Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 123 (2018) 446–450

Procedia Computer Science

www.elsevier.com/locate/procedia

8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017

Semantic-Map-Based Assistant for Creative Text Generation

Andrey Y. Shedko

National Research Nuclear University "Moscow Engineering Physics Institute"
Kashirskoe Shosse 31, Moscow, 115409, Russian Federation
ayshedko@yandex.ru

Abstract

A weak semantic map of English words is used here as an intuitive interface to a pattern generator, which produces paragraphs of text in the style of a specific writer. For this purpose, a recurrent neural net based on the Long Short-Term Memory (LSTM) model is used. The objective is to generate a poem in the style of Byron. The result is a guided generation of sequences of words, based on the correspondence between latent neural network representations and the semantic map. Future applications of this technique in the form of cognitive assistants are discussed, including an automated poetry-writing assistant.

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures

Keywords: personal assistant; cognitive modeling; semantic search; LSTM; text prediction

1 Introduction

Semantic maps, or semantic spaces, are widely used today in various domains, from preference handling and semantic search to brain mapping and cognitive architecture design [1]. Semantic maps of natural language are not the only, but the most known type. They are constructed either with the help of human subjects (for example performing word ranking or providing psychometrics) or through automatic processing of text corpora. In this paper we shall focus on using a semantic map to control a pattern generator, specifically, an LSTM-based neural network trained on some specific text [2].

The idea of a semantic map is to allocate meaningful representations, e.g., words and phrases of natural language, in some abstract space, so that the geometric and topological relations among the allocation points would reflect semantic relations among the allocated items. In general, there are two properties that can be achieved in this way: (1) a map represents semantic dissimilarity by geometric distance (dissimilarity metrics), or (2) map coordinates have definite semantic interpretations. Combining (1) and (2) in one and the same semantic map remains problematic. Maps possessing properties (1) and (2) are called strong and weak semantic maps, respectively [3]. Here we use a weak semantic map because of its simplicity and accessibility.

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures
10.1016/j.procs.2018.01.068

2 Conceptual Model

Let us describe the basic operation of an LSTM[4] pattern generator (Figure 1). An LSTM block has mechanisms to enable "memorizing" information for an extended number of time steps. We use the LSTM block with the following transformations that map inputs to outputs across blocks at consecutive layers and consecutive time steps:

$$g_t = tanh(X_t W_{xg} + h_t - 1W_{hg} + b_g),$$
$$i_t = \sigma(X_t W_{xi} + h_{t-1} W_{hi} + b_i),$$
$$f_t = \sigma(X_t W_{xf} + h_{t-1} W_{hf} + b_f),$$
$$o_t = \sigma(X_t W_{xo} + h_{t-1} W_{ho} + b_o),$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t,$$
$$h_t = o_t \odot tanh(c_t),$$

where $\odot$ is an element-wise multiplication operator, and $\forall x = [x1, x2, \ldots, xk]^T \in R^k$ the two activation functions:

$$\boldsymbol{\sigma(x)} = \left[\frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{exp}(-\boldsymbol{x}\mathbf{1})}, \cdots \frac{,\mathbf{1}}{\mathbf{1} + \boldsymbol{exp}(-\boldsymbol{xk})}\right]^{\boldsymbol{T}},$$
$$\boldsymbol{tanh(x)} = \left[\frac{\mathbf{1} - \boldsymbol{exp}(-\mathbf{2}\boldsymbol{x}\mathbf{1})}{\mathbf{1} + \boldsymbol{exp}(-\mathbf{2}\boldsymbol{x}\mathbf{1})}, \ldots, \frac{\mathbf{1} - \boldsymbol{exp}(-\mathbf{2}\boldsymbol{xk})}{\mathbf{1} + \boldsymbol{exp}(-\mathbf{2}\boldsymbol{xk})}\right]^{\mathrm{T}}.$$

In the transformations above, the memory cell c_t stores the "long-term" memory in the vector form. In other words, the information accumulatively captured and encoded until time step *t* is stored in c_t and is only passed along the same layer over different time steps.

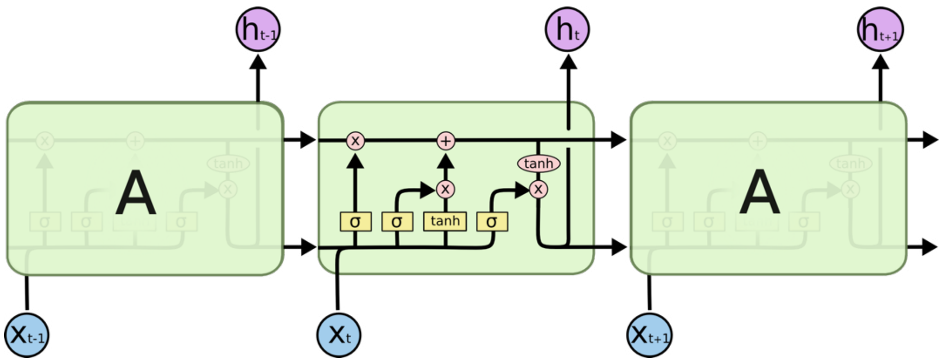

Figure 1. The LSTM neural network architecture (based on [4]).

Given the inputs c_t and h_t, the input gate it and forget gate f_t will help the memory cell to decide how to overwrite or keep the memory information. The output gate o_t further lets the LSTM block decide how to retrieve the memory information to generate the current state h_t that is passed to both the next layer of the current time step and the next time step of the current layer. Such decisions are made using the hidden-layer parameters *W* and *b* with different subscripts: these parameters will be inferred during the training phase by `MXNet` [5].