Neurocomputing 71 (2008) 3150–3157

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Modeling word perception using the Elman network

Cheng-Yuan Liou *, Jau-Chi Huang, Wen-Chie Yang

Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, ROC

ARTICLE INFO

Available online 24 June 2008

Keywords:
Word perception
Compositional representation
Authorship
Stylistic similarity
Categorization
Semantic search
Elman network
Linguistic analysis
Personalized code
Content addressable memory
Polysemous word

ABSTRACT

This paper presents an automatic acquisition process to acquire the semantic meaning for the words. This process obtains the representation vectors for stemmed words by iteratively improving the vectors, using a trained Elman network. Experiments performed on a corpus composed of Shakespeare's writings show its linguistic analysis and categorization abilities.

1. Introduction

The semantic meaning of a word or a word sequence is often nonquantifiable. A central problem in the analysis of such a sequence is determining how to effectively encode and extract its contents. Existing analyses are primarily based on certain statistical linguistic features [2,7,23–25]. The semantic search [26] constructs a mathematical model that analyzes semantic features and creates a semantic operation space. It sorts data according to the semantic meaning of the devolved requests. Nevertheless, there are difficulties in implementing the model. The task of constructing a prime semantic space is extremely expensive and complex, because experienced linguists are needed to analyze huge numbers of words. This paper presents an automatic encoding process to accomplish this task.

Both the frequencies and the temporal sequence of words carry semantic meaning. When one listens to a talk or reads an article, one should get information from both isolated words and their sequences. Complying with temporal information [18,16], our approach employs the Elman network [4,3,13], which works well with temporal sequences, as an encoding mechanism [15,17,12]. This network can extract and accommodate the rich syntax grammars associated with each word in sentence sequences [9].

The automatic encoding method will be presented in Section 2.1. The semantic search [26] and its notations will be reviewed in Section 2.2. A method for dealing with polysemous words will be discussed in the third section. Applications to literary works will be presented in these two sections.

2. Encoding method

Semantic meaning comes from a sequence of words. It is sequential and temporal. We employ the Elman network to extract the meaning from sentence sequences.

2.1. The Elman network

The network is a single recursive network that has a context layer as an inside self-referenced layer, see Fig. 1. During operation, both current input from the input layer and previous state of the hidden layer saved in the context layer activate the hidden layer. Note that there exists an energy function associated with the hidden layer, context layer, and input layer [16,12]. With successive training, the connection weights can load the temporal relations in the training word sequences.

The context layer carries the memory. The hidden layer activates the output layer and refreshes the context layer with the current state of the hidden layer. The back-propagation learning algorithm [21] is commonly employed to train the weights in order to reduce the difference between the output of the output layer and its desired output. Note that in this paper, the threshold value of every neuron in the network is set to zero. Let L_o, L_h, L_c, and L_i be the number of neurons in the output layer, the hidden layer, the context layer, and the input layer, respectively. In the Elman network, L_h is equal to L_c, that is, $L_h = L_c$. In this paper,

* Corresponding author.
E-mail address: cyliou@csie.ntu.edu.tw (C.-Y. Liou).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.04.030

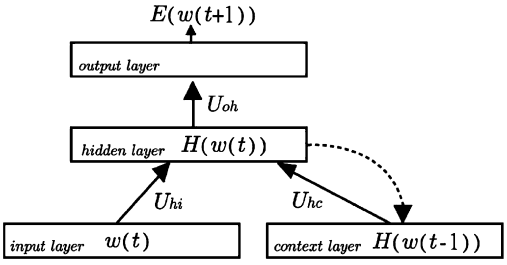

Fig. 1. The Elman network.

the number of neurons in the input layer is equal to that in the output layer and is also equal to the number of total features, that is, $R = L_o = L_i$.

Let $\{w_n, n = 1-N\}$ be the code set of different words in a corpus. The corpus, D, contains a collection of all given sentences. During training, a sentence is randomly selected from the corpus and fed to the network sequentially, word by word, starting from the first word of the sentence. Let $|D|$ be the total length of all the sentences in the corpus, D. $|D|$ is the total number of words in D. Usually, $|D|$ is several times the number of different words in the corpus, so $|D| > N$. Initially, $t = 0$, and all weights are set to small random numbers. Let $w(t)$ be the current word in a selected sentence at time t, i.e.,

$$w(t) \in D, \quad w(t) \in \{w_n, n = 1-N\}, \quad t = 1-T, \tag{1}$$

where $w(T)$ is the last word of a training epoch. In this paper, we set $T = 4|D|$ in one epoch. This means that in each epoch, we use all the sentences in the corpus to train the Elman network four times. Let the three weight matrices between layers be U_{oh}, U_{hc}, and U_{hi}, where U_{oh} is an L_h by L_o matrix, U_{hc} is an L_c by L_h matrix, and U_{hi} is an L_i by L_h matrix, as shown in Fig. 1. The output vector of the hidden layer is denoted as $H(w(t))$ when $w(t)$ is fed to the input layer. $H(w(t))$ is an L_h by 1 column vector with L_h elements. Let $E(w(t+1))$ be the output vector of the output layer when $w(t)$ is fed to the input layer. $E(w(t+1))$ is an L_o by 1 column vector.

The function of the network is

$$H(w(t)) = \varphi(U_{hi}w(t) + U_{hc}H(w(t-1)), \tag{2}$$

where φ is a sigmoid activation function that operates on each element of a vector [21]. We use the sigmoid function $\varphi(x) = 1.7159 * \tanh(x * 2/3)$ for all neurons in the network. This function gives a value roughly between $+1.7159$ and -1.7159. In Elman's experiment, the first step is to update the weights, U_{hi}, U_{hc} and U_{oh}, through training. The second step is to encode words with a tree structure. All the attempts are aimed at minimizing the error between the network outputs and the desired outputs to satisfy the prediction

$$w(t+1) \approx E(w(t+1)) = \varphi(U_{oh}H(w(t))). \tag{3}$$

From a trained network, Elman uses a measure to locate the relationships among words. Before training, he prepares a list of words without inflections or rules. We will follow his preparation on words. All words are coded with given lexical codes. The available semantic combination is a fixed syntax ($Noun + Verb + Noun$). Elman generates sentences and temporal word sequences with this syntax grammar and collects all the sentences in a training corpus, D, for training a network [3]. The network has equal numbers of neuron units in its four layers. This network is trained sequentially by using the generated sentences. Elman defines the desired outputs as the sufficient words. For example, when the first word 'man' in a generated sentence 'men sleep' is used as the input, the sufficient word 'sleep' is its desired output. The network is trained to predict the following word. This training process continues until the variation of weights cannot be reduced. After training, Elman inputs the generated sentences again and collects all the output vectors of the hidden layer corresponding to each individual word in a separate set, $s_n^E = \{H(w(t))|w(t) = w_n\}$. Then he obtains new code, w_n^E, for the nth word by averaging all vectors in set s_n^E:

$$w_n^E = \frac{1}{|s_n^E|} \sum_{\substack{w(t)=w_n \\ w(t)\in D}} H(w(t)), \quad n = 1-N, \tag{4}$$

where $|s_n^E|$ is the total number of vectors inside set s_n^E. Then, he constructs a tree for the words based on their new codes, w_n^E, to explore the relationships among the words.

Note that there exist extra temporal relations in the generated sentences with the simple fixed syntax $Noun + Verb + Noun$. For example, when $w(t)$ is a noun, $w(t+2)$ is most likely a noun, and when $w(t)$ is a verb, $w(t+3)$ is most likely a verb. These extra relations are additive to resolve the dichotomous classification between the verb and noun. A compound sentence may not possess such extra relations, and may not have additive resolutions.

2.2. The semantic search

The semantic search constructs a semantic model and a semantic measure. A manually designed semantic code set is used in the model. It assumes that the encoding task will be assigned to linguistics experts. It is hypothesized in advance that one can build a raw semantic matrix, W, as

$$W_{R\times N} \equiv [w_1 \;\; w_2 \;\; \ldots \;\; w_N]_{R\times N}, \tag{5}$$

where w_n, $n = 1-N$, denotes the code of the nth stemmed word and N denotes the total number of different words. A code of a word is a column vector with R features as its elements:

$$w_n \equiv [w_{1n}, w_{2n}, \ldots, w_{Rn}]^T. \tag{6}$$

To manage abstract features, one may use the orthogonal space configured by the characteristic decomposition of the matrix, WW^T:

$$W_{R\times N}W_{R\times N}^T = F_{R\times R}^T \begin{bmatrix} \lambda_1 & 0 & \cdot & 0 \\ 0 & \lambda_2 & 0 & \cdot \\ \cdot & 0 & \cdot & 0 \\ 0 & \cdot & 0 & \lambda_R \end{bmatrix}_{R\times R} F_{R\times R}, \tag{7}$$

where

$$F_{R\times R} \equiv [f_1, f_2, \ldots, f_R]_{R\times R}, \quad \|f_r\| = 1,$$

and

$$\lambda_r \geqslant \lambda_{r+1}, \quad r = 1-R. \tag{8}$$

Since WW^T is a symmetric matrix, all its eigenvalues are real and nonnegative numbers. Each eigenvalue λ_i equals the variance of the N projections of the codes on the ith eigenvector, f_i, that is, $\lambda_i = \sum_{n=1}^{N}(\langle w_n \cdot f_i\rangle)^2$.

2.2.1. Multi-dimensional scaling (MDS) space

We select a set of R^s eigenvectors, $\{f_r, r = 1-R^s\}$, from the R eigenvectors to build a reduced feature space:

$$F_{R\times R^s}^s \equiv [f_1, f_2, \ldots, f_{R^s}]_{R\times R^s}. \tag{9}$$

This selection is based on the distribution of the projections of the codes on each eigenvector. An ideal distribution is an even distribution with large variance. We select those eigenvectors, $\{f_r, r = 1-R^s\}$, that have large eigenvalues. The MDS space is

$$MDS \equiv span(F^s). \tag{10}$$