PREDICTIVE ANALYSIS OF COHESIVENESS IN MULTIVARIATE SEQUENCES USING RECURRENT NEURAL NETWORKS

[1]ROBERT NSINGA, [2]STEPHEN KARUNGARU

Intelligent Systems, Tokushima University, 2-1 Minamijosanjima, 770-8506 Tokushima, Japan
E-mail: [1]robert.nsinga@ceable.com, [2]karungaru@tokushima-u.ac.jp

Abstract - This study is a revised analysis of team effectiveness prediction based on a probability of social sensitivity described as avoluntary inclination of the members to adhere to group general etiquette, practices and norms. Rather than focusing on individual performance metrics like skills and job experience, this study formulates the arbitrary organizational composition of members in a team as a sequence problem. The proposed measure of collective intelligence is modeled through input-to-state and state-to-state transitions in neural networks based on the concept of memory cells to learn the vocabulary of social cohesiveness in variable length sequences. By extending the type of dataset to unconventional time-series, like members in a team engaged in problem-solving activities, this study hopes to achieve results at par with derived survey findings in applied disciplines (socio-psychology).

Keywords - Sequence modeling, prediction, memory-cell

I. INTRODUCTION

Traditional employee analysis surveys require the subject to fill information on a sheet of paper and answer questions objectively to the best of their recollection. Performance surveys, for example, require direct input from the subject. The counter-intuitive practice has long been a breeding ground for subjectivity. When evaluating performance for a given subject (or set of subjects) the activity ought to be the responsibility of a third-party, neutral, observing agent. Performance is also better analyzed in a shared environment where each individual contributes towards a common goal, making the activity a routine metered review of milestones towards that goal.

There has been mounting interest in the idea of aggregating individual IQs of members in a team into a unifying, collective IQ[9], with reproducible associations to performance outcomes, and whether the existence of such formula makes sense. The debate is still ongoing on the cognitive and affective/emotional fronts, though this paper chooses the latter to investigate a socio-psychological approach to performance analysis. (Edmondson, 1999) defines psychological safety as the magnitude to which an individual feels the need to participate in a team. Along the same approach, Woolley et al., 2010 concludes the existence of a factoring element that is in direct correlation with a general propensity to social sensitivity, or how members get along out of their own will. It is imperative to assume the members are free to engage in social sensitivity without coercion, whether directly or indirectly.

II. LITERATURE REVIEW

Since the mid-90's the functional structures of medium to large organizationsembraced a progressive and adaptive change that grew as the Internet grew. Increasing competition, accelerated innovation, and the sharing economy pushed the trends in organizational charting towards functional teams rather than individual jobs. Before the end of the century, multiple reviews on work-teams literaturehad emerged, one of which reflected on the dynamic nature of teams. It argued that teams develop through members' interactions, evolve and adapt to situations as they arise, and exist in a context, (Kozlowski et al., 1999). Though team performance has eluded the majority, most literature assume the following heuristics:

- there is uncertainty about the tackled problem(s),
- there is concerted need for inter-dependency in finding solution(s),
- each member frames the problem(s) as an opportunity to learn

(Nsinga et al., 2018; Edmondson, 1999; Kozlowski & Bell, 2003).

Psychological safety begets team learning, which in turn begets team performance as illustrated in Fig. 1.

Cognitively, teams have long been regarded as processors of tasks, lending perspective to team cohesiveness as an indirectly proportional factor of its members' deference to task-relevant interactions (Edmondson, 1999).



Fig. 1.Cascade of organizational performance. (Edmondson, 1999)

Deep learning algorithms are heavily influenced by the design and function of the human brain. The computer science school of thought assumes the brain

to have a logic unit (processor) and a memory unit, but largely remains vague to detail its implementation at granular (neuron) or higher-level abstraction, and whether the former implies merging units together or not. The (largely unknown) complexity of the cerebral neural network make the brain a giant function that processes input from our senses (sight, hearing, etc.) and provide the best course of action, output, based on instinct, memory, or at times, the conscience. The latter being of particular interest to my research, though I choose to frame it as "gut feeling" for simplistic reasons and limited knowledge ability on the matter.

Deep learning is hierarchical in the sense that the algorithm extracts features into factors at each layer. A deeper layer's factors become the next layer's features [3, 10]. Approximating neuronal complexity has created, among others, mathematical theories and implementations like recurrent feed-forward networks that are not limited to a singular output per input, but rather can induce a chain of cohesively related outputs. This is also a particular characteristic of our brain which actively generates output over extended periods of time. A simplified functional implementation of deep learning goes like this: First, a collection of functions "neurons" are created and connected together, allowing them communicate with each other. Next, the network is asked to solve a problem by repetitively applying the same solution over and over, each time strengthening the connections that lead to success and weakening those that lead to failure; a process called supervised learning.

Mathematically, we model a high dimensional mapping "F" by passing it through a superposition of semi-affine functions such that given an input "x" the mapping generates a predictor "Y". The mapping process $Y = F(x)$ is the core function of a deep learning algorithm. The brain apparatus looks nothing like the concepts I have highlighted above. For example, feed-forward recurrent networks implementations opted for naive approaches to learning time series data by configuring as many neurons as there are time steps and sequentially feeding the model with time-step data as a block, usually referred to as context, on the premise that the data shows a certain periodic property. Time should be understood as a measure of sequential ordering, like text or music. The order matters because it cements the context, although research has shown that the human brain can perfectly ignore an out-of-sequence ordering and rectify it [12].

By revisiting the sequence-based type abstraction of context, this research proposes to regard a team as a multivariate sequence of its members, or context block, and treating its features as a chain of its members' most salient socio-psychological properties engaged in achieving cohesiveness (Festinger, 1950). The team can be of any size.

A. The Concept of Memory in Recurrent Networks

LSTM is based off RNN and was proposed by Hochreiter and Schmidhuber (1997) to address shortcomings in its predecessor, specifically the vanishing and exploding gradient problems. This led the community to refer to the original RNN as Vanilla-RNN and LSTM as RNN-LSTM, or simply LSTM. Similar to the butterfly effect studied in meteorology, LSTM models the impact of input sensitivity on a short-, as well as long-term recurrence space. Recurrence, or stepping, is a flow of state (memory) from the past "$t-1$" to the present "t", or a function that takes old information and presents new information. The recurrence function ensures the impact of information vanishes, explodes, or fluctuates in time (sequential ordering).

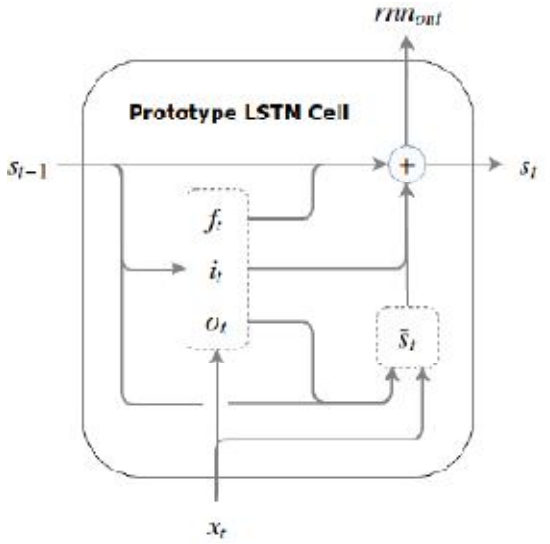


Fig. 2 Prototype LSTM cell. Source: r2rt.com

It should be noted here that sensitivity is also known as gradient. Here is the intuition:

1. If the gradient explodes, the learning becomes clogged. The process stops.
2. If the gradient vanishes, fresh information "distracts" from longer-term dependencies. Learning is difficult.

The recurrence function is gated, or selective, in its arbitrary state-flow control. The gates are input, output and forget/remember denoted i_t, o_t, f_t respectively.

$$\begin{aligned} i_t &= \sigma(W_i S_{t-1} + U_i x_t + b_i) \\ o_t &= \sigma(W_o S_{t-1} + U_o x_t + b_o) \\ f_t &= \sigma(W_f S_{t-1} + U_f x_t + b_f) \end{aligned} \quad (1)$$

Where:

- σ is the activation function,
- $S_{t-1} \in R^n$ is the prior state,
- x_t is the current state,
- W, U, and b are the weights and biases.

Let's consider $\tilde{S}_t$ the next change of state expected. Gated selectivity requires prior state to be multiplied element-wise to the output gate as follows:

$$\tilde{S}_t = \phi(W(o_t \odot S_{t-1}) + Ux_t + b) \qquad (2)$$

Gated selectivity gives an overview of the memory intuition behind LSTM, The next state is then the addition of two element-wise multiplications, one on prior state by forget gate, and the other with the expected change by input gate:

$$S_t = f_t \odot S_{t-1} + i_t \odot \tilde{S}_t \qquad (3)$$

State-flow control is idealistic. The fundamental issue with its real-life application is in the arbitrary initialization of its weights and biases, no matter how careful they are picked. B
ut the above formulas are key to understanding how context is remembered. The diagram of state-flow control supports the above notations of recurrence control, see Fig. 2.

III. PROPOSED METHODS

By analogy to its use in natural language processing, LSTM has shown successful memory capacity in inferring a cohesive sequence of words from unpredictable recurring patterns. It can guess words with relative accuracy. Consider the following:

TABLE I: TEXT RECOMMENDER USING SEQUENCE MODELING

Context	Recurrence/input				Label
	t-3	t-2	t-1	t	
1	At last,				
2		the ladd			
3			er, whic		
4				h, had be	[en built]
Full-text	At last, the ladder, which had be(en built)				

Each recurrence is a context block of equal size (8 characters each).
The slicing is arbitrary but commensurate with the reading speed of the LSTM, and the last column represents the label (or real output). Let's theorize about replacing the LSTM with a human (or the brain of that human) and introduce a second human, a speaker. Assume that the speaker has uttered the words in sequence, from t-3 to t, and the human model is evaluated at t+1 (future). Here are the possible scenarios:

a) The speaker reads out loud the line, and the listener concludes with "en built"as their best guess given the grammatical structure so far, or
b) The listener has knowledge of the source, and joins the speaker and says "en built, slowly by slowly, one hope at a time" (extracted from "The Ladder", a poem by Katherine G. Berry).

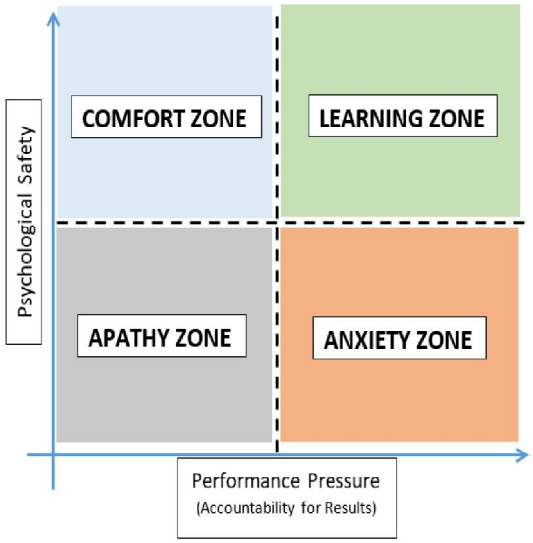


Fig. 3 The 4 quadrants of team performance benchmarking.
Source: p2theblog.com

Maintaining a similar thought pattern, let us consider a team composed of the following individual names (input) ordered arbitrarily, and let's consider the LSTM as the same human brain, whose task is to guess the next name:

- Maj. John Reisman
- Maj. Gen. Sam Worden
- Joseph Wladislaw
- Robert T. Jefferson
- Victor R. Franko
- Sgt. Clyde Bowren
- [...]

Each name representsa character; if thelearning model has extensively been exposed to the most famous teams in cinema history, then it should output the last name as either of the remaining 6 members in the 1967 war film The Dirty Dozen.

What if, instead of names the list presents tables of properties, character traits, or behavior engaged in maintaining cohesiveness? And what if, instead of another name the modelreturns a probability of cohesiveness?

Cohesiveness is a measure of effectiveness, or how well a mix-and-match of members achieves the best outcomes of performance (Edmondson, 1999; Kozlowski et al., 1999).

The probability would then be analyzed on a fitting graph of organizational expectations, see Fig. 3.

IV. EXPERIMENTS

The dataset used in experiments is employee logs in behavioral and social metrics simulating a series of problem-solving activities. Extensively it includes budget-spending information, team/group code, and behavioral data like deference, openness, etc. and a number that determines the ratio of contributions per meeting or group activity.The data is synthetic.

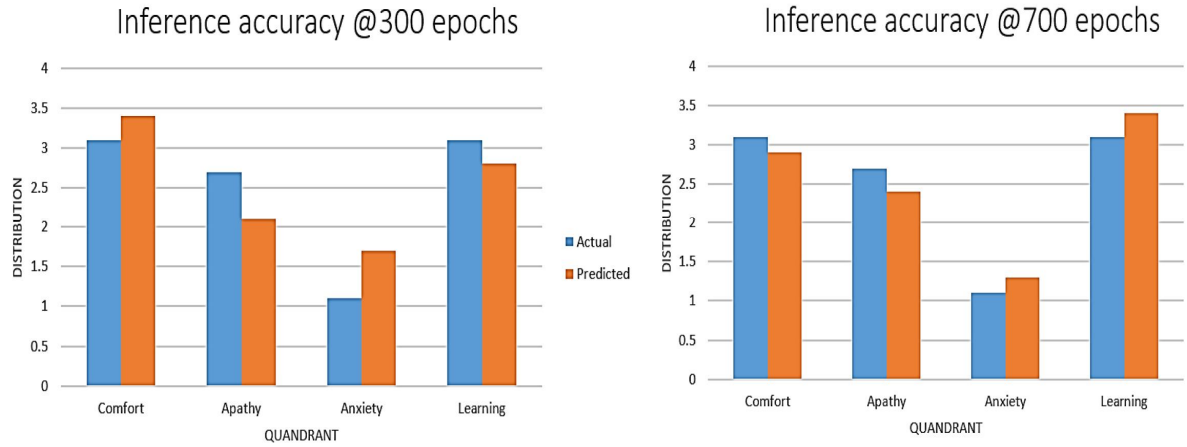


Fig. 4A: batch size is 15, the embedding is 150. No dropout layer; B: Included a dropout and bumped the batch at 30 with an unchanged embedding of 150.

To assist in the modeling and preparation, all this information is pre-processed and clustered. For example a value-pair of normalized contributions per meeting and the normalized percentage budget-spending is considered a psychological-safety versus performance-pressure data point for each input (employee), see Fig. 3.

It is worth noting that the elements fed to our model are of variable length, and go in at once. LSTMs are notoriously tricky to parallelize because of their sequencing nature, but mini-batches help in optimizing only if padding is used [3, 4].

This is achieved by padding all sequences to the size of the biggest element, overall it modifies the nature of the dataset from a time-series to a supervised learning set.

A. Network Definition

Structurally, the network is a slight variant of (Shi Xingjian et al., 2015) used in speed forecasting [13], optimized with Adam (Diederik et al., 2014), see Fig. 5. The sequences, however, are of variable length as noted above.

Practically, the experiments estimate the function that best defines the collective IQ [9] of the team. Arrangement and ordering is assumed to hold contextual structure, see Fig. 5.

By passing through all sequences the network learns a vocabulary that could:

a) Predict a member that fits a pattern ofteam effectiveness.
b) Or output coordinates that map to either quadrant of performance benchmarking, see Fig. 3. This paper is interested about this output more.

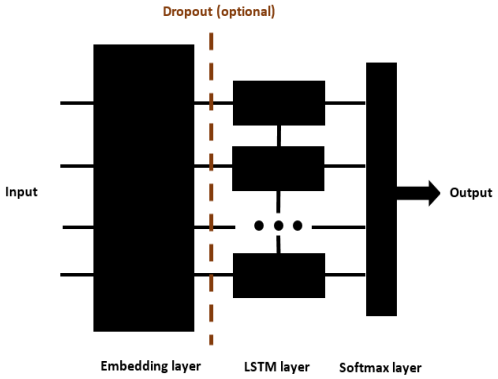


Fig. 5 Overview of the recurrence network. Note the optional dropout [9]

RESULTS AND CONCLUSION

The network is implemented with the TensorFlow library[10]. The dataset is a trimmed version of (Nsinga et al., 2018) updated to reflect the latest iteration in the study of sequence models. Evaluation is performed on-the-fly, but inference is written in C++.

A. Results

See Fig. 4 showing success rate on 2 experiments of differing hyper-parameters.The highest recorded accuracy is 65.7%, for a total of 8 experiments of hyper-parameters tuning.

B. Conclusion

This research uses LSTM to monitor performance by focusing on behavioural metrics in a problem-solving space. Though the semantics for collective intelligence are not defined, the estimator expertly clusters work-teams, whose members feel relatively safe to engage in interpersonal risk taking, by memorizing the features logged during problem-solving activities.

REFERENCES

[1] Kozlowski, S.W.J. & Bell, B.S., Work Groups and Teams in Organizations, Industrial and Organizational Psychology (vol. 12), pp. 333-375, 2003.

[2] Hochreiter et al., Long short-term memory, Neural Computation 9.8, pp. 1735-1780, 1997

[3] S. Pitis, Written Memories: Understanding, Deriving and Extending the LSTM, Available: https://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html

[4] F. Chollet, Keras: Deep Learning for Python, http://github.com/fchollet/keras, 1f0bd8c, 2015.

[5] Edmondson, Amy, Psychological safety and learning behavior in work teams, Administrative science quarterly vol. 44/2, pp. 350-383, 1999.

[6] Baek, P. and Reidy, S., Is there a correlation between diversity in the workforce and diversity in the leadership and business results? Available: http://digitalcommons.ilr.cornell.edu/student/119, (2016).

[7] Guide: Understand team effectiveness. Available: https://rework.withgoogle.com/print/guides/5721312655835136/ (Nov 2016)

[8] Gal, Yarin, and Zoubin Ghahramani, Dropout as a Bayesian approximation: Insights and applications, Deep Learning Workshop, ICML, 2015

[9] A. W. Wooley et al, Evidence for a Collective Intelligence Factor in the Performance of Human Groups, Science 2010 (pp. 686-688).

[10] Abadi et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, tensorflow.org, 2015.

[11] R. Nsinga et al, Benchmarking Organizational Performance Dynamics Using Memory Cells Recurrent Neural Networks, ICIAE2018, 2018.

[12] Economic and Social Research Council (ESC). How can we still raed wrods wehn teh lettres are jmbuled up? ScienceDaily, 2013

[13] Shi Xingjian, Chen Zhourong, Wang Hao, Yeung Dit-Yan, Wong Wai-kin, Woo Wang-chun, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting" arXiv preprint arXiv:1506.04214 (2015).

★★★