the book to me by express.* have different meanings. HTM will encode *express* with the same active columns (which is also called Sparse Distributed Representation, SDR), however, cells in these columns show different states, depending on the context of the word.

2) *Neurons*: Human neocortex is involved in cognitive functions such as sensory perception, generation of motor commands, reasoning, consciousness, and language, etc [17]. Pyramidal cells, stellate cells and granule cells are three of the most common types of cells in the neocortex. Neurons connect to each other to form a neural network to represent, store, transform and deal with patterns.

Inspired by the properties of neurons in the neocortex, HTM researchers summarize that there are three kinds of sources of synaptic input to cortical neurons, namely, proximal

zone, basal zone and apical zone [13]. The proximal zone receives feedforward input, the basal zone receives contextual input, mostly from nearby cells in the same cortical region while the apical zone invokes a top-down expectation [13].

3) *Distal dendrites*: In the human brain, neocortical neurons are cells which carry information in the cortex. A neuron is composed of cell body (i.e. soma), axon and dendrites. Axons transmit signals, starting at soma and terminating at points where the axon makes synaptic contact with target cells. While the soma receive signals from the synapses aligned along the dendrites which feed to it [13]. Multiple active synapses on proximal dendrites have roughly linear additive effect at the soma. Based on this mechanism, artificial neural networks make great progress in pattern recognition tasks.

However, a majority of synapses are distal, far from soma, also have little effect on the neuron responses [18]. Antic *et al.* proposed that NMDA (N-methyl-D-aspartate) spikes are likely to play significant roles in cortical information processing in awake animals and during slow-wave sleep [20]. Within close spatial and temporal proximity, the signal of several distal dendrites can lead to a local dendritic NMDA spike and bring the soma into a sustained depolarization state [19].

Inspired by some research on pyramidal neurons, in the HTM model, the patterns recognized by a neuron's distal synapses are used for prediction [13]. This is the basis of its Memory-Prediction framework. Instead of directly causing an action potential, the recognition of any one of these learned patterns acts as a prediction by depolarizing the cell [13].

#### B. HTM model

Taking these biological mechanisms into consideration, Hierarchical Temporal Memory (HTM) was proposed as a kind of pattern recognition model [13]. In this section, we will introduce several details of HTM, which consists of two stages, spatial pooling (SP) and temporal pooling (TP). SP aims to convert the origin pattern input into SDR, imitating the combination of cell columns to represent the pattern. While TP makes use of SDRs and sequence information, to get the output state and predicted state of cells at the current time.

1) *Spatial Pooling*: We define  $\mathbf{x}_i$  as the original input, where  $i = 1, 2, \dots, n$ ,  $n$  denotes the number pattern types. Let the binary vector  $\mathbf{SDR}_i$  be the SDR of the  $i$ th pattern, imitating the states of the cell columns.  $\mathbf{W}$  as the binary weight matrix, and the permanence represents the degree of connection. Each entry  $w_{pq}$  represents whether the  $p$ th dimension of original input and the  $q$ th columns are connected, where  $p = 1, 2, \dots, d$  and  $q = 1, 2, \dots, D$ . Here,  $D$  is the number of columns while  $d$  depends on the dimension of origin pattern inputs [13], [21]. SDRs can be generated as follows [21]:

$$w_{pq} = \begin{cases} 1 & \text{if } \text{permanence}_{pq} > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathbf{x}_i \cdot \mathbf{W} = \mathbf{SDR}_i \quad (2)$$

2) *Temporal Pooling*: We denote  $S_i^t$  as the active columns set of the  $i$ th pattern in the form of SDR.  $A^t$  and  $\Pi^t$  represent the current state and predicted state of cells at the time  $t$ . They are both  $D \times C$  dimension matrixes, and  $a_{qc}$  and  $\pi_{qc}$  are their inner element. Each cell in the HTM region has many segments, with different permanence to imitate the effect of distal dendrites. Permanence of different segments represent different degree of connection with other cells. We use a  $D \times C$  matrix  $D_{qc}^k$  to denote the permanence of  $k$ 'th segment of the  $c$ 'th cell in the  $q$ 'th column and use a binary matrix  $\tilde{D}_{qc}^k$  to denote only the connected synapses which means that the number of active connected synapses in a segment is greater than the activation threshold [13].

As the following equations shows, a cell in a winning column becoming active if it was in a predictive state during the preceding time step. If none of the cells in a winning column are in a predictive state, all cells in that column become active [21].

$$a_{qc}^t = \begin{cases} 1 & \text{if } c \in S_i^t \text{ and } \pi_{qc}^{t-1} = 1 \\ 1 & \text{if } c \in S_i^t \text{ and } \sum_c \pi_{qc}^{t-1} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

While for the predicted matrix  $\Pi^t$ , HTM concerns mainly the number of connected segments of the current active cells. If there exist one dendritic segment receiving enough input, it becomes active and subsequently depolarizes the cell body without causing an immediate spike [21].

$$\pi_{qc}^t = \begin{cases} 1 & \text{if } \exists_{den} \|\tilde{\mathbf{D}}_{qc}^{den} \circ \mathbf{A}^t\|_1 > \text{minThreshold} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

3) *Learning Rules*: The learning rules of HTM is based on Hebbian-like rules, details are shown in [21].

#### IV. SHTM BASED LANGUAGE MODEL FOR SEQUENCE GENERATION

HTM is an online learning algorithm, SP and TP can deal with natural signals via its Memory-Prediction mechanism. Unlike natural signals such as image, speech signal and digital sequence, nature language is more abstract to be represented in the form of SDR, which is friendly to HTM.

Although based on any kind of common word representations, spatial pooling can convert each word into SDRs, it only focus on low repetition rate, and have not taken semantic information into consideration. While especially for human language processing, we need to realize the fact that we do sequence generation with semantic word representations. So we take semantic information into consideration and propose the SHTM model for sequence generation task. Then we utilize the predicted cells in HTM model to get a predicted word, making it possible to generate text continually.

##### A. Word representation

There are many kinds of semantic word embeddings representing the meaning of a word via a vector. They make machine better understand human language. The common

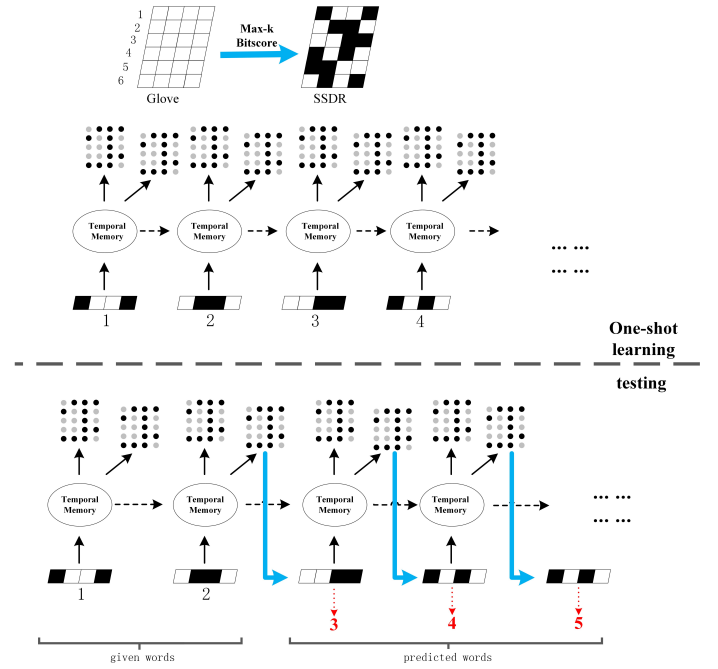


Fig. 1: The Flow Chart of SHTM for one-shot text generation

methods to get these word representations are latent semantic analysis, topic models, matrix factorization and neural networks.

GloVe, is a kind of word representation, which utilizes the benefit of global word-word co-occurrence statistics from a corpus, while simultaneously capturing the meaningful linear substructures prevalent in log-bilinear prediction-based methods like word2vec [22]. It is an unsupervised learning algorithm of word representations that implies semantic information [23]. Because of good results of GloVe in many nature language processing tasks, we consider using GloVe to represent the original semantic input of words.

In this paper, we make use of the GloVe embeddings trained by Pennington on 6 billion tokens of Wikipedia 2014 and Gigaword 5 as our word representation. And then, we convert dense GloVe vectors into binary word embeddings—Semantic Sparse Distributed Representation(SSDR), which is biologically plausible and friendly to HTM model.

##### B. Semantic Sparse Distributed Representation

In HTM model, SDR is the basic data structure, imitating the pattern representation in cortex which is with excellent property of robustness. However, SDRs do not take the abstract meaning of patterns into consideration, just to distinguish them. SSDRs are generated based on the semantic word embeddings, and details are as follows:

In the semantic word embeddings matrix  $W^{GloVe}$ , each dimension has special meaning in semantic space. What we want to do is to transfer the dense GloVe matrix into binary SSDR matrix, making semantic loss as little as possible. We firstly denote two factors Bit Importance(BI) and Bit

Discrimination(BD) for each bit of each word in SSDR space.

$$BI_{wb} = \frac{W_{wb}^{GloVe}}{\sum_w W_{wb}^{GloVe}}, \quad BD_{wb} = \left| \frac{W_{wb}^{GloVe} - \mu_b}{\sigma_b} \right|$$

$$\text{where } \mu_b = \frac{\sum_w W_{wb}^{GloVe}}{|W|}, \quad \sigma_b = \sqrt{\frac{\sum_w (W_{wb}^{GloVe} - \mu_b)^2}{|W|}} \quad (5)$$

These two factors respectively represent the importance of a bit in the same word and the discrimination in the range of all the words. Assuming the factors of  $w$ th word's  $b$ th bit in SSDR space are  $BI_{wb}$  and  $BD_{wb}$ , where  $w = 1, 2, \dots, |W|$ ,  $b = 1, 2, \dots, |B|$  and  $|W|$  is the dimensions of original GloVe and  $|B|$  is the dimensions of SSDR.

Then, we take an overall consideration, defining the Bit Score of each bit. For each word, we sort the bit scores and convert the max- $k$  bits into 1 and 0 otherwise.

$$BitScore_{wb} = BI_{wb} \cdot BD_{wb} \quad (6)$$

$$SSDR_{wb} = \mathbb{I}_{max-k \text{ subset}}(BitScore_{wb}) \quad (7)$$

SSDRs are binary word embeddings which not only combine semantic information to word representation, but also are with low dimension, unlike the one-hot representation of words. In addition, SSDRs imitate information storage way in neocortex, which is friendly to temporal pooling in HTM framework.

### C. Reducing

During the period of temporal pooling, each word in the sequence is passed to the temporal memory. At the time  $t$ , temporal memory merge the information from the present word's SSDR and the previous predicted cell states  $\Pi^{t-1}$  to generate the present cell states  $A^t$  and the next predicted cell states  $\Pi^{t+1}$  during TP. In this fashion, we can generate the sequence following when several beginning words are given.

We call the operation "Reducing", indicating that it converts the predicted cell states into next word. We scan all the cell columns, if any cell in some columns are active, we denote the bit of the predicted SSDR (pSSDR) as 1, and otherwise 0. However, the semantic pSSDR does not exactly represent the next word, we match the pSSDR with the most similar SSDR among all the learned words. And then based on the matched SSDR, we can generate the following text sequence in this form.

$$\mathbf{pSSDR}^{t+1} = \text{sign}(\mathbf{1}_{1 \times t} \pi_{qc}^t) \quad (8)$$

$$SSDR^{t+1} = \arg\max_i (W_i^{GloVe} \cdot \mathbf{pSSDR}^{t+1}) \quad (9)$$

In addition, SHTM is an online learning algorithm. Namely, during the testing period, the cell states in the model can also be changed, SHTM can learn new things even when we are testing it. When operating "Reducing", the cell states are also tend to adapt to the sequence better. This is a brain-inspired memory mechanism that really have a positive effect on one-shot learning tasks.

### D. Algorithm flow

To summarize, SHTM firstly transforms words in the sequence into semantic binary representation SSDRs, and put these SSDRs into the temporal pooling stage of HTM continuously. As the illustrative example shown in Fig 1, SHTM learn the sequence "1,2,3,4,5" once, and try to generate the sequence given part of it. Here, we give "1,2,3", let SHTM generate "4,5" in succession. During training, SHTM learns the connection weight and the state of the cell which is suitable to predict the next word. While during the test stage, via temporal pooling of HTM, we can get the predicted state of cell given part of the sequence "1,2,3". And then SHTM will conduct the "Reducing" operation to get the current output "4" and predicted SSDR as the follow-up "input" to generate next number.

## V. DATASETS AND EXPERIMENTAL SETUP

### A. Datasets

We validate our algorithms on three public text datasets, namely, Inaugural Address Corpus, Search Snippets and WikiQA. The summary statistics of the datasets are described in Table I.

1) *Inaugural Address Corpus*: The Inaugural address corpus include US presidential inaugural addresses from 1789 to 2009.

2) *Search Snippets*: The Search Snippets is a short text dataset collected by Phan [24], which is selected from the results of Web search transaction using predefined phrases of 8 different domains. We further delete the stop words and filter the sentences of its training part whose length is great than 15.

3) *WikiQA*: WikiQA is a dataset for open domain question answering, which is constructed in a natural and realistic manner [25]. In order to test our model reasonably, we use the training part of the dataset. When one question has multiple answers, we just select one of them randomly as the 'true answer'.

TABLE I: Statistics of the Datasets

Dataset	Vocab size	Coverage of GloVe	AvgLen	SenNum
Inaugural	14494	54.94%	28.38	4823
Snippets	23545	71.96%	19.63	7839
WikiQA	6782	95.50%	36.79	873

### B. Long-Short Temporal Memory

Long Short-term Memory (LSTM) is an RNN architecture model designed to be better at storing and accessing information than standard RNNs [26], which have many memory cells containing input gate, forget gate and output gate in its hidden layer. LSTM and its variants have recently get state-of-the-art results in a variety of natural language processing tasks, including speech, handwriting recognition, character generation and machine translation.

In this paper, we test LSTM on one-shot learning task as a comparative study. We utilize the GloVe embeddings as the input of every word, encode the given sequence with LSTM cells and decode the next word with softmax function. No matter for training or test stage, we both set the sliding window as one, making it possible to generate words one by one continuously.

### C. Experiments Details

We test both LSTM and our SHTM model on text generation and question answering tasks. Different kinds of datasets are used, and experiment details are as follows.

1) *Text Generation Task*: Text generation is the traditional NLP task, and it is the basis of language model, machine translation and automatic summarization. We test both our SHTM model and LSTM model on one-shot text generation task with first two datasets. For every lecture in the Inaugural Address Corpus, we scan it at a time during training and select sentences in that lecture that contains more than 15 words. And for snippets search dataset, we scan each snippet at a time. In addition, both models are trained only once.

2) *Question Answering Task*: Question answering (QA) aims to automatically answer questions posed by humans in natural language. In our experiment, we not regard the QA task as the sentence selection task, which are looking for the most suitable sentence in the database according to the question. We do question answering in a text generation way. We join the question and answer pair as a sequence, both models learn the new sequence and try to generate the ‘answer’ when asking the model questions. WikiQA corpus are used in the QA task.

3) *Parameters setup*: For SHTM model, we set the number of columns  $D$  equals to the dimensions of SSDR, while the number of cells per column is 5. Dendritic segment activation threshold and minThreshold are 10 and 8 respectively, while for SSDR of 200 dimensions are 15 and 12. Initial synaptic permanence is 0.5 and synaptic permanence increment  $p^+$  and synaptic permanence decrement  $p^-$  are both 0.1. Synaptic permanence decrement for predicted inactive segments is 0.01.

For LSTM model, LSTM cells are 256 and 64 dimensions, while SSDR are 200 and 50 dimensions. Weights are scaled uniform initialization [27], the loss function is multi-class logloss and optimizing parameters in the way of RMSProp.

4) *Results and Analysis*: Both SHTM and LSTM models learn texts with 50 and 200 dimensions word representations respectively. In text generation task, we test the models under two kinds of situation, given 5 words to generate 10 words and given 7 words to generate 8 words. We use mean precision (mP@(%)) to measure the precision of generated text or generated answer, and use BLEU to measure the fluency of the result. The detailed results are show in Tabel II.

As shown in Table II, it is obvious that, both precision and BLEU, SHTM is significantly better than LSTM. The high accuracy of generated words and the fluency of sentence validates the effectiveness of SHTM. This indicates that the semantic representation SSDR and SHTM make sense during one-shot learning. Only learning the patterns once, SHTM

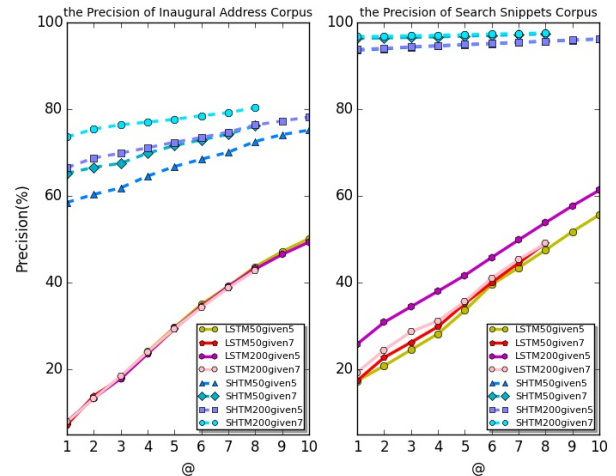


Fig. 2: The mean Precision Results of the Two Datasets

can generate text sequence well. SHTM deserves the results because of its brain-inspired memory mechanism. While for LSTM, the result relates to the initial weight in their network, because it just update their network once, far away from its optima.

In the QA experiment, due to the length of answers are different, we only measure the model with BLEU score. And results are shown in Table III. We can also illustrate the results via the two examples below.

**Question:** what is honey bee propolis?

**Answer:** Propolis as hive sealing

**SHTM:** propolis as hive sealing

**LSTM:** what sealing propolis hive

**Question:** what is grist mill stone?

**Answer:** A gristmill ( also : grist mill , corn mill or flour mill ) grinds grain into flour .

**SHTM:** . grist mill , corn mill or flour mill ) grinds grain into flour . grist mill , corn mill

**LSTM :** flour flour mill mill into gristmill also also also or stone grist mill mill grinds gristmill

TABLE III: BLEU Score Results of WikiQA Dataset

-	50dim	200dim
LSTM	0.0001906	0.0004354
SHTM	0.3438518	0.3029991

## VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a brain-inspired model SHTM, based on the traditional HTM model, to deal with one-shot text

TABLE II: Mean Precision and BLEU Results of Inaugural Address and Search Snippets Corpus

Methods	Inaugural Address Corpus					Search Snippets Corpus				
	P@1	P@3	P@5	P@10	BLEU	P@1	P@3	P@5	P@10	BLEU
SHTM50given5	58.51%	61.87%	66.79%	75.18%	0.2789	93.63%	94.32%	94.87%	96.17%	0.7050
SHTM50given7	65.23%	67.54%	71.63%	-	0.3175	96.35%	96.56%	96.87%	-	0.6787
SHTM200given5	66.53%	69.86%	72.39%	78.20%	0.2437	93.80%	94.42%	94.98%	96.26%	0.6985
SHTM200given7	73.60%	75.42%	77.68%	-	0.2797	96.74%	96.94%	97.21%	-	0.6806
LSTM50given5	7.75%	18.31%	29.71%	50.16%	$3.48 \times 10^{-7}$	17.36%	24.54%	33.61%	55.66%	$6.13 \times 10^{-4}$
LSTM50given7	7.23%	17.92%	29.32%	-	$2.68 \times 10^{-7}$	17.45%	26.26%	35.22%	-	$8.92 \times 10^{-5}$
LSTM200given5	7.98%	17.93%	29.53%	49.36%	$4.53 \times 10^{-7}$	25.88%	34.48%	41.71%	61.34%	$1.24 \times 10^{-3}$
LSTM200given7	7.88%	18.47%	29.43%	-	$9.03 \times 10^{-7}$	19.34%	28.70%	35.65	-	$9.99 \times 10^{-4}$

generation problem. SHTM is an online learning algorithm that makes it possible to imitate human learning process in one-shot text generation task. The model is not data-hungry and it does not have too much parameters to fine-tuning. Its Memory-Prediction framework and its concise learning rules guarantee its generalization.

In the future, we will focus on the following directions for improvements: (1) introducing more biological mechanism which is useful to improve our model, and (2) improving the semantic word embeddings that is friendly to HTM. In addition, how to generate SHTM to other NLP tasks is still an ongoing research.

#### ACKNOWLEDGMENT

This study was funded by the Strategic Priority Research Program of the Chinese Academy of Sciences (XD-B02060007), and Beijing Municipal Commission of Science and Technology (Z151100000915070, Z161100000216124).

#### REFERENCES

- [1] Sutskever I, Martens J, Hinton G E. Generating text with recurrent neural networks. //Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011: 1017-1024.
- [2] Zaremba W, Sutskever I. Learning to execute. arXiv preprint arXiv:1410.4615, 2014.
- [3] Lake B M, Salakhutdinov R, Tenenbaum J B. Human-level concept learning through probabilistic program induction. *Science*, 2015, 350(6266): 1332-1338.
- [4] Antic S D, Zhou W L, Moore A R, et al. The decade of the dendritic NMDA spike. *Journal of neuroscience research*, 2010, 88(14): 2991-3001.
- [5] Major G, Larkum M E, Schiller J. Active properties of neocortical pyramidal neuron dendrites. *Annual review of neuroscience*, 2013, 36: 1-24.
- [6] van Doremalen J, Boves L. Spoken digit recognition using a hierarchical temporal memory. //INTERSPEECH. 2008: 2566-2569.
- [7] Lavin A, Ahmad S. Evaluating real-time anomaly detection algorithms-the numenta anomaly benchmark. arXiv preprint arXiv:1510.03336, 2015.
- [8] Cui Y, Surpur C, Ahmad S, et al. Continuous online sequence learning with an unsupervised neural network model. arXiv preprint arXiv:1512.05463, 2015.
- [9] Yip K, Sussman G J. Sparse representations for fast, one-shot learning[J]. 1997.
- [10] Li F F, Fergus R, Perona P. A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories// null. IEEE Computer Society, 2003:1134.
- [11] Li F F, Rob F, Pietro P. One-shot learning of object categories. *Pattern Analysis & Machine Intelligence IEEE Transactions on*, 2006, 28(4):594-611.
- [12] Lake B M, Lee C, Glass J R, et al. One-shot learning of generative speech concepts. //Proceedings of the 36th Annual Meeting of the Cognitive Science Society. 2014.
- [13] Hawkins, Jeff, Subutai Ahmad, and D. Dubinsky. "Hierarchical temporal memoryincluding HTM cortical learning algorithms." Technical report, Numenta, Inc, Palo Alto <http://numenta.org/cla-white-paper.html>, 2010.
- [14] Tsunoda K, Yamane Y, Nishizaki M, et al. Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns. *Nature neuroscience*, 2001, 4(8): 832-838.
- [15] Yen S C, Baker J, Gray C M. Heterogeneity in the responses of adjacent neurons to natural stimuli in cat striate cortex[J]. *Journal of neurophysiology*, 2007, 97(2): 1326-1341.
- [16] Martin K A C, Sylvia S. Functional heterogeneity in neighboring neurons of cat primary visual cortex in response to both artificial and natural stimuli. *Journal of Neuroscience*, 2013, 33(17):7325-44.
- [17] Lui J H, Hansen D V, Kriegstein A R. Development and evolution of the human neocortex. *Cell*, 2011, 146(1): 18-36.
- [18] Major G, Larkum M E, Schiller J. Active properties of neocortical pyramidal neuron dendrites. *Annual review of neuroscience*, 2013, 36: 1-24.
- [19] Elston G N. Cortex, cognition and the cell: new insights into the pyramidal neuron and prefrontal function. *Cerebral Cortex*, 2003, 13(11): 1124-1138.
- [20] Antic S D, Zhou W L, Moore A R, et al. The decade of the dendritic NMDA spike. *Journal of neuroscience research*, 2010, 88(14): 2991-3001.
- [21] Hawkins J, Ahmad S. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. arXiv preprint arXiv:1511.00083, 2015.
- [22] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. //Advances in neural information processing systems. 2013: 3111-3119.
- [23] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation. // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.
- [24] Phan X H, Nguyen L M, Horiguchi S. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. //Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 91-100.
- [25] Yang Y, Yih W, Meek C. WIKIQA: A Challenge Dataset for Open-Domain Question Answering. //Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015: 2013-2018.
- [26] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural computation*, 1997, 9(8): 1735-1780.
- [27] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. //International conference on artificial intelligence and statistics. 2010: 249-256.
- [28] Ahmad S, Hawkins J. Properties of sparse distributed representations and their application to hierarchical temporal memory[J]. arXiv preprint arXiv:1503.07469, 2015.