SELF-ORGANIZING WORD REPRESENTATIONS FOR FAST SENTENCE PROCESSING

STEFAN L. FRANK

Nijmegen Institute for Cognition and Information, Radboud University Nijmegen; and
Institute for Logic, Language and Computation, University of Amsterdam
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
E-mail: sfrank@science.uva.nl

Several psycholinguistic models represent words as vectors in a high-dimensional state space, such that distances between vectors encode the strengths of paradigmatic relations between the represented words. This chapter argues that such an organization develops because it facilitates fast sentence processing. A model is presented in which sentences, in the form of word-vector sequences, serve as input to a recurrent neural network that provides random dynamics. The word vectors are adjusted by a process of self-organization, aimed at reducing fluctuations in the dynamics. As it turns out, the resulting word vectors are organized paradigmatically.

Keywords: Word representation; Sentence processing; Self-organization; Recurrent neural network; Reservoir computing.

1. Introduction

There exist several psycholinguistic models that represent words as vectors in a high-dimensional state space. Distances between vectors encode strengths of relations between the corresponding words. Invariably, these are *paradigmatic* relations: Two vectors are close together in the state space if the represented words have a strong paradigmatic relation, that is, they belong to the same part-of-speech and/or have similar meaning. The best known models of this sort are Latent Semantic Analysis [1] and Hyperspace Analog to Language [2], but there are many others (for an overview, see [3]).

Such models have accounted for a considerable amount of experimental data regarding, among others, synonym judgement [1], lexical priming [4,5], vocabulary acquisition [1], and semantic effects on parsing [6]. This suggests that the mental lexicon is indeed organized paradigmatically, raising the question *why* this would be so. It seems unlikely that our mental lexicon has developed to make synonym judgement or lexical priming possible. Rather, it makes more sense to organize words in a manner that facilitates fast sentence processing and production. In such an organization, two word vectors would be close together if one words is likely to follow the other in a sentence. However, this constitutes a *syntagmatic* rather than a paradigmatic organization.

This chapter presents a connectionist model demonstrating that the two types of organization are in fact strongly related: A syntagmatic organization of word sequences is facilitated by a paradigmatic organization of individual words. That is, word representations encode paradigmatic relations because this allows for time-efficient sentence processing.

A related explanation for the nature of word representations was provided by the work of Elman [7], who trained a Simple Recurrent Network to predict which word would occur next at each point in a large number of sentences from an artificial language. During training, word representations were adapted to become more useful for this word-prediction task. As it turned out, the resulting organization was clearly paradigmatic. This suggests that a paradigmatic organization of words facilitates word prediction, which is presumably useful for sentence processing.

The model presented here is also a recurrent neural network that processes sentences from an artificial language, but differs from Elman's work in three respects. First, I assume that word representations are explicitly adapted to allow for faster sentence processing, and that word prediction only follows from this implicitly. In Elman's case, this relationship is reversed since his network was explicitly trained to perform word prediction, while it was left implicit how this is beneficial for sentence processing.

Second, word representations are adjusted by an unsupervised process of self-organization rather than by supervised backpropagation. In has been argued that, in the brain, unsupervised learning occurs in the cortex while supervised learning only takes place in the cerebellum [8]. Given ample evidence that word meanings are stored in the cortex, unsupervised learning is preferred for the current simulations.

Third, the weights of recurrent connections in the network are not adapted, making neural network training much more efficient. Current developments in recurrent network research have focused on so-called 'reservoir computing' [9–11], in which the recurrent part of the network is not trained but serves as a reservoir of complex dynamics that forms a task-independent memory trace of the input sequence. A non-recurrent network is then trained to transform the reservoir's activation states into target outputs. Only recently have such systems been applied to language processing [12–15]. The simulations presented here differ from other applications of reservoir computing in that learning is unsupervised, meaning that there are no target outputs and, therefore, no output connections to train. Instead, it is the input representations that are adapted.

The rest of this chapter is organized as follows: Section 2 describes the semi-natural language that was used in the simulations. Following this, Sec. 3 gives the details the model and the rationale behind the algorithm for adaptation of word representations. Simulation results are presented in Sec. 4 and discussed in Sec. 5.

2. The language

The artificial language used for the simulations was originally designed by Farkaš and Crocker [16] for training a network on the word-prediction task. All sentences of this language are also sentences of English but, of course, the language has much smaller vocabulary and simpler grammar.

2.1. *Lexicon*

There are 71 words in the language, as listed in Table 1. Note that the word 'who' serves both as a relative and as an interrogative pronoun. Additionally, there is a period symbol to mark the end of a sentence, making a total of 72 symbols.

2.2. *Sentences*

Words are combined into sentences according to a probabilistic context-free grammar (PCFG) that is too complex to be printed here in full. A simplification of the grammar is presented in Table 2. Note that noun-verb number agreement and some semantic constraints are not shown, but do apply nevertheless.

Sentences come in three types: declaratives, interrogatives, and imperatives. Declaratives can contain subject-relative clauses and object-relative clauses, which can also be nested. The average sentence length is 5.5 words, plus the period that ends each sentence.

Table 1. Words and word classes in the language.

Class	Subclass	Words
Noun	Proper	*John, Kate, Mary, Steve*
	Mass	*bread, meat, fish*
	Singular	*boy, cat, dog, girl, man, woman*
	Plural	*boys, cats, dogs, girls, men, women*
Verb	Singular	*barks, bites, chases, eats, feeds, hates, hears, likes, runs, sees, sings, swims, talks, walks*
	Plural	*bark, bite, chase, eat, feed, hate, hear, like, run, see, sing, swim, talk, walk*
	Auxiliary singular	*does, is ,was*
	Auxiliary plural	*do, are, were*
	Other	*wanna*
Adjective		*crazy, ferocious, good, happy, hungry, mangy, nice, pretty, sleazy, smart*
Article		*a, the*
Pronoun	Demonstrative	*that, those*
	Interrogative	*what, where, who*
	Relative	*who*

Table 2. Simplification of the PCFG for producing sentences. Items in square brackets are optional. N = singular or plural noun; N_{pr} = proper noun; N_{mass} = mass noun; V_{tr} = transitive verb; V_{in} = intransitive verb; Adj = adjective; Dem = demonstrative pronoun; Art = article.

Head		Production
S	→	Declarative . $\mid$ Interrogative . $\mid$ Imperative .
Declarative	→	NP [*who* RC] VP $\mid$ NP V_{be} Adj $\mid$ Dem V_{be} NP
Interrogative	→	Q_{wh} $\mid$ Q_{aux}
Imperative	→	VP
NP	→	Art [Adj] N $\mid$ [Adj] N_{pr} $\mid$ N_{mass}
VP	→	V_{in} $\mid$ V_{tr} NP
RC	→	V_{in} $\mid$ V_{tr} NP [*who* RC] $\mid$ NP [*who* RC] V_{tr}
Q_{wh}	→	*where/who* V_{be} NP $\mid$ V_{do} NP V_{in} $\mid$ *what* V_{do} NP *do*
Q_{aux}	→	V_{do} NP [*wanna*] VP $\mid$ V_{be} NP Adj
V_{be}	→	*is* $\mid$ *are* $\mid$ *was* $\mid$ *were*
V_{do}	→	*do* $\mid$ *does*

3. The model

3.1. *The dynamical system*

We begin by taking the simplest possible discrete-time linear dynamical system:

$$\mathbf{x}_t = \mathbf{W}\mathbf{x}_{t-1} + \mathbf{y}_t,$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the system's n-dimensional state vector (with $\mathbf{x}_0 = \mathbf{1}$) and $\mathbf{y}_t \in \mathbb{R}^n$ the input at time step t. Matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ has values randomly chosen from a uniform distribution centered at 0, and is rescaled to have a spectral radius of 1. In neural networks terms, $\mathbf{x}_t$ would be the activation pattern over n units, $\mathbf{y}_t$ the input activation, and $\mathbf{W}$ the matrix of the recurrent network's connection weights.

A sequence of t input words (e.g., a sentence) corresponds to a sequence of t input vectors $\mathbf{y}_1, \ldots, \mathbf{y}_t$. The sequence $\mathbf{x}_0, \ldots, \mathbf{x}_t$ is the state-space trajectory resulting from that input. More specifically, each word w in the language's 72-symbol vocabulary is represented by a vector $\mathbf{v}_w \in \mathbb{R}^k$, with $k < n$. If word w is the input word at time step t, then the first k elements of input vector $\mathbf{y}_t$ equal $\mathbf{v}_w$, while the other elements are 0. In neural network terms, this means that only the first k of the recurrent network's n units receive input activation.

3.2. *Adjusting word vectors*

Initially, each word vector $\mathbf{v}_w$ has random values, uniformly distributed between ±1. These vectors are adjusted on the basis of 5000 training sentences that were randomly generated from the PCFG of Table 2.

Let $w_1, \ldots, w_{t-1}$ be the training sentence so far (hence, $\mathbf{x}_{t-1}$ is the current state), and w_t the next word to occur. This provides some evidence that w_t is likely to occur after $w_1, \ldots, w_{t-1}$. In a syntagmatic organization of state space, this would mean that the state $\mathbf{x}_t$, resulting from input $\mathbf{y}_t$, is relatively close to $\mathbf{x}_{t-1}$ because, in such an organization, the nearness of two consecutive states mirrors the likelihood of the second state following the first. This consideration leads to the following informal rule for adjusting word vectors: Whenever $\mathbf{x}_{t-1}$ and $\mathbf{y}_t$ occur as a result of training input, input $\mathbf{y}_t$ is changed to $\mathbf{y}'_t$ such that the resulting $\mathbf{x}'_t$ is closer to $\mathbf{x}_{t-1}$ than $\mathbf{x}_t$ would have been. An even less formal way to put this is: Reduce the fluctuations of network activation resulting from the training input.

Formally, the learning rule is expressed by:

$$\Delta\mathbf{v}_w = \eta\left(\mathbf{x}_{t-1}^{(k)} - \mathbf{W}^{(k)}\mathbf{x}_{t-1}^{(k)} - \mathbf{v}_w\right), \tag{1}$$

where w is the word that occurs at time step t in the training sequence, $\mathbf{x}_{t-1}^{(k)}$ denotes the vector consisting of the first k elements of $\mathbf{x}_{t-1}$, $\mathbf{W}^{(k)}$ is the matrix consisting of the first k rows of $\mathbf{W}$, and $\eta = .001$ is a learning rate parameter.[a]

[a] The reason why $k < n$ is that, otherwise, $\mathbf{x}_t = \mathbf{x}_{t-1}$ can easily be obtained for all t,

3.3. *Measuring syntagmaticity*

If the state-space trajectories indeed show a syntagmatic organization, trajectories resulting from grammatical sentence input should be shorter than those resulting from random word sequences. Syntagmaticity is therefore measured by comparing trajectory lengths resulting from grammatical sentences to those resulting from 'pseudo sentences'.

A set of 3352 test (i.e., non-training) sentences was fed through the system and the euclidean distances between all consecutive points in all resulting trajectories were summed to give the total trajectory length l_{test}. Next, pseudo sentences were constructed from the test sentences by randomly reordering the words, while leaving the end-of-sentence markers in place. This guarantees that pseudo sentences have the same length distribution and word frequencies as test sentences. Also, care was taken to make sure that word repetitions occur as often in the pseudo sentences as in test sentences. The extent to which the system is syntagmatic is now defined as

$$\text{syntagmaticity} = \frac{l_{\text{pseudo}}}{l_{\text{test}}},$$

where l_{pseudo} is the total trajectory length resulting from processing the pseudo sentences. Before training, there is no reason to expect any difference between l_{test} and l_{pseudo}, so the syntagmaticity level will be close to 1. If training is successful, syntagmaticity becomes larger than 1.

3.4. *Measuring paradigmaticity*

Word representations are organized paradigmatically if the vectors for words that belong to the same part-of-speech and/or have similar meaning, are closer together than vectors for words that are not paradigmatically related. The paradigmaticity of word vectors is measured by first defining classes of paradigmatically related words. There are 12 such classes, and they are exactly the 12 (sub)classes of Table 1 that contain more than one word.

Next, the average euclidian distances among vectors of words within a class (d_{within}) and between classes (d_{between}) are computed. The extent to which word vectors are organized paradigmatically is the ratio between the two:

by setting all $\mathbf{v}_w$ to $\mathbf{x}_0 - \mathbf{W}\mathbf{x}_0$. In other words, the recurrent units that do not receive input provide some 'noise' which cannot be compensated perfectly by adjusting the word representations.

$$\text{paradigmaticity} = \frac{d_{\text{between}}}{d_{\text{within}}}.$$

Initially, all word vectors are random so paradigmaticity will be close to 1. If word vector adjustment leads to a paradigmatic organization of the words, the measure for paradigmaticity will become larger than 1.

4. Results

4.1. *Parameter setting*

To investigate the effects of the dimensionalities of the state space and of the word vectors, the values of n and k were varied from 25 to 150, and from $.2n$ to $.9n$, respectively. There turned out to be no large qualitative effect of n. Syntagmaticity improved with larger k (albeit at the expense of paradigmaticity), which was to be expected since larger k means that more of the system's dynamics can be controlled by the adaptation algorithm that was designed to increase systematicity.

4.2. *Syntagmaticity and paradigmaticity*

Figure 1 shows how syntagmaticity and paradigmaticity develop during training, with parameters set to $n = 60$ and $k = 50$. As expected, syntagmaticity quickly rises above 1. This shows that the adaptation rule of Eq. (1) had the desired effect on syntagmaticity. After a few training cycles, however, syntagmaticity decreases slightly and levels of at around 1.28 (note the logarithmic scaling of the x-axis). More interestingly, the organization of word representations becomes strongly paradigmatic. Even after syntagmaticity has stabilized, the self-organizing process that was designed to increase syntagmaticity results in an increase in paradigmaticity instead. This is clear evidence for a link between the two types of organization.

4.3. *Word representations*

As Fig. 1 shows, the level of paradigmaticity more than doubles as a result of adapting word representations. It is not obvious, however, what this means in practice. Is the clustering of word vectors into meaningful groups strong enough to be noticeable? In Figs. 2 and 3, the word vectors are plotted according to their first two principal components, which account for as much as 93.8% of variance. Signs of a paradigmatic organization are clearly visible. For instance, the four proper nouns cluster together,

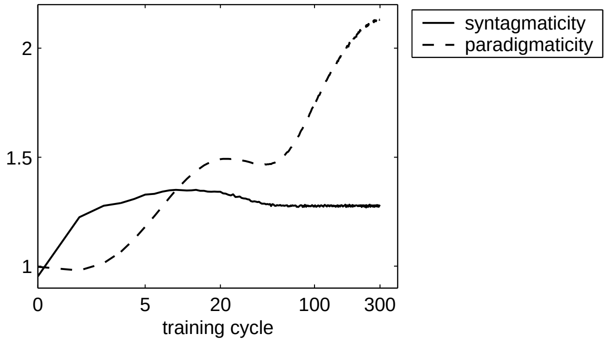

Fig. 1. The effect of training on syntagmaticity of state-space trajectories and paradigmaticity of word representations. In each training cycle, all 5000 training sentences are processed.

as do the singular verbs. However, there is also some evidence that the organization is incomplete, for example, the singular nouns do not seem to be clearly separated from the plurals. In retrospect, this is easy to explain: In Eq. (1), the change in $\mathbf{v}_w$ depends only on the previous inputs and not on what follows. Whether a singular or plural noun can appear, does not depend on the previous context, so the two subclasses will not be separated.

5. Discussion

The model's results clearly support the claim that words are represented according to their paradigmatic relations because this facilitates a syntagmatic organization of word sequences. The latter is useful for fast sentence processing and production, because it means that words that are likely to occur next are near the current position $\mathbf{x}_{t-1}$ in state space, so they can be accessed quickly.

Most likely, the paradigmaticity of word representation can be improved by changing Eq. (1) such that $\Delta\mathbf{v}_w$ comes to depend not only on the current word's previous context, but also on the following input word. Of course, it remains an empirical question whether the word representations constructed by such a model would explain more experimental data than a model like the current one, in which only the previous context is relevant.

If the distance between consecutive states $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ corresponds closely to the probability that the word that gave rise to $\mathbf{x}_t$ occurs in the context of $\mathbf{x}_{t-1}$, the model can be said to perform word prediction implicitly. Unlike Elman's network, the model does not give an explicit probability estimate for each word but such estimates could be derived from the orga-

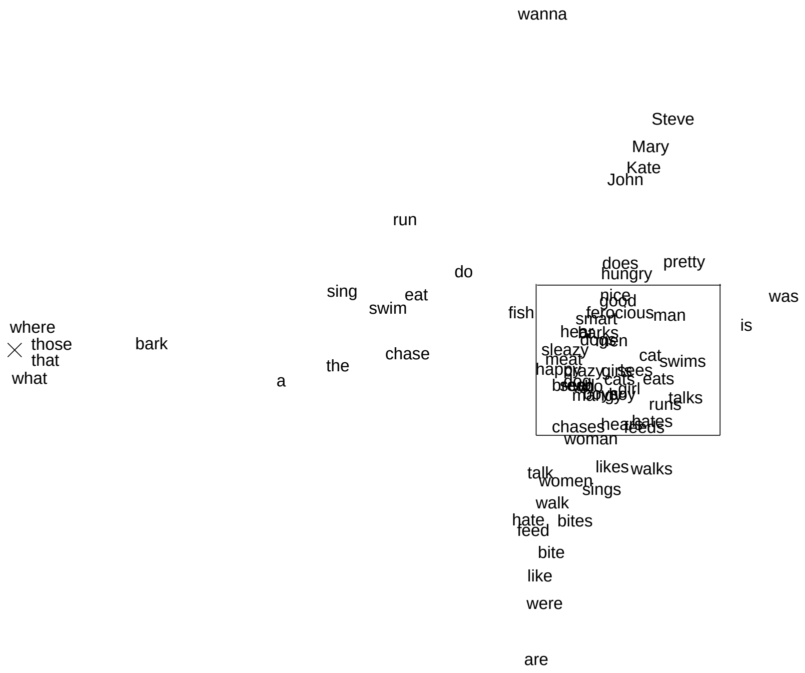

Fig. 2. Representations of the 71 words, projected onto the first two principal components. The '×' on the left-hand side indicates the position shared by the words *where*, *what*, *those*, and *that*. The area in the rectangle is shown enlarged in Fig. 3.

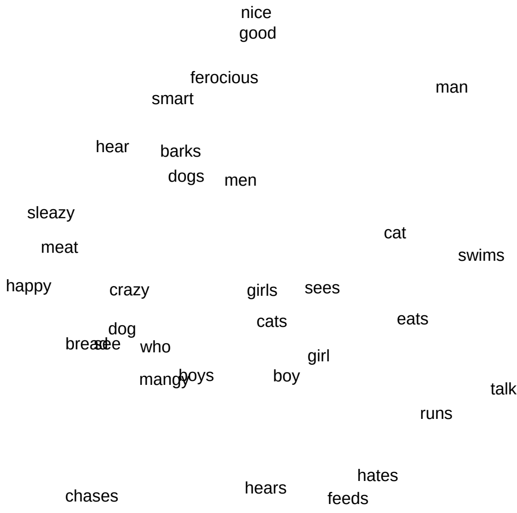

Fig. 3. Close-up of the rectangular area in Fig. 2.

nization of the state space.

If these word-probability estimates are accurate, it might be possible to use the model for predicting word-reading times. It has been argued by Hale [17] and Levy [18] that the time needed to read a word is proportional to its ‘surprisal’, which is simply the negative logarithm of its probability. If Hale and Levy are correct, and the model’s state-space distances correlate positively with word surprisal, the model would predict word-reading times.

The sentences were generated by a known PCFG, so the probability of each word in each test sentence is available. However, the correlation coefficient between the negative logarithms of these probabilities and the model’s state-space distances is only .27 (compared to .23 in advance of training), so the model cannot be said to be accurate enough to account for reading-time data. Considering that matrix **W** had fixed random values, such accurate predictions could hardly be expected. It is not unlikely that predictions could be improved by also adjusting **W** to the training inputs.

References

1. T. K. Landauer and S. T. Dumais, *Psychological Review* **104**, 211 (1997).
2. C. Burgess, K. Livesay and K. Lund, *Discourse Processes* **25**, 211 (1998).
3. J. A. Bullinaria and J. P. Levy, *Behavior Research Methods* **39**, 510 (2007).
4. M. N. Jones and D. J. K. Mewhort, *Psychological Review* **114** (2007).
5. W. Lowe and S. McDonald, The direct route: mediated priming in semantic space, in *Proceedings of the 22nd annual conference of the Cognitive Science Society*, eds. L. R. Gleitman and A. K. Joshi (Mahwah, NJ: Erlbaum, 2000) pp. 806–811.
6. C. Burgess and K. Lund, *Language and Cognitive Processes* **12**, 177 (1997).
7. J. L. Elman, *Cognitive Science* **14**, 179 (1990).
8. K. Doya, *Neural Networks* **12**, 961 (1999).
9. H. Jaeger, Adaptive nonlinear system identification with echo state networks, in *Advances in neural information processing systems*, eds. S. Becker, S. Thrun and K. Obermayer (Cambridge, MA: MIT Press, 2003) pp. 593–600.
10. H. Jaeger and H. Haas, *Science* **304**, 78 (2004).
11. W. Maass, T. Natschläger and H. Markram, *Neural Computation* **14**, 2531 (2002).
12. S. L. Frank, *Connection Science* **18**, 287 (2006).
13. S. L. Frank, Strong systematicity in sentence processing by an Echo State Network, in *Proceedings of ICANN 2006*, eds. S. Kollias, A. Stafylopatis, W. Duch and E. Oja, LNCS, Vol. 4131 (Berlin: Springer, 2006) pp. 505–514.
14. S. L. Frank and W. F. G. Haselager, Robust semantic systematicity and distributed representations in a connectionist model of sentence comprehension, in *Proceedings of the 28th annual conference of the Cognitive Science Society*, eds. R. Sun and N. Miyake (Mahwah, NJ: Erlbaum, 2006) pp. 226–231.

15. M. H. Tong, A. D. Bickett, E. M. Christiansen and G. W. Cottrell, *Neural Networks* **20**, 424 (2007).
16. I. Farkaš and M. W. Crocker, Recurrent networks and natural language: exploiting self-organization, in *Proceedings of the 28th annual conference of the Cognitive Science Society*, eds. R. Sun and N. Miyake (Mahwah, NJ: Erlbaum, 2006) pp. 1275–1280.
17. J. Hale, A probabilistic Early parser as a psycholinguistic model, in *Proceedings of NAACL*, 2001 pp. 159–166.
18. R. Levy, *Cognition* (in press).

Acknowledgements

I would like to thank Igor Farkaš for kindly providing the sentences used in the simulations presented here, and Michal Čerňanský for useful discussions. This research was supported by grant 451-04-043 of the Netherlands Organization for Scientific Research (NWO).