More on Richness in Semantic Meaning

Cheng-Yuan Liou[a], Jau-Chi Huang[a], Wen-Chie Yang[a]

[a]*Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, Republic of China*

Abstract

The values of richness in semantic meaning recorded in Table 3 in [1] are further discussed in this work. These values according to the writing time of Shakespeare can reveal the changes of writing style. The values for Mark Twin's works show similar changes.

Keywords: Elman network; syntax-based semantic space model; stylish analysis; computational linguistic

1. Review: the measure of richness in semantic meaning

The re-encoding method [5] developed a measure of richness in semantic meaning (RSM) and applied it to analyze Shakerspeare's 36 dramas. The 36 values of RSM recorded in 'Table 3 in [5]' reveal the evolution of writing style of Shakespear. This work provides a further discussion on those records. We briefly review the measure in this section and discuss the records in the next section. The RMS values for Mark Twain's works are also included in the next section.

1.1. Elman network

In [1], Elman network has two important achievements. One is that it can discover the underlying structure of word. The other is that it can discover lexical classes from word order. This network is a simple recurrent network that has a context layer as an inside self-referenced layer, see Fig. 1. It was designed to find the hidden structure of sequential inputs [1]. 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. The context layer in the network is a copy of the hidden layer at previous time step, $L_h = L_c$. The weights of self-referenced links which are from hidden neurons to context neurons are fixed to 1. During operation, at time step t, the output of hidden layer will be loaded to the context layer and together with the input layer to activate the hidden layer at time $t + 1$.

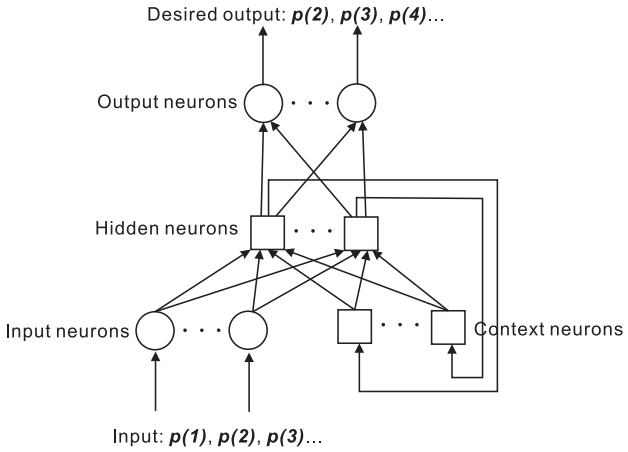


Figure 1: Illustration of Elman network

Let the two weight matrices between layers be W_{oh} and W_{hic}, where W_{oh} is an $L_h + 1$ by L_o matrix, and W_{hic} is an $L_i + L_c + 1$ by L_h matrix. Consider a sequence of words, $\{p(t), t = 1, 2, 3, ...\}$. The output vector of the hidden layer is denoted as $h(p(t))$ when $p(t)$ is fed to the input layer. $h(p(t))$ is an L_h by 1 column vector. Let $o(p(t))$ be the output vector of the output layer when $p(t)$ is fed to the input layer. $o(p(t))$ is an L_o by 1 column vector. The function of the hidden layer is $h(p(t)) = \varphi(W_{hic}^T[in(p(t))])$, where $in(p(t))$ is an $L_i + L_c + 1$ by 1 column vector and φ is a sigmoid activation function that operates on each element of a vector [2]. The $in(p(t))$ has the form, $in(p(t)) = [p(t),\ h(p(t-1)),\ 1]^T$. The function of the output layer is $o(p(t)) = \varphi(W_{oh}^T\left[h(p(t)),\ 1\right]^T\right])$.

The back-propagation algorithm [2] is commonly employed to train the weights, W_{oh} and W_{hic}, to reduce the difference between $o(p(t))$ and its desired output. What is the desired output of the input pattern $p(t)$? Elman used the network to model human language processing, and he chose the next input pattern $p(t + 1)$ as the desired output [1]. For example, consider a sequence of input patterns, $\{p(1), p(2), p(3), ...\}$. The input at time $t = 1$ is $p(1)$, and the desired output at time $t = 1$ is $p(2)$. All the attempts are aimed at minimizing the error between the network's outputs and the desired outputs, $\|o(p(t)) - p(t + 1)\|^2$, to satisfy the prediction $o(p(t)) \approx p(t + 1)$.

There are several reasons to choose prediction as the main task of the network [3]. When human brains process language, there is limited observation from the environment except the sequence of word events. The desired outputs are immediately available and require minimal priori theoretical analysis. On the other hand, prediction plays a role in language processing. Listeners indeed predict

and notice sequences of words which violate expectations. Moreover, the prediction task can explain why children won't get the faulty over-generalizations from positive data when they lack for negative evidence (wrong usage of words) in the process of language learning.

1.2. Iterative re-encoding

Based on Elman network's achievements of linguistic process [1][5], we used it to study and solve language issues. Instead of using simple generated sentences as corpora, the work [5] analyzed the real corpus which contains complex sentences and large amount of vocabularies. Elman's method has been redesigned in [5][6] to accomplish such complex task. In the redesigned method, consider a corpus D with N different words $\{c_n;\ n = 1, 2..., N\}$. Each word c_n initially has a random lexical code vector with R dimensions, $c_n^{t=0} = [c_{n1}^{t=0}, c_{n2}^{t=0}, ..., c_{nR}^{t=0}]^T$. The input word stream is encoded accordingly, $p(t) = c_i^t;\ c_i^t \in \{c_n^t; n = 1, 2..., N\}$. This stream $\{p(1), p(2), p(3), ...\}$ is formed by concatenating all sentences in the corpus. $p(1)$ is the first word of the first sentence. Minimize the error between the network's output $o(p(t))$ and its desired output $p(t + 1)$ to satisfy the prediction $o(p(t)) \approx p(t + 1)$ by using the back-propagation algorithm.

The prediction error is drastically reduced by introducing a renew process for the code of each word [5]. The training curve is similar to that in [6]. A renewed code is obtained by normalizing c_n^{raw} every T training steps ,

$$c_n^{raw} = \frac{1}{freq_n} \sum_{\{t|p(t)=c_n\}} o(p(t-1)), n = 1 \sim N, \tag{1}$$

where $freq_n$ is the number of times that the word c_n appears in the corpus. Note that Elman averaged all the hidden output vectors for each word c_n, but we averaged all the prediction vectors for it instead.

All raw codes must be normalized [5] before using them in the next iteration. This is because the network may reduce the prediction error simply by decreasing the distances among all codes. The worst case is that every words converge to a same code vector. In this case, the network achieves zero prediction error. Setting the norm of each code vector as 1, it is able to prevent a diminished solution, $\{\|c_n\| \sim 0, n = 1 \sim N\}$, derived by the back-propagation algorithm.

1.3. Experiments setting

The architecture of the network is $L_i = 64$ input neurons, $L_o = 64$ output neurons, $L_h = 200$ hidden layer neurons, and $L_c = 200$ context layer neurons. The number of neurons are determined empirically. The initial values of synapse

weights W_{hic} and W_{oh} are randomly assigned in the range $[-1, 1]$, and the initial values of the neurons in the context layer are set to zero. The initial code $c_n^{t=0}$ is randomly assigned under the restriction that different words have different codes and then is normalized. Then use the back-propagation algorithm to reduce the prediction error which is $\|o(p(t)) - p(t+1)\|^2$. The learning rate is fixed to 0.01. W_{hic} and W_{oh} are updated after each word presented. Set one epoch, $g = 1$, as when all words in the corpus were presented, and renew the codes every k epochs. {{The corpus contains all Shakespeare 36 plays. All functional words are removed from the corpus. Each word is stemmed as deep as possible. The training is stopped when the error reaches a minimum value. The reduced errors during the training epoches are similar to those in [6]. The trained weights, W_{hic} and W_{oh}, are used to predict the next word of $p(t)$, $o(p(t))$. We use the 10000 words in the corpus with the highest frquencies, $freq_n$.}}

1.4. Multi-meaning re-encoding

The output vectors, $o(p(t))$, are collected for each word when the minimum error is reached [6]. These vectors possess very rich information on the meanings of each word. To save these rich meanings of a word that has a continuous spectrum, the work [5] developed a new code structure to load such richness of semantic meaning. This new structure has the following constraints:

$$\Pi_{r=1}^{R} c_{nr} = 1 \text{ and } c_{nr} \geq 0;\ n = 1 \sim N \text{ and } r = 1 \sim R. \tag{2}$$

It modifies every saved output, $o\left(p(t-1)\right)$, to satisfy the constraints in (2). Instead of a vector position, this new code has a form of a unit volume and all positive features.

Note that a new code, c_n^{new}, obtained by combining two other codes, c_n^A and c_n^B, will keep these constraints. The combining operation is

$$c_{nr}^{new} = c_{nr}^{A} \times c_{nr}^{B},\ r = 1 \sim R. \tag{3}$$

This operation (3) strengthens the consistent features of two codes and neutralizes the discordant features. The constraint for the new code, $\Pi_{r=1}^{R} c_{nr}^{new} = 1$, is naturally satisfied. For functional words, because discordance neutralization happens in every feature, their codes will have a form similar to that of a unit hypercube with each feature equal to 1. This code structure is also a new kind of the semantic space model of word. In this structure, cosine and Euclidean distance are not suitable for measuring the semantic similarity between any two words. A new measure "RSM" (richness in semantic meaning) is developed as the product of all features larger than 1:

$$RSM \equiv ln(\Pi_{r=1}^{R} \max(c_{nr}, 1)). \tag{4}$$

2. Discussion

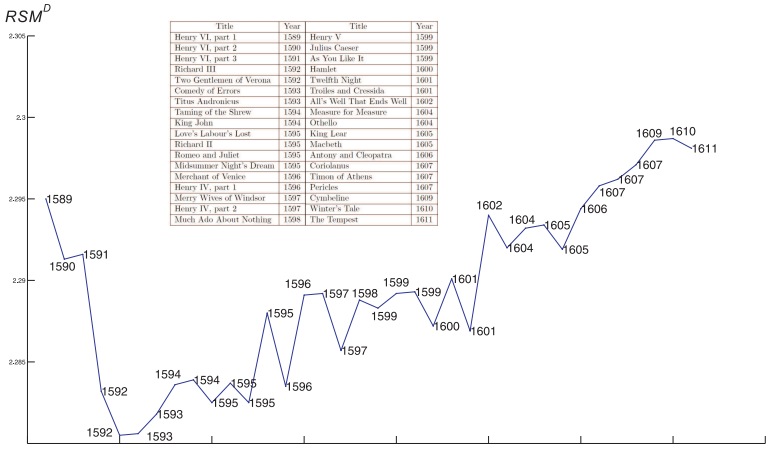
A single RSM value could reveal the richness of a whole play. This single vector is obtained by combing all of the word vectors (all $|D|$ words in a play D) using (3), that is, $c_r^D = \Pi_{c_i \in D}^{|D|} c_{ir}$, $r = 1 \sim R$. Divid the RSM value by the total number of words in each play and apply a natural logarithm function to it: $RSM^D \equiv \ln((1/|D|)\Pi_{r=1}^{R} \max(c_r^D, 1))$. This value, RSM^D, can reveal the average meaning of a word in a whole play. {{Note the 10000 highest frquency words in the corpus are used in the formula RSM^D. All functional words and low frquency words are not included in $|D|$.}} Plot the RSM^D values of Shakespeare's works recorded in 'Table 3 in [5]' and of Mark Twin's works according to the writing time in Fig. 2 and Fig. 3. The high and increasing values after 1604 in Fig. 2 show why people name the years around 1604 "the peak" of Shakespeare [7][9]. To our knowledge, RSM^D is the only one that can reveal the peak. For Mark Twin's works in Fig. 3, RSM^D increases over time also. It's reasonable that an author has better writing skill to handle words and apply rich semantic meanings to words in his/her works when he/she has more living or writing experiences.

There is no clear record to dating Shakespeare's plays. The commonly used chronology of Shakespeare's plays is shown in Fig. 3. We see RSM^D increases over time after the play "Two Gentlemen of Verona" (1952) in Fig. 2. The inconsistent high RSM^D of "Henry VI part 1"(1589), "Henry VI part 2"(1590), and "Henry VI part 3"(1591) may be caused by wrong dating or even different authorship. In [8], E.A.J. Honigmann believed the chronology begins with "Titus Andronicus", which Shakespeare estimated was written in 1586, and continued with "The Two Gentlemen of Verona", which Shakespeare placed in 1587. This implies that "Henry VI" may not be the first work of Shakespeare. Moreover, a number of Shakespeare's early plays have been examined for signs of co-authorship, and especially "Henry VI part 1", which many notable scholars argue, is definitely a collaboration between Shakespeare and at least one, but probably more, other dramatists whose identities remain unknown, although Thomas Nashe, Robert Greene, George Peele and Christopher Marlowe are common proposals [10]. That may cause the RSM^D of "Henry VI part 1" inconsistent, compared with other works in the same period.

2.0.1. Acknowledgement

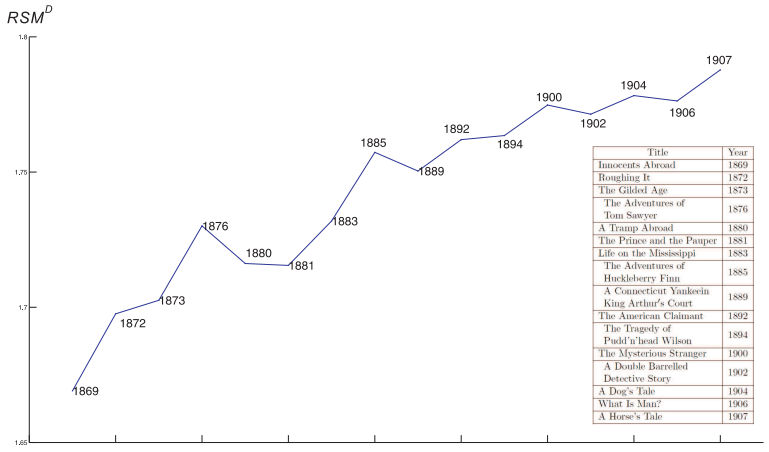
This work was supported by National Science Council under project NSC 100-2221-E-002-234-MY3.

[1] Elman, J.L.: Finding structure in time. Cognitive Science 14, 179–211 (1990)



Title	Year	Title	Year
Henry VI, part 1	1589	Henry V	1599
Henry VI, part 2	1590	Julius Caeser	1599
Henry VI, part 3	1591	As You Like It	1599
Richard III	1592	Hamlet	1600
Two Gentlemen of Verona	1592	Twelfth Night	1601
Comedy of Errors	1593	Troiles and Cressida	1601
Titus Andronicus	1593	All's Well That Ends Well	1602
Taming of the Shrew	1594	Measure for Measure	1604
King John	1594	Othello	1604
Love's Labour's Lost	1595	King Lear	1605
Richard II	1595	Macbeth	1605
Romeo and Juliet	1595	Antony and Cleopatra	1606
Midsummer Night's Dream	1595	Coriolanus	1607
Merchant of Venice	1596	Timon of Athens	1607
Henry IV, part 1	1596	Pericles	1607
Merry Wives of Windsor	1597	Cymbeline	1609
Henry IV, part 2	1597	Winter's Tale	1610
Much Ado About Nothing	1598	The Tempest	1611

Figure 2: The RSM^D of each Shakespeare's work over time



Title	Year
Innocents Abroad	1869
Roughing It	1872
The Gilded Age	1873
The Adventures of Tom Sawyer	1876
A Tramp Abroad	1880
The Prince and the Pauper	1881
Life on the Mississippi	1883
The Adventures of Huckleberry Finn	1885
A Connecticut Yankeein King Arthur's Court	1889
The American Claimant	1892
The Tragedy of Pudd'n'head Wilson	1894
The Mysterious Stranger	1900
A Double Barrelled Detective Story	1902
A Dog's Tale	1904
What Is Man?	1906
A Horse's Tale	1907

Figure 3: The RSM^D of each Mark Twin's work over time

[2] Rumelhart, D.E.: Parallel distributed processing: Explorations. In McClelland, J.L. (eds.), The microstructure of cognition: Foundations. MIT Press, Cambridge, MA (1986)
[3] Elman, J.L.: Distributed representations, simple recurrent networks, and grammatical structure. Machine Learning 7, 195–225 (1991)
[4] Elman, J.L.: Generalization, simple recurrent networks, and the emergence of structure. Proceedings of the 20th Annual Conf. of the Cognitive Science Society, Mahway, NJ (1998)
[5] Liou, C.-Y., Huang, J.-C., Yang, W.-C.: Modeling word perception using the Elman network. Neurocomputing 71, 3150–3157 (2008)
[6] Huang, J.-C., Cheng, W.-C., Liou, C.-Y.: Distributed representation of word. In: ACIIDS, April. 20-22, Daegu city, Korea. LNAI 6591, pp. 169–176. (2011)
[7] Bloom, H.: Shakespeare: The invention of human. Riverhead Books, New York, (1998)
[8] Honigmann, E.A.J.: Shakespeare's Impact on his Contemporaries (1982)
[9] McEnery, T., Oakes, M.: Authorship identification and computational stylometry. Handbook of Natural Language Processing, Marcel Dekker, Inc. 545–562 (2000)
[10] Burns, Edward (ed.) King Henry VI, Part 1. The Arden Shakespeare, 3rd Series; London: Arden (2000)