Comenius University in Bratislava
Faculty of mathematics, physics and informatics

Connectionist model of sentence comprehension

Master thesis

2018

Bc. Anton Kováč

Comenius University in Bratislava
Faculty of mathematics, physics and informatics

Connectionist model of sentence comprehension

Master thesis

Study programme:	Cognitive Science
Field of Study:	2503 Cognitive Science
Training work place:	Department of Applied Informatics
Supervisor:	doc. RNDr. Martin Takáč, PhD.

Bratislava, 2018
Bc. Anton Kováč

33900609

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Anton Kováč
Study programme:	Cognitive Science (Single degree study, master II. deg., full time form)
Field of Study:	Cognitive Science
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Connectionist model of sentence comprehension

Annotation: Sentence comprehension models (e.g. Frank, Rohde) usually create a compressed representation of a sentence using a recurrent network and train to map it unto a situation/meaning representation. Meaning in this project will be a SOM-like representation of working memory episode (Takac & Knott). The goal of the thesis will be to compare a baseline SRN model of sentence representation to a MSOM based sentence processing wherein the representation of meaning is obtained by top-down backward propagation in a model trained for sentence production task.

Aim: Design, implement and test a connectionist model of sentence comprehension based on WM architecture of Takac & Knott using MSOM vs SRN as the sentence processing component.

Literature: Frank, S (2005): Sentence comprehension as the construction of a situational representation: a connectionist model. Proceedings of AMKLC'05, Espoo, Finland.
Plaut, D., Rohde, D. (2003): Connectionist Models of Language Processing. In Tsuzuki T. (Ed.), Special issue of Cognitive Studies, Japan, 10(1), 10-28.
Takac, M. and Knott, A.: Working memory encoding of events and their participants: a neural network model with applications in sensorimotor processing and sentence generation. In Proceedings of the 38th Annual Conference of the Cognitive Science Society, Austin, TX. 2345-2350. 2016.

Supervisor:	RNDr. Martin Takáč, PhD.	
Department:	FMFI.KAI - Department of Applied Informatics	
Head of department:	prof. Ing. Igor Farkaš, Dr.	
Assigned:	22.12.2015	
Approved:	22.12.2015	prof. Ing. Igor Farkaš, Dr. Guarantor of Study Programme

....

Student Supervisor

Acknowledgment: I would like to thank my supervisor Martin Takáč for his insight, constructive feedback, and invaluable help throughout the creation of the thesis. I would also like to thank my sister for big support and help with grammatical issues.

Abstract

Language and ability to communicate are thanks to their importance to humanity one of the most discussed phenomena in scientific field. The aim of detailed analysis of their nature lead to the creation of separate scientific disciplines focused on their specific attributes. Behavioural methods were the key explanations in certain particular signs although because of the difficulties of their design and often ethical restrictions they cannot be applicable in some cases of research. Computational models bring into this manner new direction of their investigation.

In this work we aimed to create biologically plausible model of language comprehension based on a principle of self-organization and its comparison with model using supervised learning in sentence processing component. We used Self-Organizing Map for the representation of the meaning of the sentences. As a sentence processing component was chosen Merge Self-Organizing Map. On the other hand the reference model used Simple Recurrent Network. These components were trained to map the meaning of the sentences to the their text representation even after processed part of these sentences. We tested these models to predict the meaning of the sentences after the particular words were presented to them. We also scrutinized how well they can reconstruct the specific elements representing meaning of particular sentence.

The results suggest, the model based on self-organization principle can sufficiently predict the meaning of the sentence after presenting each word in comparison with the reference model. In the prediction task the self-organization based model performed even better then the other model. Thus, we came to the conclusion that biologically plausible model based on self-organization principle can compete with classical models using supervised learning and therefore it can be appropriate alternative in simulation of cognitive processes related to the tasks of sentence comprehension.

Keywords: sentence comprehension, self-organizing map, merge self-organizing map, simple recurrent network, language, connectionist model

Abstrakt

Jazyk a schopnosť komunikácie patria vďaka svojmu významu pre ľudstvo k jedným z najdiskutovanejších fenoménov vo vedeckých kruhoch. Snaha o detailenjší rozbor ich podstaty viedla k vytvoreniu samostatných vedeckých disciplín zameraných na špecifické atribúty, ktorými disponujú. Behaviorálne metódy boli kľúčovými vo vysvetlení niektorých špecifických znakov, avšak kvôli náročnosti ich dizajnu a častokrát etickým výhradám nemôžu byť aplikovateľné v niektorých prípadoch vedeckého skúmania. Výpočtové modely prinášajú v tomto ohĺade nové smerovanie ich výskumu.

V tejto práci bolo naším cieľom vytvoriť biologický plauzibilný model porozumenia jazyka na princípe učenia samo-organizáciou a jeho porovnanie s modelom výžívajúcim učenie s učiteľom v časti architektúry pre spracovanie viet. Na reprezentáciu významu jednotlivých viet sme použili samo-organizujúcu sa mapu. Ako komponent pre spracovanie viet prezentovaných po slove sme zvolili tzv. "zlúčenú" samo-organizujúcu sa mapu (Merge Self-Organizing Map). Referenčný model využíval pre túto úlohu jednoduchú rekurentnú sieť (Simple Recurrent Network). Komponenty na spracovanie viet po slovách boli natrénované, aby dokázali reprezentovať význam danej vety a to aj po predložení len jej časti. Modely sme testovali, ako dobre dokážu predpovedať význam danej vety po tom, ako im boli prezentované jednotlivé slová. Taktiež sme skúmali, ako dokážu modely rekonštruovať špecifické elementy reprezentujúce význam v danej vete.

Výsledky našich experimentov naznačujú, že model na princípe samo-organizácie dokáže uspokojivo predpovedať význam vety po predkladaní jednotlivých slov v porovnaní s referenčným modelom, pričom v predikčnej úlohe bol jeho výkon lepší oproti modelu s jednoduchou rekurentnou sieťou. V úlohe na rekonštrukciu sémantických elementov viet prezentovaných po slovách bol výkon modelu so samo-organizáciou porovnateľný s referenčným modelom. Vďaka tomu môžme konštatovať, že biologicky plauzibilný model na báze samo-organizácie dokáže konkurovať klasickým modelom využívajúcim učenie s učiteľom a môže byť preto vhodnou alternatívou pri simulovaní kognitívnych procesov súvisiacich s úlohami na porozumenie jazyk.

Kľúčové slová: porozumenie vetám, samo-organizujúca sa mapa, "zlúčená" samo-organizujúca sa mapa, jednoduchá rekurentná sieť, jazyk, konekcionistický model

Content

List of figures

List of tables

Introduction

Communication is one of the most fascinating and probably the most important tool which animals have. While the majority of animal world communicates with partially learned system (e.g. bird songs), or an innate ability to produce a limited number of meaningful vocalizations (e.g. bonobos). There is no other species than human which can express infinite ideas with limited set of symbols. Thus, it is naturally in sight of interest of many researchers over past decades.

Human language is a complex system of rules, symbols, and abilities to work with them appropriately. To fully understand such phenomena we need to include knowledge of neural systems in human brain, process of acquisition, social and cultural environment and other aspects. That leads to developing of many particular research fields which study specific nature of language (e.g. psycholinguistics, sociolinguistics, neurolinguistics, study of grammar, semantics, discourse and text analysis etc.). Naturally, every field uses different sophisticated methods to its study. In the early research typical were behavioral experiments with usage of invasive techniques. But there are several ethics and moral restrictions which lead to use mainly non-invasive techniques. However, some hypotheses cannot be tested directly (e.g. when we want to learn the effect of some substances in the brain such as neurotransmitters). Furthermore, sometimes we want to answer the questions which require time consuming research design (e.g. how the children acquire language). In such situations computational models can be helpful.

Nowadays, with massive improvements of computational power and progress in high level machine learning methods, the natural language processing becomes very popular (Kumar et al., 2016). Simply said, this field of computer science concerns to process massive natural language data and solve particular tasks such as language production and comprehension, sentiment analysis, speech recognition, text-to-speech transformation, translation and many more. Rapid progress of particular areas like deep learning (LeCun et al., 2015) or convolutional neural networks (Krizhevsky et al., 2012) encouraged the interest in natural language processing even more. However, these methods demand high computational power and very large datasets. In this sense, the research interests in area of computational modelling of language processes remains very popular and a demand for more accurate and less computational power consuming methods increases rapidly. Therefore, it is reasonable to investigate language comprehension using computational modelling.

Large amount of interests in NLP area is not only motivation for using computational modelling for research of language comprehension. Behavioural research can cover a broad area of language processes. Using neuro-imagining methods is an answer to variety of research questions in more detail. Nevertheless, there are many cases where the behavioural methods have their shortcomings. These are mainly related to the direct manipulation with human participants which can be dangerous for them (e.g. application intracellular in substances, direct stimulation of specific areas of the brain etc.). Here are the computational methods very helpful. Specifically, when the sufficiently accurate computational model is found out, it demands only manipulation with the parameters of the model which can be done easily by rewriting them in the source code of the program. Therefore, the manipulation with such parameters is very easy and even though it costs a lot of effort by testing, it is safe for human participants.

In our work we aim to examine language comprehension using connectionist modelling. This paradigm has several advantages such as distributed representation of many concepts or background in biological processes. We will compare two models of language comprehension using the artificial neural networks for representing the meaning of the text sentences and for the sentences processing task per se. We will focus to investigate this task in an interdisciplinary way through cognitive science field.

First chapter is devoted to theoretical background of language comprehension, the models that try to simulate this phenomena and artificial neural networks used in our architectures.

In the second chapter we will focus on technical explanation of our models' architecture.

Finally, the third chapter will contain the results of our research.

Chapter 1

Language comprehension

In this chapter we will discuss sentence comprehension in theoretical perspective. We will introduce research motivation and describe several models which try to examine the nature of this phenomena using computational and connectionist paradigm.

1.1 Comprehension of the language

Comprehension of the text is an important function of cognitive system which we can describe as an ability to extract information out of verbally and textually described situations. It has vital role in remembering, understanding and inferring. Hereby emerges several questions. How can we extract information from the plain text? What happens to it when it is extracted? And how can we infer from such extracted material? Several researchers tried to investigate these questions. Probably the biggest impact in this area had works of Van Dijk & Kintsch (1983) *Strategies of Discourse Comprehension* and Johnson-Laird (1983) *Mental models: Towards a cognitive science of language, inference, and consciousness.* Van Dijk & Knitsch postulated concept of ”situational model” while Johnson-Laired formulated ”mental model”. Both terms have similar properties. Simply speaking, they assumed that we create mental representations of what text is about. These representations are not just simple propositions extracted from the text but they combine previous experience and knowledge which assists to comprehension of the text and inference of what a text is about. Precisely, mental representation is abstract concept of the real world which emerges as a product of neural activity of human brain system. In our work we will use the term ”situational model” defined by Van Dijk & Knitsch as it is widely used in literature and in our opinion best describes this phenomenon. Before we describe situational model into the detail we shortly discuss other levels of representation of information in text comprehension.

1.1.1 The levels of representations in text comprehension

There are several theories which try to explain the situation model or so called mental model in different representations. Some of these deal with the comprehension from the text into the situation model itself. In situational level it can be simply described as:

Sentence -> Situational Model.

However, researchers differentiate two other representation of the text. Including situational representation there are (Van Dijk & Kintsch, 1983):

1. Surface representation

2. Textbase representation

3. Situational representation

Surface representation is a simple visual representation of the text itself which consists of literal meaning of particular words without relationships between them. The meaning of the text base representation can be expressed as a network of concepts and propositions from the text. The nodes in such network are connected according to the similar structural symbols (they share common, variable). Thus, we can call this propositional representation. As an example should be the sentence *Jacob loves Susan*. In the text base form we can express this sentence as LOVES[JACOB, SUSAN]. Finally, situational representation is the mental model constructed by the text base form. To better illustrate the relationships of these levels we construct following image.

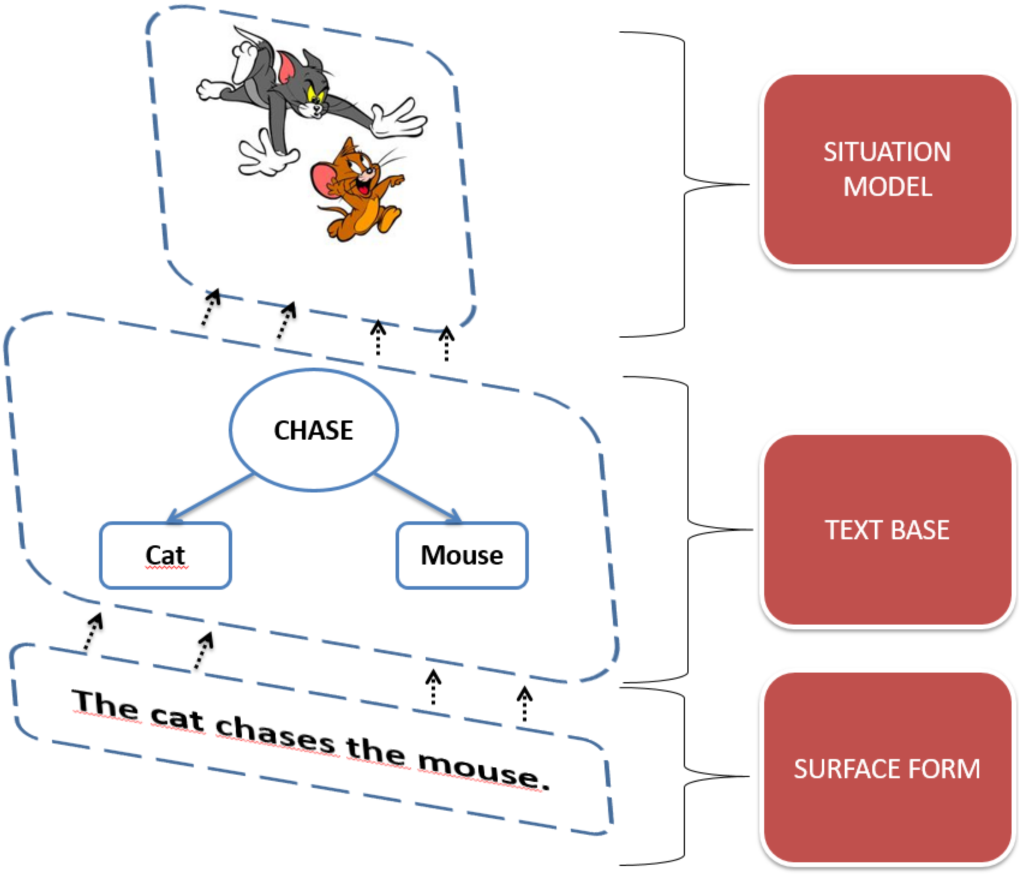


Figure 1.1: Inspired by (Van Dijk & Kintsch, 1983)

1.1.2 Situational models

The concept of situational models helps us understand how the situational information from the text is collected and integrated. Van Dijk & Kintsch (1983) argue that situational models are responsible for processes like translation, learning from multiple sources, domain-expertise, or completely understanding situations just by reading about them. These models are multidimensional in their nature. In this sense, some researchers claim that the weight of the dimensions shifts according to the situation which is described. Van Dijk & Knitsch explained the processing of new information by activation one of the dimensions according to the information. Situational model is changed accordingly to the situation described in the text. The bigger this change is more time the reader needs to comprehend. When the new information does not fit into the model, the reader tends to fail and he needs to re-read the text again. Next, we will explain several language processing tasks where the situational models find their application.

- Integration of information across sentences:

  E.g. "Barack Obama stays in front of some journalist. Public speaking is nothing special for the Ex-president of United States." The reader should know that these two sentences have the common subject - the same person - Barack Obama. The subject can be considered as token which allows to integrate two sentences into one coherent situational model.

- Explanation of similarities in comprehension performances across modalities

  By this example we can simply imagine that we read the newspaper, listen to the radio and watch late television where the same information is presented. Often it is easy to combine these information into coherent whole. Baggett (1979) found out that students who saw a short film and students who heard a spoken version of the events in the short film finally produced a structurally similar recall protocol. Further, (Tanenhaus et al., 1995) examined if the visual context influences spoken language comprehension. They concluded that visual spoken word recognition and sentence processing are influenced by visual context, even during the earliest phase of sentence processing.

- Domain expertise on comprehension

  Situational models have influence on effects of domain expertise on comprehension. Specifically, experts in particular domain use their knowledge stored in long term memory when they create situational models during comprehension of a text while novices in a domain have only the text (Ericsson & Kintsch, 1995). This idea was supported by Schneider & Körkel (1989). They realized an experiment with 3rd, 5th and 7th soccer experts[1]. The participants had to recall units from the text they have read. The 3rd grade experts recalled 54% of units with compared to 42% by the 7th grade novices.

- Explanation of translation skills

  Translation of a text is not simply done by translating each word until we construct some sentence structure which seems to be sound. We have to consider a right meaning of a sentence or a text before we build up appropriate translation. R. Zwaan et al. (1998) provided an experiment where participants had to translate sentences from French into English. They capitalized on the fact that the French does not have a neuter pronoun, whereas theEnglish does. They found out that participants which considered meaning of the text across several sentences were more successful in translation task. Furthermore, more fluent English speakers were better in comparison to less fluent English speakers. Such findings supported assumptions of effect of situational models in both domain expertise and translating of a text.

[1] The lower grade number indicates the higher domain knowledge.

- Multiple source learning

  and reasoning from multiple sources. Perfetti et al. (2012) provided an example of how situation models are needed to be taken into account for text-based learning and reasoning about historical events. They argue that people can form text base for each material they read. Nevertheless, the actual learning and reasoning come into account when information from the documents are integrated into situational models.

Dimensions of situational models

As it was mentioned previously, many researchers claim that situational models are multidimensional in their nature. In this section we will shortly describe particular dimensions of situational models. For deeper insight into this topic we recommend the work of R. A. Zwaan & Radvansky (1998).

1. **Space** – Comprehensive linguists make mental models of protagonist's perspective (e.g. when they know the space/place of the story, like building, they comprehend better). Also, when they know properties of the space (e.g. sizes, locations, shapes, spacial layout).

2. **Time** - We a priori assume the chronological order of the story. When the sentences do not preserve chronological order, it takes longer time to comprehend such text. Study of Münte et al. (1998) shows that "before" [2] sentences elicit greater negativity than "after" sentences during Event-Related-Potential measurements.

3. **Causation** - As we interact with the environment, we have a strong tendency to interpret event sequences as causal sequences. It is important to note that, just as we infer the goals of a protagonist, we have to infer causality - we cannot perceive it directly.

4. **Intentionality** - We are often able to predict people's future actions by inferring their intentionality, i.e. their goals. (e.g. man walks to the chair after he stands for a long time -> he wants to sit.).

5. **Protagonists and objects** – Comprehensive linguists are quick to make inferences about protagonists, about the emotional states of characters etc. In study of Carreiras (1996) the reading speed of participants was slowed down when the mismatch with the stereotypical gender of electrician occurred. Specifically, the sentences "The electrician examined the light fitting. She took out her screwdriver" showed slower reading speed comparing to the second sentence "He took out his screwdriver".

[2] Indicating that an event has happened before something that will be explained next.

1.2 World knowledge

In modelling of higher cognitive processes like text comprehension need to be implemented so called *world knowledge*. Since no realistic amount of such information can be made available to a model, researches had to find out how they can be input. Frank et al. (2008) described four types of world knowledge implementation:

- Disregarding world knowledge
- Ad hoc selection of world knowledge
- Extracting knowledge from text corpora
- The microworld strategy

First type of implementation of world knowledge is not appropriate by creating appropriate model of text comprehension. However, for particular low-level sub-processes, the influence of world knowledge may seem small enough to be ignored. Such model simulated only sub-processes is *The Resonance model* (Myers & O'Brien, 1998). This model attempts to simulate the process called *reinstatement*, which is reactivated previously background text when it is required, while central concepts and propositions of the text remain in working memory. Since this model was not intended for a functioning with a large amount of added knowledge, it cannot deal with the highly connected network resulting from including semantic relations (Frank et al., 2008).

Next technique of adding world knowledge takes pieces that seem relevant to the particular text under consideration and only those will be provided to the model. This prevents technical problems with running a model in the context of large amounts of knowledge, but introduces free parameters which are set by the modeler on an ad-hoc basis. However, these parameters can be considered as an input to the model rather than part of the model itself. Therefore, the models of text comprehension should avoid free parameters taking into account the plausibility of the model.

The Construction-Integration model (Kintsch, 1988) uses this type of adding world knowledge into the model. This model consists of two phases: *construction*, where world knowledge potentially relevant to the text is selected, and *integration*, where inappropriate materials are discarded. This model, however, introduces many free parameters and with appropriate setting can give almost any desired output (Frank et al., 2008).

Another way to include world knowledge into the model is computing word representations that encode semantic relations among the words. That computation can be done from large text corpora. *Latent Semantic Analysis* (LSA) is an example of technique which can accomplish this task Landauer & Dumais (1997). Here each word is represented as a vector in high-dimensional semantic space. The similarity between two word vectors is measured by $\cos(X, Y)$ where X and Y are two word semantic vectors. The result is value $< -1, 1 >$ and indicates semantic relatedness of the two words. However, we cannot extract the meaning of the whole sentence what is more important in text comprehension (Frank et al., 2008).

The last strategy of implementation of world knowledge is the use of microworld. Because this topic is closely related to our own models we will pay more attention to it.

1.3 The microworld strategy

At this moment the reader can assume that modelers have to consider several reflections implementing world knowledge into the model. There should not be too many free parameters because it decreases credibility of the model. The generalization to the unseen texts would be problematic as well. Since the model can operate with constrained computational power, the world knowledge should be trimmed off useless information. However, the last point should be considered carefully because humans can be very effective in selection important information to the current problem and in the end of the day we try to simulate cognitive processes in humans per se. Furthermore, the constrained world knowledge is valuable since the modeler has higher control in the process of simulation, therefore it can increase general interpretability of the simulating mechanism.

The microworld strategy helps to solve the issues mentioned above. Frank et al. (2008) suggest that corpus included world knowledge should not consist of texts but of events or situations in the world. But such corpora are not readily available and need to be build up from scratch, therefore, it can bring to the modeler even higher control of simulating process since he/she has to construct it. As a result a tiny subset of real world situations in which the knowledge is encoded will be provided to the model. This subset is called *microworld*.

In the microworld all knowledge is encoded in fixed parameters. That means their values remain the same whatever the model's input is. Thus, they can be viewed as part of the model itself. Therefore, such models can be trained on different texts, while the modelers can stay certain that the results are not caused by their interference. One of the model which uses microworld strategy is called *Distributed Situational Space model* (Frank, 2004, 2005; Frank & Haselager, 2006; Frank et al., 2008).

Distributed situational space

The model of distributed situational space (DSS) can be described as followed: There are limited possible events (situations) which can occur in the microworld. Some of the events are more probable. But, there are also some restrictions on the events and their protagonists like in the real world. For example, a boy can play computer only inside, not outside. The events are than represented as n-dimensional binary vector where 0 indicates basic proposition which is not the case in the situation, and 1 a proposition that is present. Since particular propositions can be combined, the number of situations is a result of all possible combinations, depending on the constrains defined by the modeler. These situational vectors are then used as the train data for the *Self-Organizing Map* (SOM) (Kohonen, 1998). As a result, each basic proposition is represented by a pattern of activation over the cells of the SOM. Specifically, the situations are encoded as m-dimensional vector of SOM units' activation, where m is the number of neurons in SOM. These vectors are therefore called *situational vectors.*

As the authors claims, to each SOM unit and basic situation is associated membership value $\mu_i(p) \in < 0, 1 >$ that indicates the extent to which cell i forms part of the representation of p. If the SOM has m units, the representation of p can be viewed as a m-element situational vector of membership values $\mu(p) = (\mu_1(p), ..., \mu_m(p))$ (Frank & Haselager, 2006). Then we can express a priori probability that the situation occurs as follows:

Let $(x_1, ..., x_m)$ be a vector represented some situation in microworld. Then

$$\tau(X) = \frac{1}{m}\sum_{i}^{m} x_i \tag{1.1}$$

is the a priori probability of event. The DSS allows to extract the content of any situation X by comparing its representation to several known situation vectors $\mu(p)$. With adding the rules of fuzzy logic, the estimated conditional probability that some p is the case in situation X is:

$$\tau(p|X) = \frac{\tau(p \wedge X)}{\tau(X)} = \frac{\sum_i^m m_i(p)x_i}{\sum_i^m x_i} \tag{1.2}$$

These τ-values are called *belief values* and they are accurate estimates of (un)conditional probabilities in the microworld. That can be consider as a prove that relations among microworld situations are encoded (implicitly) in the organization of situation space (Frank & Haselager, 2006; Frank et al., 2008). This properties of DSS are very useful and to some extent we use them in our model.

1.3.1 Semantics of the sentences

Encoding meaning of the sentences is an ongoing question in connectionist models of sentence comprehension. In our work we use the method proposed in Takac et al. (2012). The authors suggest that meaning of the sentences can be represented as structured sequences of semantic elements, whose structure reflects the sequential framework of the sensorimotor (SM) routines through which they are experienced. These sequences have canonical structure, so the semantic roles like Agent, Patient or Action are associated with specific serial positions. The whole idea has its core in embodied cognition paradigm. The models based on this paradigm argue that the high-level semantic representations may reflect the SM routines through which these representations were obtained. Ballard et al. (1997) find out that the agent's interaction with the world often take specific form of short sequences of SM processes which structure is defined internally. They called these sequences *deictic routines*. This concept was studied and explained into the detail in Knott (2012) on elementary transitive action: reaching to grasp a target. In summary, whether the participant is executing or perceiving the action, attends the protagonists of the situation in following order:

1. Attention to the agent - activating a representation of the agent
2. Attention to the target[3] - activating a representation of the target
3. Activation of a reach/grasp motor programme - reattending to the agent in the process
4. Reattending to the target - this is done at the end of the action, when a stable grasp is achieved.

Because we use this concept in the encoding the meaning of the sentences we use the sentence ”*Man grasps a cup*” as an illustrative example for better understanding (we will explain the process of encoding and training into the detail later). Semantic roles in this sentence are:

- *Agent* : Man
- *Action* : Grasps
- *Target* : Cup

Let's imagine that we watch this situation as observers. Our cognitive system firstly activates representation of the ”Man”. Then we focus attention to the ”cup”. After that we activate motor program ”Grasps an object”. Finally, when the action is done, we are reattending to the ”cup” again. Activation of a motor program while we observe the action is presented not only in humans but also in other primates. For that activation are responsible specific types of neurons called *mirror neurons* (Kohler et al., 2002).

[3] Another name for this semantic role is a patient or simply an object

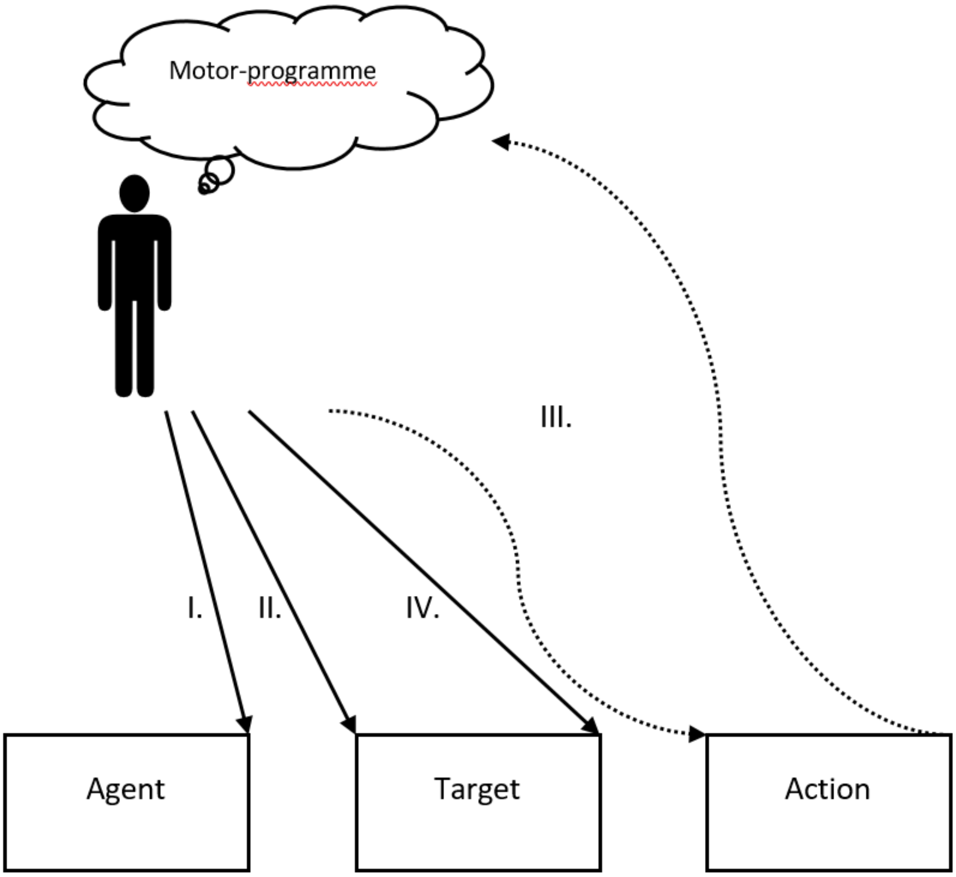


Figure 1.2: Ilustrative example of deictic routine: I. activating representation of the agent; II. activating a representation of the target; III. activation of a reach/grasp motor programme; IV. reattending to the target.

Thus, we can use the properties of DSS or specifically of SOM to encode such sentence into representation of its meaning.

1.4 Artificial Neural Networks

Artificial Neural Networks (ANN) are computing system based on connectionist paradigm. Their theoretical background comes out from biological systems in the brain. They simulate the core units of biological neural architecture - neurons. The artificial counterpart of connections between biological neurons (throughout axons and dendrites) are represented as weights between particular units. Individual artificial cells can transform the signal through their weights. Since their investigation at the end of 60's they found application in many areas both in academic or practical areas (e.g. speech recognition, computer vision, image processing, social network filtering and many more). As it is in general machine learning, we know two types of ANN based on the type of the training process (Haykin et al., 2009):

1. *Supervised networks* - the target to these networks is known. Thus, an error of the network's output is known and the weights are updated to get desired target accordingly. This is usually done by backpropagation algorithm. Example of such network is perceptron, simple recurrent network etc.

2. *Unsupervised networks* - the target to these network is unknown. Here we let the network to create its own representation of training data. We call this type of training process *self-organisation.* The well-known example of such network is Self-Organizing Map.

In the next sections we will describe three types of ANN which we used in our models. They are:

- Self-Organizing Map (SOM)
- Merge Self-Organizing Map (MSOM)
- Simple Recurrent Network (SRN)

1.4.1 Self-Organizing Map (SOM)

Self-Organizing Map can be considered as self-organizing mapping of high-dimensional data into 2-dimensional representation while it preserves topology of layout of input vectors (Kohonen, 1998). Typical representation of the SOM is in 2-dimensional grid where the nodes indicates particular SOM units, cells or neurons. The learning principle of the SOM is mapping input vectors into weight vectors. While the network is able to capture similar patterns between input vectors and weight vectors, it arranges the units according to their similar weight vectors - the units with smaller distance between their weight vectors are arranged close to each other. We differentiate three phases in the training process of the SOM (Haykin et al., 2009):

1. Competitive phase
2. Cooperative phase
3. Adaptive phase

The unit with weight vector most similar to particular input vector (their Euclidean distance is the lowest across all pairs $(unit - input)$) wins the competitive phase and the unit goes closer to particular input vector. This phase can be expressed by following equation:

$$i(\vec{x}) = \arg\min_j \parallel \vec{x} - \vec{w_j} \parallel^2; j \in N \tag{1.3}$$

where i is a function of input vector x that returns the lowest Euclidean distance between input vector x and weight vector w_j. N is the number of neurons in SOM.

After finding winner neuron learning process continues to cooperative phase. The goal is to find the neighbors of the winner and shift them according to their closeness to the winner. The closest units to the winner are shifted more than the cells far away from the winner. This process is expressed by equation:

$$h_{j,i(x)} = exp\left(- \frac{d^2_{i,j}}{2\lambda^2} \right) \tag{1.4}$$

where h is gaussian function that favors the units closer to the winner, d is Euclidean distance between j-the unit and i-the input, and λ helps to decrease resulting value of h during the ongoing training process. That causes stabilization of the network.

Finally, the weights are updated according to following equation:

$$\triangle\vec{w} = \alpha h_{j,i(\vec{x})}(\vec{x} - \vec{w}(t)) \tag{1.5}$$

where α is a learning rate, h is gaussian neighborhood function, x is input vector and w is weight vector of particular neuron.

The mechanism in the last phase enables the neurons which are excited to increase their individual values of discriminant function in relation to the input pattern. This is done by suitable adjustment of the neurons applied to their synaptic weights. This phase can be divided further into *ordering* and *convergence* phase. In ordering phase the large amount of neighbours of the winning neuron are included in the topological ordering while in phase of convergence is the neighborhood function reduced to small value and only small number of neurons (if any) around the winner are adjusted.

Learning rate α and parameter λ

As the process of learning proceeds, the learning rate α and λ parameter needs to decrease. Decreasing learning rate causes stabilization of the network while decreasing of the λ reduce the neighborhood of the winner neuron that is updated. Therefore, in the later phase of the learning only the small number of neighbours are updated with the winner. The λ gradually decreases according to (Ritter & Kohonen, 1989):

$$\lambda(t) = \lambda_0 \left(\frac{\lambda_f}{\lambda_0} \right)^{\frac{t}{t_max}} \tag{1.6}$$

where λ_0 refers to initial setting of parameter λ, λ_f is the final value to which the λ aims, t indicates current training epoch and t_max is maximum number of training epochs. Learning rate α can be updated according to the same rule. There are also other possible ways how these values can be updated (see Haykin et al. (2009); Van Hulle (2012))

1.4.2 Merge Self-Organizing Map (MSOM)

Next type of ANN is similar to previous one. Difference between them is in storing a context - representation of material what has been seen so far. This kind of ANN are called *recurrent*. In general, as a recurrent can be considered whatever ANN in which there is subset of neurons that stores information about activities in the past. They perform the same task for every element of a sequence, with the output being depended on the previous computations. They have a ”memory” which captures information about what has been calculated so far. The main area of using their properties are serial processing tasks (e.g. processing sequences like letters in words, words in sentences etc).

The learning architecture of Merge Self-Organizing Map implements a compact back-reference to the previous winner with separately controllable contribution of the current input and the past with arbitrary lattice topologies (Strickert & Hammer, 2005). Thus, the neurons in MSOM specialize not even on the input data but also on the previous winners, then neuron activation orders are established. Therefore, order is coded by a recursive position of the current input and already trained neurons.

The context of Merge SOM

The MSOM context can be characterized as a fusion of two properties reflecting the previous winner. First, the weight of its units and the second, context of last winner neuron. These two parts are merged by a weighted linear combination. During MSOM training, context of last winner neuron (context descriptor) is kept up-to-date and it is used as the target for the context vector c_j of the winner neuron j and its neighborhood. By target we mean that the combined vector of weights w_j and context c_j of neuron j is adapted into the direction of the current input pattern and context. Definition of MSOM is the following (Strickert & Hammer, 2005):

MSOM network is composed of neurons $N = \{1, \ldots, m\}$ which are equipped with a weight $w_i \in \mathbb{R}^d$ and context $c_i \in \mathbb{R}^d$. The best matching unit $i(x)$ (or winner neuron) has minimum recursive distance between given sequence entry $x(t)$ and the context descriptor $c(t)$:

$$i(\vec{x}) = \arg\min_{j} \big((1-\beta)* \parallel \vec{x}(t) - \vec{w_j} \parallel^2 + \beta* \parallel \vec{c}(t) - \vec{c_i} \parallel^2 \big); j \in N \tag{1.7}$$

Contributions of weights and context are balanced by the parameter β. The context descriptor

$$\vec{c}(t) = (1-\gamma) * \vec{w}^{I_{t-1}} + \gamma \vec{c}^{I_{t-1}} \tag{1.8}$$

is the linear combination of the properties of winner neuron I_{t-1} in the last time-step. We set initial c_1 to a fixed vector, e.g. 0.

Finally, the weight and context vector are adapted towards the current input and context descriptor

$$\begin{aligned} \triangle\vec{w_i} &= \alpha_1 h_{j,i(\vec{x})}(\vec{x}(t) - \vec{w_i}) \\ \triangle\vec{c_i} &= \alpha_2 h_{j,i(\vec{x})}(\vec{c}(t) - \vec{c_i}) \end{aligned} \tag{1.9}$$

α_1 and α_2 are learning rates and h is Gaussian shaped function, as it was in SOM.

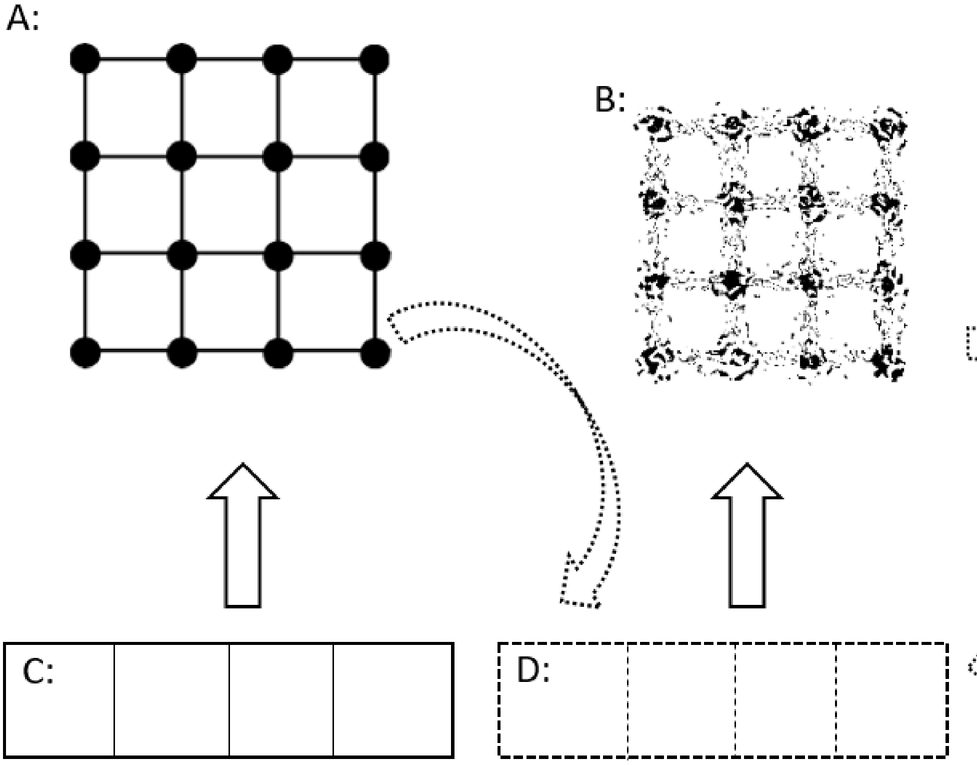


Figure 1.3: MSOM architecture: A: weight space ($\vec{w_i} \in \mathbb{R}^d$); B: context space ($\vec{c_i} \in \mathbb{R}^d$); C: current input vector ($\vec{x}(t)$); D: context descriptor $\vec{c}(t)$. Solid lines indicate finding a best matching unit. Input vector is comparing with vectors of weights and context descriptor with vectors of context weights. Dashed lines represent updating context descriptor in time step $t+1$ according to best matching unit I^{t+1}

1.4.3 Simple Recurrent Network (SRN)

The last ANN that we used was proposed and designed by Elman (1990, 1991). Similar to the MSOM, Simple Recurrent Network (SRN) is recurrent type of ANN but unlike the previous one it uses target output in the learning process. Thus, we consider it as supervised ANN. However, in this network is the learning process different than in the classical ANN like perceptron or multi-layer perceptron (MLP). While MLP uses connections between neurons in direction from input to output, recurrent ANNs can have connections in backward direction or even inside a single layer. In principle, output values of recurrent neurons in time $t+1$ are copied onto the context neurons. Next, they will join to input vector in time t. Thus, this copying process is ongoing with one unit delay and that causes expanding of network with *internal memory*. Those properties increase the possibility of using these networks, however, it has also impact on the computational capacity because of more complicated training process.

Simple Recurrent Network has the recurrent connections located in hidden layer (see figure 1.4). Usually they are trained using algorithm called *Back Propagation Through Time* (BPTT) (Werbos, 1990). Specifically, activations on the individual layers are computed using following equations

$$h_k(t+1) = f_h(\sum_j w_{kj} x_j(t) + \sum_l c_{kl} h_l(t)) \tag{1.10}$$

for the hidden state activations, where w are weights connecting input layer with hidden and c are weights connecting context units with hidden layer in time step $t-1$, and

$$y_i(t) = f_y(\sum_k v_{ik} h_k(t)) \tag{1.11}$$

for the output activations. Here v are weights connecting hidden layer with output units and h_k is activations of the hidden units in time step t.

f_h and f_y are activation functions, usually set to sigmoid

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{1.12}$$

tangent-hyperbolic

$$tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{1.13}$$

or softmax function

$$softmax_i(x) = \frac{e^{x_i}}{\sum_j^J e^{x_j}}, for\ i = 1, \ldots, J. \tag{1.14}$$

The learning process can be described as follows:

1. SRN is getting sequences of the patterns from training data as an input to the network. These sequences can be variable in length.

2. The network expand through the time. It has as many hidden layers as it is the length of an input sequence.
3. That expanded SRN is then trained by back-propagation algorithm. Simply said, desired target is compared to the actual output of the network. Local gradients are recorded during every single forward pass. Computed error is then back-propagating (spreading backwards) and the weights are updated according to desired output (see Elman (1990); Werbos (1990) for the detail).

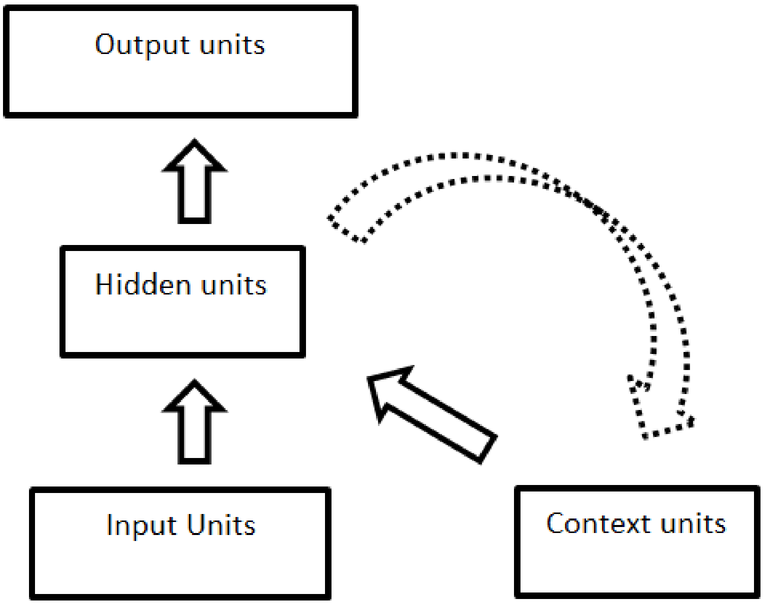


Figure 1.4: SRN architecture: signal is spreading in forward direction. Activations of hidden units in time step $t + 1$ are copied onto the context neurons.

Chapter 2

Models

In this chapter we will introduce two connectionist models of sentence comprehension that we created. Each of them included Self-Organizing Map for representation of situational vectors of the text. In addition, first uses Simple Recurrent Network for encoding a sequence of words to the meaning of the situations that they represent. The second one uses Merge Self-Organizing Map for the same task as SRN in the first model. Thus, we compare two types of architectures where one combines supervised and unsupervised type of learning while the other one uses only unsupervised type of learning. As it is known, self-organization is a common process in human brain, occurring in sensorimotor cortex by movement (Jirsa et al., 1998), visual perception (Gray & Singer, 1987; Gerstner & Kistler, 2002), dreaming (Kahn, 2013) and more (see Singer (1986); Kelso (1997) for further reference).Therefore, we claim that such model can be biologically plausible and can be used for further research.

2.1 Microworld

We mentioned in previous chapter that representing situations described in a text is an important question for validity of the connectionist model of text comprehension. In our work we are inspired by methodology proposed by Frank (2005) where the author used microworld strategy for importing world knowledge into the model as well as representing situations which are described in a text. We also base on the assumption that during comprehension humans create structured representation of the situation which is described in the text. These representational sequences have canonical structure as it was described in Takac et al. (2012). Next, we will describe our method into the detail.

Our training data was composed from sentences of different lengths. The shortest sentences have only two words containing a subject and an action that a subject performs. Maximum length of the sentences is seven words. The sentences describe basic situations and they do not form coherent story. It means that individual sentences do not follow on the previous text. Each sentence contains a *subject* and a *verb* an optional *object*. Particular protagonists in the sentences can have two properties: *size* and *color*. Both of them are optional (subject or object can have it but it is not a condition). Other attributes which the words can have are those which are in their nature but they are not described in the text. These attributes are included in situational vectors of the sentences that these situations describe. Specifically, nouns can be *human, animal* or *item* and verbs can be *transitive* (there occurs an object/patient with them) or *intransitive* (there are no object/patient). Table 2.1 shows possible words that can occur in the text.

Table 2.1: Words that can occur in the text

Class	Property	Words							
Nouns	Human	Man	Woman	Jacob	Susan				
	Animal	Dog	Cat	Mouse					
	Item	Cup	Chair	Ball	Bottle				
Verbs	Transitive	Grasps	Hits	Pushes	Sees	Catches			
	Intransitive	Walks	Sneezes	Runs	Arrives				
Sizes		Small	Medium	Big					
Colors		White	Yellow	Black	Blue	Red	Brown	Pink	Green

2.1.1 Sentence meaning

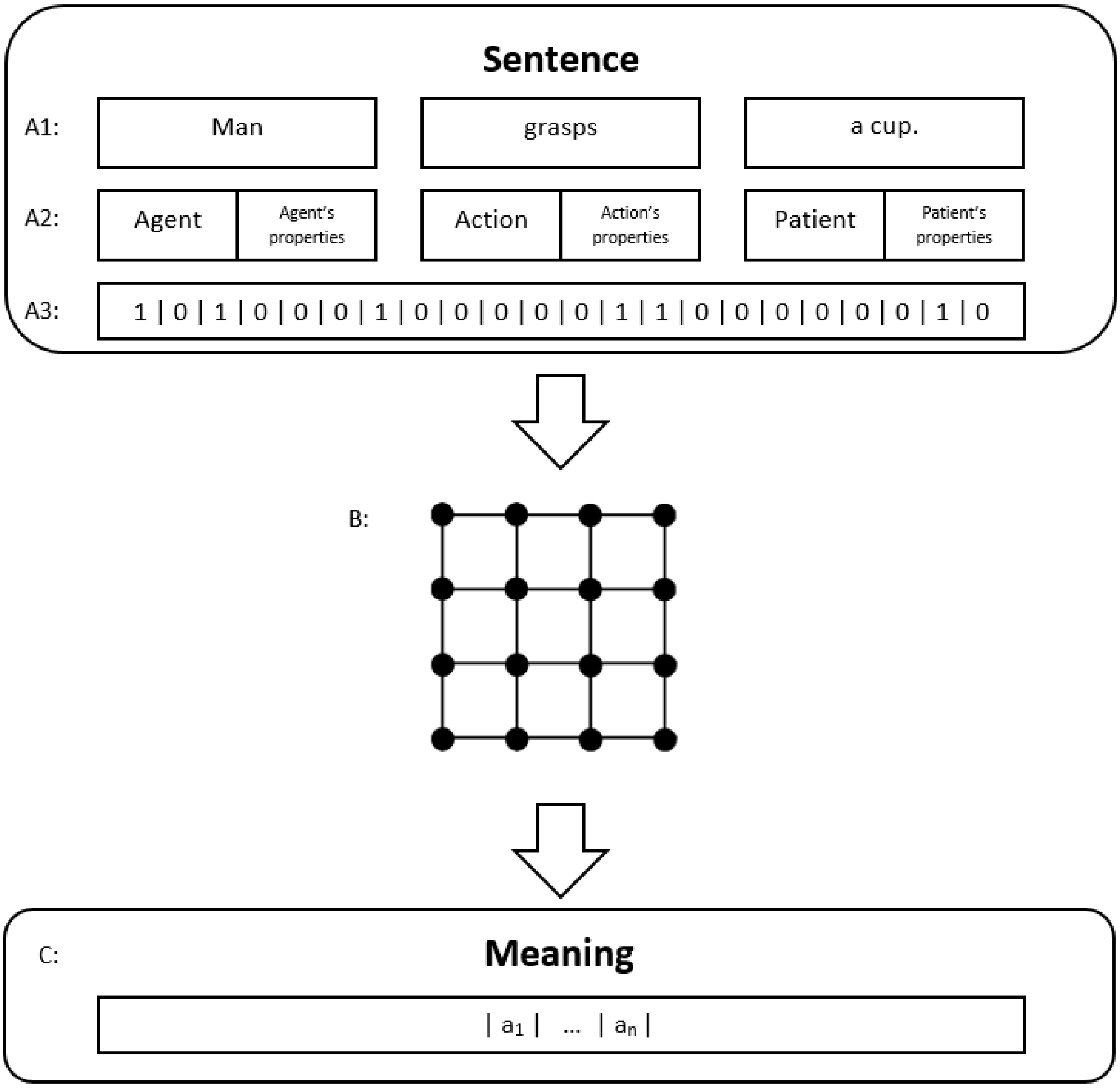


Figure 2.1: Mapping sentences to their meaning: A1: Text of the sentence. A2: Semantic roles of the elements of the sentence. Each element can have additional property (noun - human, animal, thing; verb - transitive, intransitive). A3: Symbolic representation of the sentence meaning. Binary vector where 1 indicates that element is present, 0 that it is not. B: Self-Organizing Map trained by binary vectors representing the sentences. C: Subsymbolic representation of the sentence meaning. They can be considered as the distributed situational vectors represented by activation of the cells of the SOM.

Subject - Agent

In a sentence, every verb must have a subject. If the verb expresses action like sneeze, catch or see — the subject is *who* or *what* does the verb. We recognize two types of subjects:

- *The complete subject* is who or what is doing the verb plus all of the modifiers (descriptive words) that go with it. E.g. The big white man grabs a cup. The man is the agent who grasps a cup. The complete subject is then **the big white man**.
- *The simple subject*, on the other hand, is who or what is doing the verb without any description. From above example, ”big” and ”white” are just descriptive words that differentiate particular man. The simple subject is therefore **the man**.

There can be seven possible subjects in our text: a man, a woman, Jacob, Susan, a dog, a cat, and a mouse. As we can see in the table 2.1 they have natural property, specifically, they can be a human type subject or animal type subject. These properties are not expressed in a text explicitly. However, in situational representation of a sentence are these properties included (more on this topic later).

Verb - Action

Every sentence must have a verb. There are nine possible verbs that can occur in the text: to grab, to hit, to push, to see, to catch, to walk, to sneeze, to run and to arrive. Regarding whether the sentence has an object the verb can be *transitive* (there is an object) or *intransitive* (there is no object).

Object - Patient

An object in a sentence (patient) can be every noun from our list (see table 2.1). Instead of subject part of a sentence, an object also can be the noun which has an *item* property. Similar to human or animal type nouns, that property is not expressed in a sentence explicitly. It is rather represented in a situational vector of the sentence.

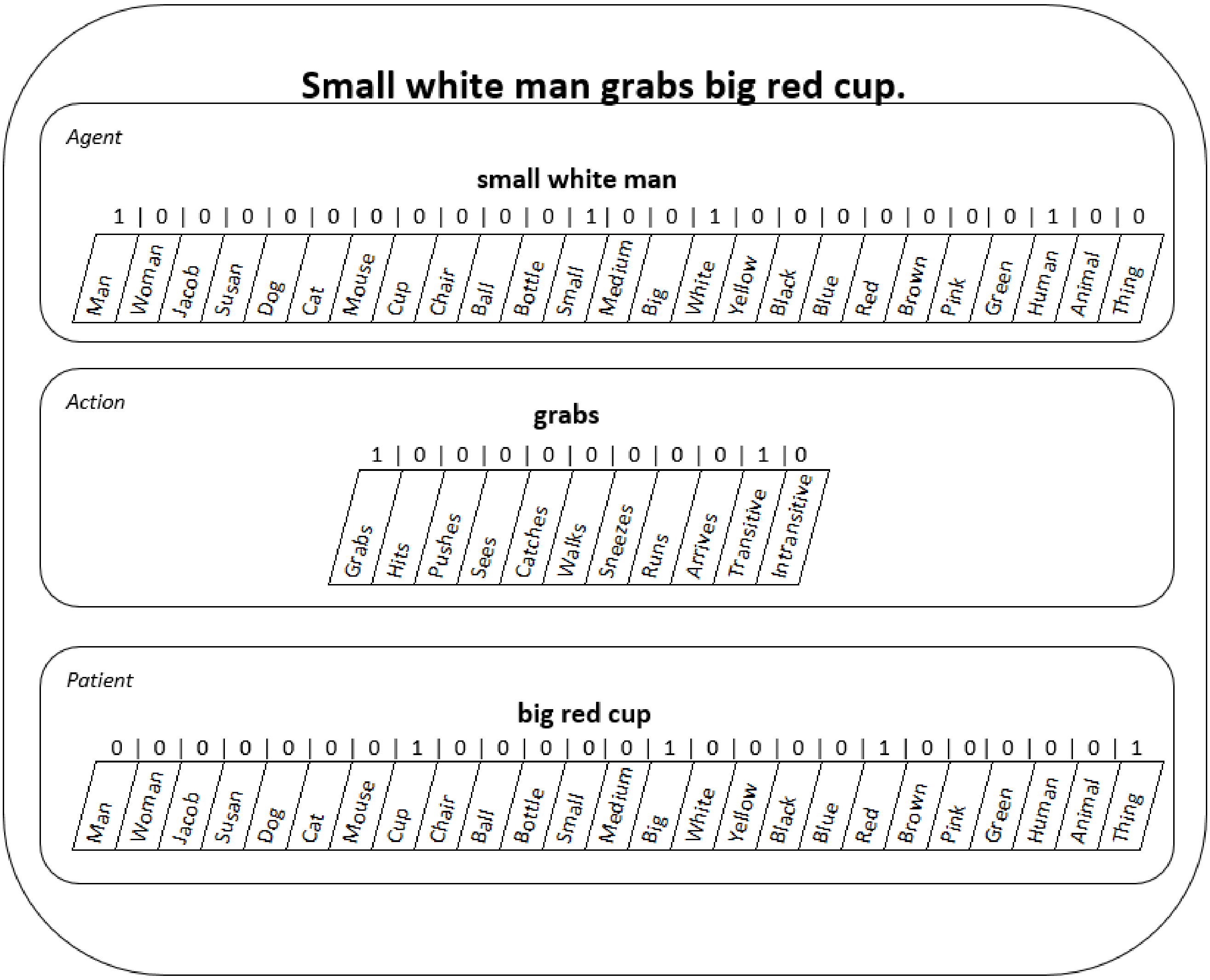
Symbolic representation of sentences’ meaning

In chapter 1 we discussed the problem of encoding situations which are described in text. In our work we base on methodology used distributed situational space. (Frank, 2005; Frank et al., 2008). We use the properties of Self-Organizing Map to encode meaning of the sentence into patterns of activity of its neurons.

Each individual sentence can be between two up to seven words long. Meaning of every sentence contains *an agent* and *an action* semantic element. *An object* element is optional. Each sentence is then encoded into symbolic representation. That means, there are k-dimensional binary vector where 1 indicates that element is present, 0 that it is not. These vectors are symbolic representation of the meaning of sentences. Specifically, each sentence’s meaning can be divided into separate parts according to individual semantic elements (agent, action, patient). Since there are eleven possible nouns, both agent and patient part are 11-dimensional binary vector a priori. We do not restrict agent part from the item nouns which cannot be agents. Further, agent and patient of the sentence can have additional properties that can be expressed in a sentence explicitly or they occur in their nature. Explicit properties can be *size* or *color*. They may occur simultaneously, one by one, or they may be absent. Implicit property of an agent or patient means that they can be *a human, an animal* or *an item*. Figure 2.2 shows symbolic representation of an example sentence's meaning. This type of representation is similar to localist type where each element (e.g. a boy) is encoding by one neuron. However, in distributed connectionist models the activation of a particular neuron may not have specific interpretation. The entities are rather represented by sets of neurons (*population coding*). As it was mentioned earlier, microworld strategy offers really good opportunity to control modeling process. Thus, before we explain the process training any further we should mention some details about the training text and restrictions which we have defined.

- The sentence meaning can be created by combination of particular semantic elements. Specifically, an agent, an action, and a patient.
- An agent and an action are mandatory parts of the sentence meaning. A patient is optional.
- Minimum length of a sentence is two words. Maximum length is seven words.
- Object type of nouns cannot stand as an agent in a sentence.
- Animal nouns cannot be combined with intransitive verb ”to sneeze”.
- Animal nouns can be combined only with transitive verbs ”to see” and ”to catch”.
- Transitive verb ”to grasp” can only be combined with object noun in the role of a patient.
- Human nouns are constrained to colors ”white”, ”black”, and ”yellow”.
- Human nouns can be only ”small” and ”big”.
- Animal and object nouns can acquire any color or size.
- Each element can only have one property from particular category. E.g. Jacob who stands as the agent in the sentence can be small and black, but he cannot acquire more than one size (small and big) or color (white and black). He is always a human noun.
- Colors and sizes are optional properties. Nouns can have them but it is not a requirement.

- Noun properties, human, animal, and object are mandatory. Each noun is in one of these categories. Every noun can only be in one such category.
- Verb properties transitive and intransitive are mandatory as well. Each verb (or action) is in one of these categories and they are mutually exclusive (if the verb is in one category, it can be in the other one).



Man	Woman	Jacob	Susan	Dog	Cat	Mouse	Cup	Chair	Ball	Bottle	Small	Medium	Big	White	Yellow	Black	Blue	Red	Brown	Pink	Green	Human	Animal	Thing
1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0

Grabs	Hits	Pushes	Sees	Catches	Walks	Sneezes	Runs	Arrives	Transitive	Intransitive
1	0	0	0	0	0	0	0	0	1	0

Man	Woman	Jacob	Susan	Dog	Cat	Mouse	Cup	Chair	Ball	Bottle	Small	Medium	Big	White	Yellow	Black	Blue	Red	Brown	Pink	Green	Human	Animal	Thing
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1

Figure 2.2: Symbolic representation of a sentence meaning. For imagining reason the sentence is divided into separate parts. In modeling, the symbolic representation of the sentence is one k-dimensional binary vector ($1 =$ element is present; 0 otherwise) combining all three parts of the sentence - if a patient is not present his part is represented as vector of all 0's of.

Situational vectors of sentences

Models of language comprehension which are based on symbolic paradigm lack various features that are present in language. Perhaps the main drawbacks of such models are that they do not perform well when it comes to semantics, graded constraints satisfaction or learning. However, connectionist models perform in such tasks better since they add robustness into the system. As mentioned earlier, they represent the patterns rather numerically than by symbols. That comes out from the nature of neurons communication between each other.

In our models the meaning of the sentences is represented in activations of SOM's neurons. Specifically, the sequences of words that build up the sentence are comprehended through creating situational representation which they are describing (Van Dijk & Kintsch, 1983). Because these situations share some similar features we can represent them by specific method which takes into consideration these similarities. It may be obvious to the reader that SOM can be used for such task. Thus, such situations can be encoded into k-dimensional vector of activities of SOM neurons where the representation of the meaning is hidden. This whole process consists of following steps:

1. Encoding sentence meanings into binary vectors of $0, 1$ where 1 indicates presence of the element and 0 absence.

2. Creating training data from encoded sentences.

3. Training SOM to represent sentence meanings in a compressed form.

4. Computing activity of the SOM neurons after presenting them the sentences each by one.

First step of the training process has been already introduced in previous section. Figure 2.2 shows it into the detail. Note, that the whole sentence is divided into particular semantic elements for better imagination. One training unit is a combination of all these elements.

Training data consists from permutation of binary vectors which number is equal to number of sentences that we chose as our training text. Similarly to the process by which the child learns to comprehend language, the sentences [1] can occur in training dataset number of times. This brings another feature into the representation of the meaning. The sentence meanings which have occurred in the training set more times, are more pronounced anchoring in pattern of activity of SOM neurons. This means, that presenting such sentence meaning to the SOM will follow to higher activations of particular cells or fields in the SOM lattice.

Next step goes further right into the training of the SOM. Particular binary vectors are presenting to the SOM each by one. Each vector are presented to the SOM once

[1] Here we admit that it would be better to speak about *situations*. However, it may bring mislead into understanding the explanation. Therefore, we write about "*sentences*" in process of learning of comprehension, instead of "situations".

during one epoch[2]. After completing the training process, the information from the ”text” vectors is stored in the layout of SOM lattice. Specifically, some cells or broader area encode similar sentences in accordance with their symbolic vectors. For example, the neurons that store information about the human type nouns or particular subject like *Jacob* are closer to each other. However, in some lattice layout more areas of neurons can appear which encode similar patterns. Figure 2.3 shows representation of example sentences.

Finally, we want to extract semantic information which SOM encodes. Similarly, as Frank proposed distributed situational space (Frank, 2005), we want to get representation of the meaning of the text sentences. Instead of representation of sentences, which is symbolic[3], representation of their meaning are on sub-symbolic level. Because of that, the SOM can represent more concepts at the same time as probability distribution, where particular elements of the distribution refer the likelihood that the concept is presented. Thus, it can be considered as an advantage of the SOM for using it as a tool for representing the meaning of the sentence against the binary symbolic encoding vectors which can only represent one concept. Specifically, rather than exact formulation of the presence of particular element (symbolic level), on sub-symbolic level is the information distributed throughout artificial neurons and is rather numeric[4]. We obtain this information by presenting training vectors to SOM each one at the time. However, this time we do not include updating of the weight vectors. It is rather extracting activity of the cells of the SOM. This is done by equation:

$$a_i(t) = exp(-c * \ \parallel \vec{w_i}(t) - \vec{x}(t) \parallel^2) \tag{2.1}$$

On the right hand side of equation 2.1 is gaussian term that reflects how likely the current input $x_i(t)$ corresponds to an episode (situation) remembered in the weights $w_i(t)$ of the i-th cell. Parameter c expresses sensitivity of the gaussian term. Finally, we normalize activity of each unit to sum up to 1 by

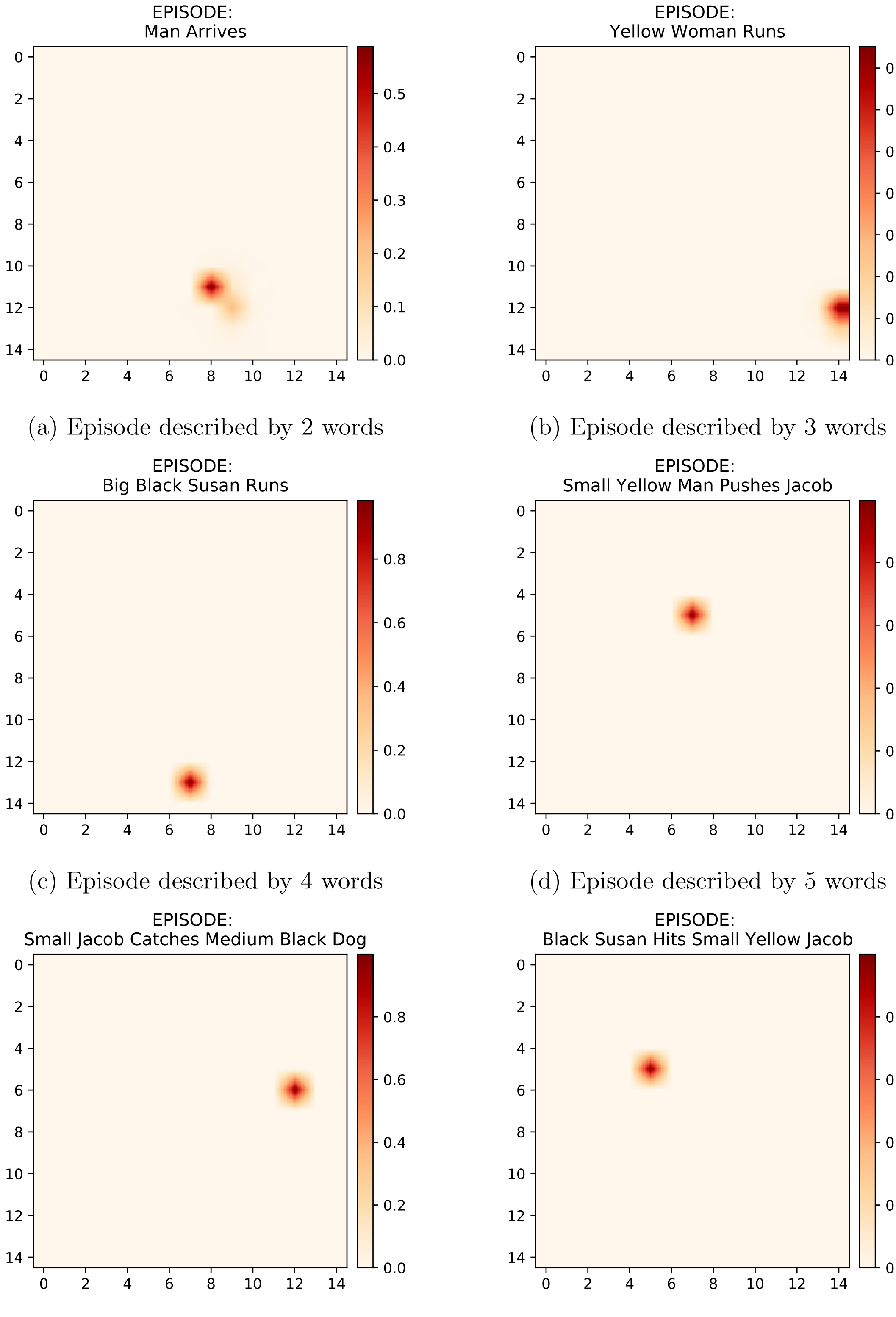
$$A_i(t) = \frac{a_i(t)}{\sum_{j=1}^{N} a_j(t)} \tag{2.2}$$

so that overall activity of the SOM for current input $x_i(t)$ can be interpreted as a probability distribution over possible memorized training episodes. (Takac & Knott, 2015).

[2] Unit of training process. Usually more epochs are needed for successful training.

[3] Or localist since the 1’s ”locate” the element in the vector

[4] On symbolic level it may be understood also as logical when 1 indicates True or that element is present and 0 False.



(a) Episode described by 2 words

(b) Episode described by 3 words

(c) Episode described by 4 words

(d) Episode described by 5 words

(e) Episode described by 6 words

(f) Episode described by 7 words

Figure 2.3: Activation of neurons on the SOM lattice after presenting them particular episodes to which they were trained. Note, that every episode has quite a distinctive representation. It means that the episodes are well distinguishable what is good in the further training process.

2.2 SOM-SRN Model

Our first model of text comprehension uses for sequential processing task the Simple recurrent Network (SRN) (Elman, 1990, 1991). The role of this network is to process the sentence word by word at the time and "to associate" it with the target - the meaning of the sentence. This process is computational simulation of the process of comprehending the text in human, where people create situational representation of the episodes that are described. This model is similar to the one proposed by Frank (2005). However, his training text created using microworld strategy was different to ours [5]. We rely, we base on assumption that during comprehension the participant creates his/her representation of episode activating sensorimotor routines called deictic routines that have canonical structure. Specifically, they consist of semantic element such as *agent, action* and *patient.* Next, we will describe the training of the model into the detail.

The sentence is divided into words which are represented as one-hot [6] vectors where 1 indicates presence of the word from all possible words. These vectors serve as an input to the SRN. The signal from the weights connected input and context layer to the hidden layer is spreading into hidden neurons. Activation function on the hidden neurons was tangent-hyperbolic function. Intermediate signal of hidden neurons is now copied into the context layer. Then, the final prediction of meaning of the sentence (episode) by computing linear combination of weights connecting the hidden units with the output layer and activation of hidden neurons. Target function on the output layer was softmax function that ensures that output vector gives by sum 1. Therefore, we can consider it as the probability distribution of units' activations. Since the target signal, which is the distributed situational representation of the episode achieved by activation of SOM neurons is known, the error between the prediction of the network and desired output is computed and stored. After finishing the sequence of words (sentence), the errors are propagating backwards and the weights are updated accordingly. In the result, the model tries to map the meaning or situational representation of the sentence to the text where the situation is described. For better imagination the model is shown in figure 2.4.

[5] "His" microworld consists only from 15 words while ours includes 32 words. We also work with the explicit attributes of the particular subjects like colors or sizes.

[6] One-hot vector is specific type of binary vector of 0s and 1s. The 1 occurs only once and all other values are 0s.

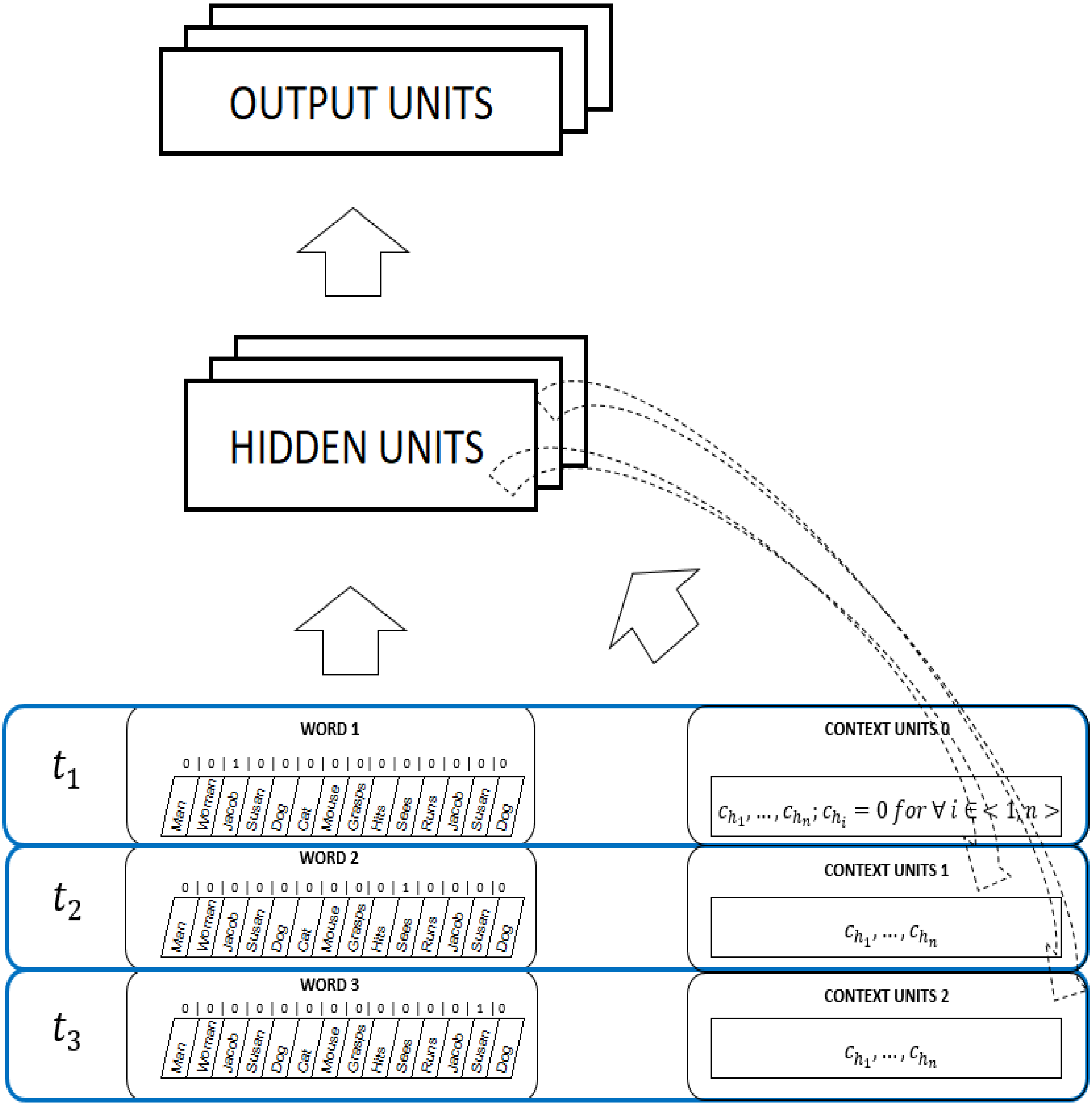

OUTPUT UNITS
HIDDEN UNITS
t_1
WORD 1
0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
Man Woman Jacob Susan Dog Cat Mouse Grasps Hits Sees Runs Jacob Susan Dog
CONTEXT UNITS 0
$c_{h_1}, \ldots, c_{h_n}; c_{h_i} = 0 \; for \; \forall \, i \in < 1, n >$
t_2
WORD 2
0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0
Man Woman Jacob Susan Dog Cat Mouse Grasps Hits Sees Runs Jacob Susan Dog
CONTEXT UNITS 1
$c_{h_1}, \ldots, c_{h_n}$
t_3
WORD 3
0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0
Man Woman Jacob Susan Dog Cat Mouse Grasps Hits Sees Runs Jacob Susan Dog
CONTEXT UNITS 2
$c_{h_1}, \ldots, c_{h_n}$

Figure 2.4: Training process of SOM-SRN model. Each input vector consists of word part, meaning part and context of previously seen elements (that was copied from hidden units from previous time step). Solid arrows indicate full connectivity from one layer to the next. On the other hand, dashed arrows indicate that after processing of each word, the activations of the hidden neurons are copied to the context layer. Note, that first context layer is vector of 0s since there was no previous word. Finally, the target of the output signal is activity of the SOM neurons that represents meaning of the sentence.

2.3 SOM-MSOM Model

The second model that we created uses for sequential processing task the Merge Self-Organizing Map (MSOM) (Strickert & Hammer, 2005). Instead of the first model that uses SRN, MSOM belongs to category of unsupervised learning networks and therefore its training process must be different. However, the principle of self-organization is more biologically plausible than backpropagation algorithm (Newman & Polk, 2008) and our work can be an initial point to further research in language comprehension using such principles.

Unsupervised learning paradigm does not allow comparing results produced by the model with desired output. That is often the case when target or dependent variable is not available. In that case, algorithms operating on unsupervised base usually use available attributes and try to find some similar relationships, patterns or dependencies between them based on distance of their vectors, similar occurrence etc. Nevertheless, in our model the task to which MSOM deals with, takes into consideration desired output values. Specifically, meaning of the episodes encoded in the activities of SOM neurons. That can be achieved in the way that we add desired distributed situational space vectors to all one-hot vectors representing the word in the current episode. It is important to combine only target vectors that correspond to current episode. Thus, the MSOM will learn to associate sequences of words with a sentence meaning (see also figure 2.5):

- **Word** part
- **Meaning** part

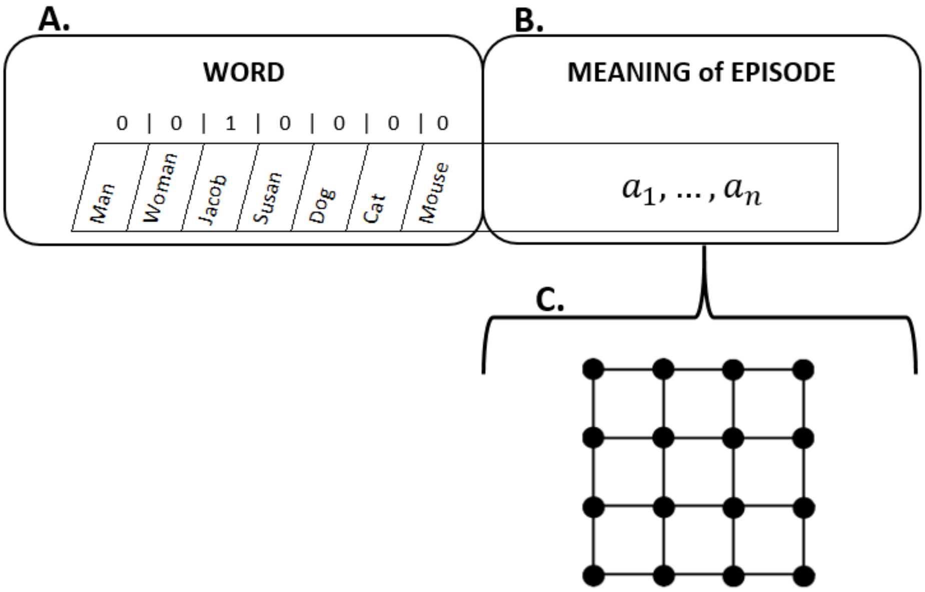


Figure 2.5: Illustration of input vector of SOM-MSOM model. **A** and **B** represents word and meaning part of the vector respectively. **C** illustrates the meaning part of the input vector which consists of the meaning of the whole episode of which the word (**A**) is a part. Example of the episode can be [*JACOB SEES SUSAN*].

As in SOM-SRN model, *word* part of the input vector is represented by one-hot vector where 1 indicates that the element is present. On the other hand, *meaning* part of the input vector is exactly desired output as in the SOM-SRN model. However, while in SOM-SRN model this vector is located ”behind” the output layer where it serves to computing error between actual result of the model and what is desired, in SOM-MSOM model is located within input vector as its part. Thus, such vector has $n + m$ dimensions, where n is dimensionality of the word part[7] and m is dimensionality of the meaning part. Since the meaning part is exactly the activations of SOM neurons, which encodes the meaning of episodes, m is equal to number of neurons of SOM. Training process of MSOM then continues regularly as it was explained in section 1.4.2. It can summarized as follows:

1. Select a sentence that describes particular episode.
2. Setting context descriptor to vector of 0s.
3. Choose input vector representing particular word from the sentence[8]
4. Finding best matching neuron. Here are also considered context descriptor and context weights as they hold the information from previous steps.

[7] How many possible words can occur in all episodes.

[8] In combination with the meaning of whole episode which the sentence is describing.

5. Updating regular and context weights of the MSOM.
6. Updating the context descriptor. Since it holds the properties of the winner neuron in the last step, there is contribution of both regular weights of the MSOM and context weights as well.
7. If the sentence continues, go to the step 3.
8. When the sentence is over, go to the step 1.
9. Repeat until the training ends.

Detailed illustration of this process can be found in figure 2.6. In that example we show the training of the episode [*JACOB SEES SUSAN*]. The one-hot vectors representing the words have the same structure as in the SOM-SRN model. Note, that in figure 2.6 are the word parts of the input vectors abbreviated but it is always vector of length equals to number of possible words in microworld. On the other hand, the meaning part of the input vector is the same for the whole sentence which is describing the episode. The vectors approach to MSOM by one at the time. Specifically, in our example there are 3 steps, one for each word. Context descriptor is updated twice, after time step t_1 and t_2 respectively. When the training approaches word "*SUSAN*", the context descriptor are setting to vector of 0s, since the new episode begins. This process is repeating for every episodes and for T times, where T is maximum number of epochs (training cycles).

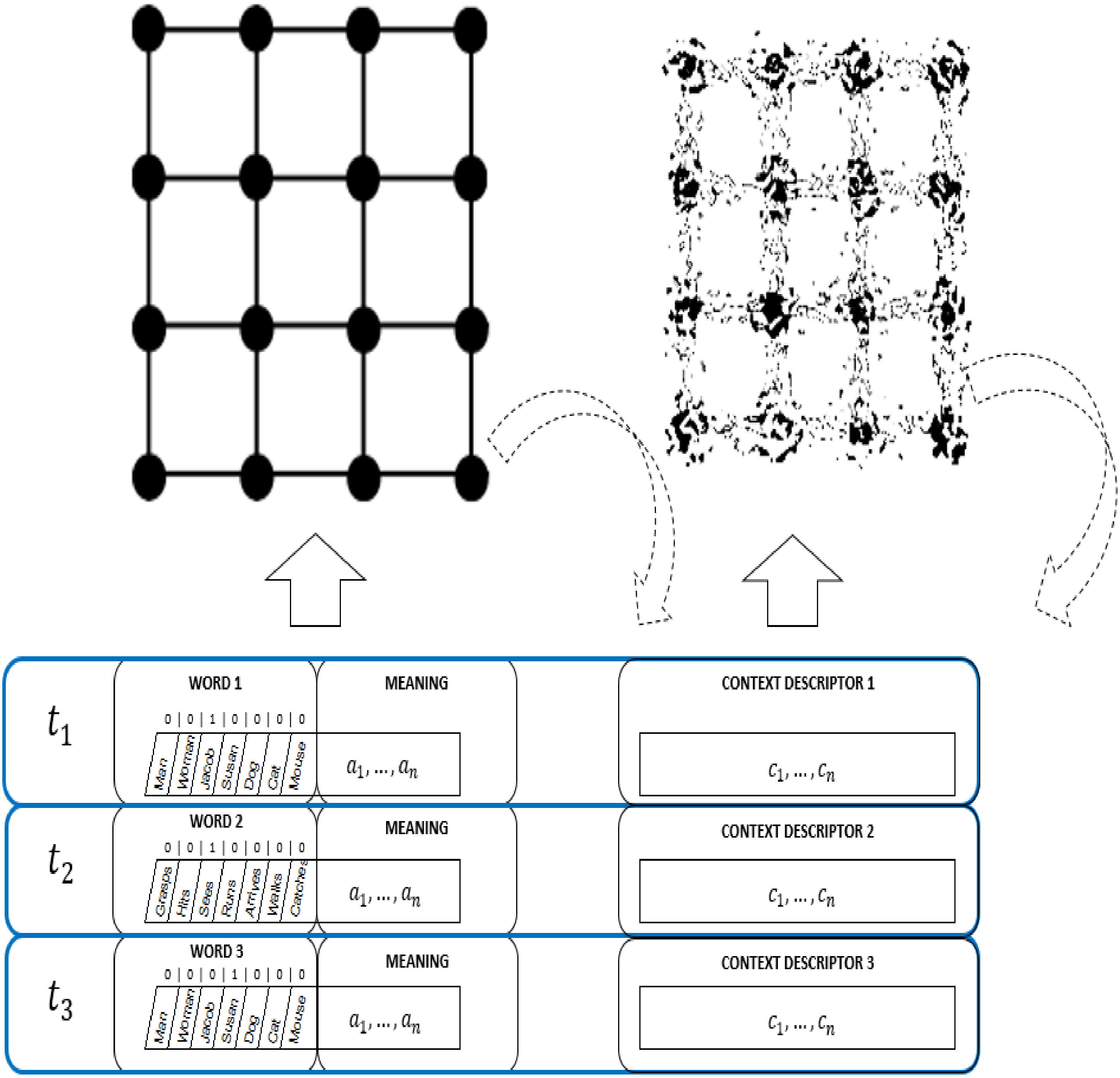


Figure 2.6: Training process of SOM-MSOM model. Each input vector consists of word part, meaning part and context descriptor. First two parts are compared with the regular weights of the neurons on the MSOM while context descriptor finds best matching unit in the context weights. Context descriptor are updated after each word. When an episode comes to an end context descriptor, are set to vector of 0's.

2.4 Reconstruction of the meaning and its prediction

The goal of our models is to bring computational architecture relevant to the task of language comprehension. That means, both models should be successful in prediction of the meaning the sentences (text) that are describing the episodes (abstract situations). Since we are simulating the process of how the human comprehends the language, it is reasonable, that our model should predict the meaning from the uncompleted information, in this case, unfinished sentences. We can take stereotypes as an example. Since early childhood we have heard that cats catch the mice. Thus, when someone tells us: "*Hey, look, a cat ...*", we may a priori assume that there is a cat that catches a mouse. Obviously, this depends also on other factors like how much we have heard about such situation or whether we have a cat in our house etc. However, when we hear or read such unfinished sentence we start to predict the situation which the sentence is describing. Specifically, we are starting to create situational model of the sentence. Naturally, we may make a mistake and create a wrong situational model[9]. For this reason we tried not only to predict the meaning of the episode from the whole sentence that describes the episode, we also tested how the models performed on the prediction of the meaning after presenting them one word at the time. Specifically, how the model predicts the meaning of the sentence after first word is presenting to it, then second word, etc.

The computation of prediction of the meaning after word is presenting to SOM-SRN is straightforward and is computing, using equations 1.10 and 1.11. Nevertheless, in the SOM-MSOM model this is achieved by propagating the activities of MSOM neurons top-down throughout the vectors of their weights:

$$\vec{y} = \sum_{j=1}^{K} A_j * \vec{w_j} \tag{2.3}$$

where A_j is normalized activity of the j-th MSOM neuron and w_j represents its vector of weights. Note, that activity of the MSOM neurons is computed considering the context as:

$$a_j = exp\left(-c\left(\big(1-\beta\big) \; \| \vec{w_j} - \vec{x}(t) \|^2 \; + \beta \; \| \vec{c}(t) - \vec{c_j}(t) \|^2 \right) \right) \tag{2.4}$$

and such activity is normalized to sum up to 1 as in equation 2.2.

As it was said previously such vector of activities can be considered as a probability distribution of possible meanings. Thus, the value of activity of the winner neuron (or,

[9] We can predict that there is only "a cat" but there is "a cat that catches a mouse" or we can incorrectly predict that there is "a black cat that crosses the street", especially if we are superstitious, but there is only "a nice cat".

vice versa, neuron with the highest activity) can be considered as the most probable hypothesis from the hypothesis space represented as the vector of activities. Therefore, such neuron has the highest impact in further reconstruction of the sentence meaning. That is computed by top-down reconstruction from the weights of the SOM (Takac & Knott, 2015):

$$\vec{y} = \sum_{j=1}^{N} A_j * \vec{w_j} \tag{2.5}$$

where A_j is the normalized activity of the jth neuron of the MSOM and $\vec{w_j}$ is the vector of vector of weights of the SOM corresponds to particular place of the element in situational space vector. N is number of neurons in SOM. Simply said, we take all the weights corresponding to particular element in DSS (e.g. Man, Human etc.) multiply them by activity of the jth neuron of the MSOM and then sum up these weights. In the result we gain the probability of the presence of particular semantic element. It is important to note, that in reconstruction of the meaning of the sentence we take into consideration only the *word* part of the input vectors (see figure 2.5 for more details).

2.5 Technical summary of the program and training data

In this section we will summarize the technical details of our models and training data. We programed the models in Python language (3.7) in Anaconda environment. We used our implementation of Self-Organizing Map and Merge Self-Organizing Map both programmed by ourselves. As implementation of Simple Recurrent Network we use the program created by Denny Britz which we modified to our purposes. The program can be found on his website (Britz, 2018).

For generation of training data we used the sentence generator created by Martin Takac. The program was made in JAVA. Using the words summarized in table 2.1 we have generated 10 000 sentences constrained to the rules showed in section 2.1.1 on page 22. From these sentences we randomly chose sentences with particular lengths from 2 up to 7 words. From each length of the sentence was chosen 50 examples. Thus, our training data contains 300 sentences where were 50 2-word sentences, 50 3-word sentences etc.

Chapter 3

Results

In this chapter we will present the evaluation of our models of sentence comprehension. We will compare these models against each other, since our aim is to introduce biologically plausible model of sentence comprehension based on self-organization. Firstly, we will shortly describe the methods of evaluation of models' performance. We will next show the progress during training process. Finally, we will compare the two models in how well they can predict the meaning of the whole episode after presenting them one word at the time.

3.1 Performance measure

Each training process does not take place without the evaluation of how well the model performs in achieving the goal to which it was created. To evaluate ongoing training process of Simple Recurrent Network we used cross-entropy loss function which is defined as following:

$$H\ (\vec{p}, \vec{q}) = -\sum_{i} p_i\ log(q_i) \tag{3.1}$$

where p is probability distribution of prediction of the vector of meaning[1] and q is the meaning vector of the episode. Recall, that softmax activation function on the output layer in SRN secures that the prediction vector gives in sum 1 and can be considered as the probability distribution. In the meaning vector this is secured by normalizing the activities of the SOM neurons.

Vančo & Farkaš (2010) described several types of measurements of the recursive self-organizing maps' performance. We used in our work:

- Quantization error

[1] We just recall that this vector is obtained from activity of neurons of Self-Organizing Map. But we have to distinct the probability distribution p which is the output predictions of the SRN.

- Winner differentiation
- Average amount of adjustment of the neurons

Quantization error is the measure of how well the input weight vectors retain statistical information about the labels distribution. Simply said, it takes into consideration the average distances between the weight vectors and their best matching units. This measure is defined formally as:

$$E = \frac{1}{N} \sum_{i} min_j \parallel \vec{x_i} - \vec{I_j} \parallel^2 \tag{3.2}$$

where $\vec{x_i}$ is the input weight vector and $\vec{I_j}$ is the weight vector of its best matching unit and N is number of training examples. Note, that this equation is valid for regular SOM. For the MSOM it has to be considered within the context, thus the formulation is :

$$E = \frac{1}{N} \sum_{i} min_j \left(\parallel \vec{x_i} - \vec{I_j} \parallel^2 + \parallel \vec{c}(t) - \vec{c_j} \parallel^2 \right) \tag{3.3}$$

where $c(\vec{t})$ is the context descriptor and $\vec{c_j}$ is the weight context vector of jth neuron.

Winner differentiation is simply the ratio of the number of all different winners from the entire dataset and the size of the dataset (or number of examples that the network has seen).

The last measure is rather informative and tells us how much the neurons adjust throughout the training process. The adjustment of neurons should be large at the beginning of the training and it should dramatically decreased during the learning. In the end of the training the adjustment of neurons is minimal. It is expressed as:

$$A(t) = \frac{1}{N} \sum_{j} \triangle \vec{w_j}(t) \tag{3.4}$$

where

$$\triangle \vec{w_j}(t) = \parallel \vec{w_j}(t) - \vec{w_j}(t-1) \parallel^2$$

is the distance of displacement of neuron w_j from time step $t-1$ to time t.

Previous types of measurements of performance are focused on testing if the training process was successful. However, our goal is to design a model which provides sufficiently good mapping of meaning of episodes to the sentences that these episodes describe. In other words, that could be considered as a proof that the model can extract the semantic information from the text whenever it is presented to the model. Recall, that the meaning of the sentences are encoded in the activities of the SOM neurons while the recurrent neural networks (SRN and MSOM) try to map these meaning to the sentences which describe their meaning. For testing whether the model correctly predict the meaning of the sentence we use Kullback-Leibler (KL) divergence which is the measure of how the one probability distribution differs from another - expected probability distribution. Or in other words, it is the measure of how much information we lost when we tried to approximate one distribution with another. KL is formally defined as:

$$KL(\vec{p}, \vec{q}) \; = \; \sum_i \; p_i \; log \; \frac{p_i}{q_i} \tag{3.5}$$

where $\vec{(p)}$ is observed or predicted probability distribution while $\vec{(q)}$ is expected one.

Similar to human who creates situational representation of an episodes which he/she perceives, computational model should be also able abstract such a representation of an episode. This should be done even at the beginning of the situation, where we do not know all of the information about the episode (e.g. we are reading second word from the eight word sentence). Specifically, we are able to predict the meaning of the sentence according to only few words which have seen so far or at least, to predict to some amount. Such ability our models should have as well. To test it, we presented to the networks the sentences word by word and after each word we measure the KL divergence between predicted meaning of the sentence and actual meaning. As it was mentioned in previous chapter, in SRN is this task quite straightforward. We only need to spread the signal from the input layer in the forward direction to the output layer. On the other hand, in MSOM we need to reconstruct the meaning by spreading the activity top-down to the weights of the MSOM. Nevertheless, it is important that we only use the part of the input vector that corresponds to the word. For better illustration, we recommend to see the figure 2.5 where the structure is shown of an input vector to the MSOM. If the meaning of the sentence after presenting a particular word were reconstructed from the whole input vector, the resulting prediction would be contaminated by the desired output and such result would be incorrect[2]. Finally, this reconstruction is done using equations 2.3 and 2.4. Note, that in this step we do not reconstruct particular elements of the meaning (e.g. Jacob, Human, Transitive Verb etc), thus we do not spread the signal top-down to the weights of the SOM (see section 2.4 for more details).

3.2 Summary of the models

In this section we will summarize the process of finding the most appropriate model. There is a lot of trial-error "playing" with the models' parameters in computational

[2] And perhaps incorrectly more accurate

modeling and researchers often try many combinations and variations of the parameters' settings. In our work we tried various combinations of parameters and from all possible combination we have chosen the best representative model. In tables 3.1, 3.2 and 3.3 respectively are the parameters that the models showed the best results with. We chose 4 models with the best performance with current parameter settings. Our methodology was as following:

1. *Find the most appropriate parameter settings of Self-Organizing Maps that would encode the meaning of the sentences in the best way.* Here we have mostly manipulated with the number of epochs, the size of the adaptive phase, learning rate α, parameter of neighborhood and gaussian sensitivity in computing activation of the SOM neurons.

2. *Find the best parameter settings for Simple Recurrent Network and Merge Self-Organizing Map.* These networks are used for serial processing task, specifically, mapping the meaning to the sentence which is processed word by word.

3. *The measure of performance of the model is an extent to which the model is able to predict the meaning of the sentence from the presenting the words of that sentence.*

Next we are presenting parameter settings of most successful models. Each model was tested for various combination of parameters. For example, by Model 1 of Self-Organising Map we tried to manipulate with initial and final values of learning rate. The values of parameters of Model 1 [3] in table 3.1 then represent the best settings of learning rate while holding other parameters constant. Note, that it would be almost impossible to try every possible combination and variation of parameters. Therefore we manipulate based on theoretical background and our intuition (see (Ritter & Kohonen, 1989; Kohonen, 1998; Haykin et al., 2009; Van Hulle, 2012) for more about SOM parameter settings). Even though the parameters of the models were introduced in section 1.4, for quick reference we again briefly explain here these parameters. However, we recommend to see section 1.4 for detailed information[4].

- $\boldsymbol{\alpha_i}$ - learning rate, where i indicates initial value[5].
- $\boldsymbol{\alpha_f}$ - learning rate, where f indicates final value[6].
- $\boldsymbol{\lambda_i}$ - parameter that helps to control the neighborhood function. i represents the value from start of the training.

[3] M_1 indicates "Model 1"

[4] We mention a reference of a section by every parameter

[5] The value of α at the beginning of the training

[6] The value to which the α decreases during the training

- $\boldsymbol{\lambda_f}$ - final value of parameter λ. All previous parameters are described in section 1.4.1 on page 14.
- $\boldsymbol{c}$ - gaussian sensitivity participating in computing the activity of neurons in SOM and MSOM. More in section 2.4 on page 34.
- $\boldsymbol{\gamma}$ - controls a contribution of the regular and context weights during computing context descriptor (equation 1.8 on page 15)
- $\boldsymbol{\beta}$ - controls a balance between regular and context weights while best matching unit finding process (equation 1.7 on page 15)

Table 3.1: Parameters of the Self-Organizing Maps

ID	**N epochs**	**Ep conv**	**Width**	**Height**	$\boldsymbol{\alpha_i}$	$\boldsymbol{\alpha_f}$	$\boldsymbol{\lambda_i}$	$\boldsymbol{\lambda_f}$	$\boldsymbol{c}$
M_1	1000	750	15	15	1	0.15	8	1	1.5
M_2	1000	850	20	20	1	0.5	8	1	3
M_3	1000	920	15	15	1	0.5	12	1	4
M_4	1000	950	15	15	1	0.5	9	1	3

Table 3.1 shows the best parameter settings for 4 models of SOM. Column *ID* indicates ID of the model, *N epochs* the number of epochs, *Ep conv* is the epoch when the training goes to adaptive phase and *width* and *height* define size of the SOM lattice. Note, that all the models were compared to the other ones where specific combination of parameters was manipulated while the other settings remain constant. The final models were chosen according to the best performance showed in table 3.4.

Next table (3.2) shows the best parameter settings for 4 MSOM models. The meaning of the parameters are equal as in previous table, however, there are two additional ones, γ and β which have been explained already as well.

Table 3.2: Parameters of the Merge Self-Organizing Maps

ID	**N epochs**	**Ep conv**	**Width**	**Height**	$\boldsymbol{\alpha_i}$	$\boldsymbol{\alpha_f}$	$\boldsymbol{\lambda_i}$	$\boldsymbol{\lambda_f}$	$\boldsymbol{\gamma}$	$\boldsymbol{\beta}$	$\boldsymbol{c}$
M_1	50	45	20	20	1	0.5	10	1	0.6	0.7	5
M_2	500	425	15	15	1	0.5	7	1	0.35	0.35	1.5
M_3	200	180	15	15	1	0.5	15	1	0.6	0.2	5
M_4	100	90	15	15	1	0.5	15	1	0.6	0.3	5

Unlike previous networks which work based on self-organization, in simple recurrent network fewer parameters are needed to be defined. Specifically, *N input* is dimension of the input vector[7], *N hidden* is a number of neurons on hidden layer, *N epochs* is number of epochs in training and α is a learning rate.

Table 3.3: Parameters of the Simple Recurrent Networks

ID	**N input**	**N hidden**	**N epochs**	$\boldsymbol{\alpha}$
M_1	32	150	1000	0.08
M_2	32	200	1500	0.08
M_3	32	120	1500	0.05
M_4	32	100	1500	0.05

Table 3.4: Summary statistics of models' training

ID	**SOM E min**	**SOM E avg**	**SOM A min**	**SOM A avg**	**SRN E min**
M_1	0.592	2.837	0.000	0.914	1.838
M_2	0.031	2.423	0.000	0.917	2.554
M_3	0.209	3.509	0.001	1.659	0.136
M_4	0.608	3.244	0.002	1.146	1.972

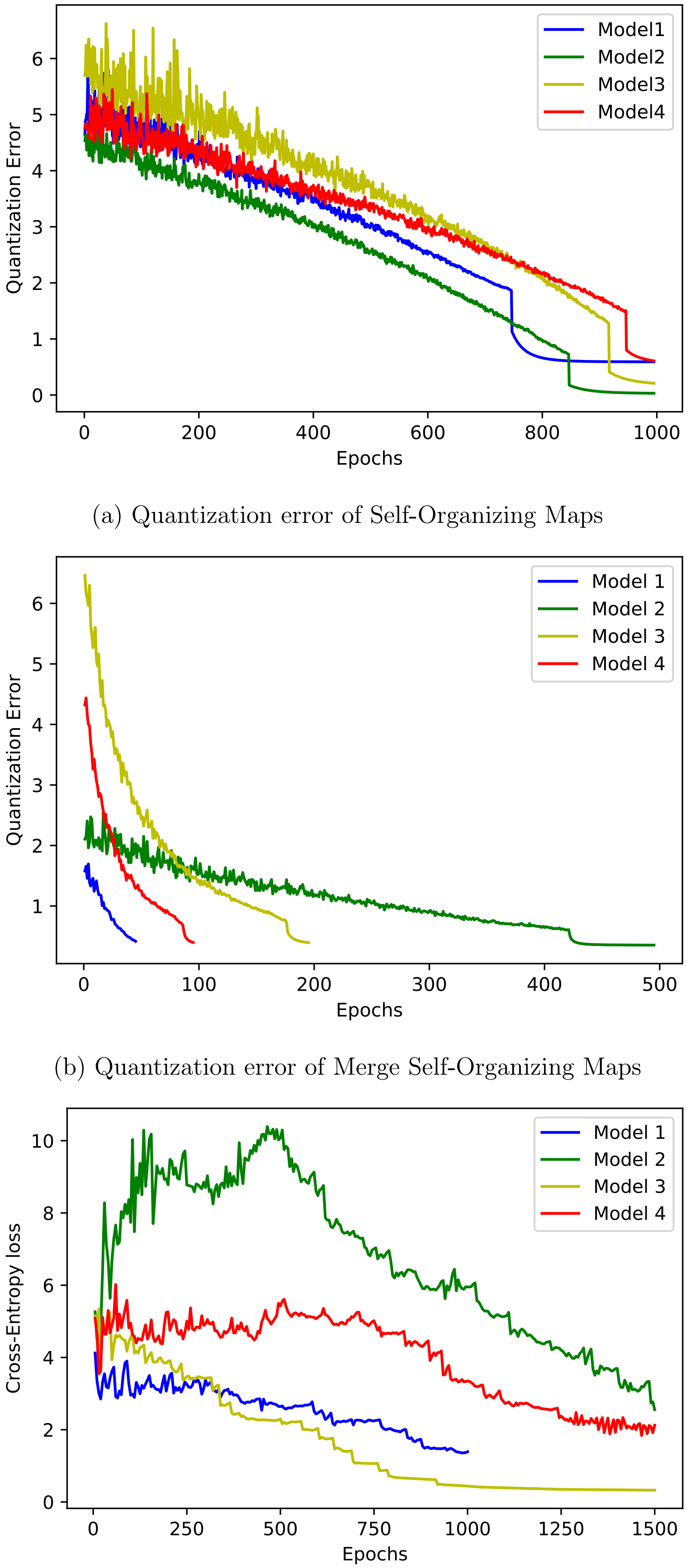
	SRN E avg	**MSOM E min**	**MSOM E avg**	**MSOM A min**	**MSOM A avg**
M_1	3.983	0.417	1.940	0.365	1.092
M_2	6.733	0.354	1.205	0.001	0.945
M_3	1.570	0.409	2.274	0.007	1.480
M_4	2.568	0.398	2.083	0.012	1.476

Table 3.4 shows basic statistics of models' performance. By "*model*" we mention here the model of particular neural network (SOM, MSOM or SRN) not a complex model of sentence comprehension per se. However, the number of the model of MSOM and SRN also represents a reference to the model of SOM. Specifically, the SRN $\boldsymbol{M_1}$ model was trained to the target meaning obtained from activities of the SOM $\boldsymbol{M_1}$ model neurons. This also applies to the MSOM $\boldsymbol{M_1}$ model.

In section 3.1 we have introduced several types of performance measure of the models. Table 3.4 summarizes how the models performed in the training. In SOM and MSOM models **E** refers to quantization error, in SRN model to cross-entropy loss. **A** indicates adjustment of the neurons of SOM and MSOM. **min** indicates minimum and

[7] Since we used only the microworld with 32 possible words, the value of *N input* can be considered as constant across all models.

avg refers to average particular values across the whole training (the size of the training can be found in tables 3.1 - SOM models, 3.2 - MSOM models, and 3.3 - SRN models). Nevertheless, these measures should not be considered as an overall measures of model quality. Rather it is an indicator that the models have been trained sufficiently.



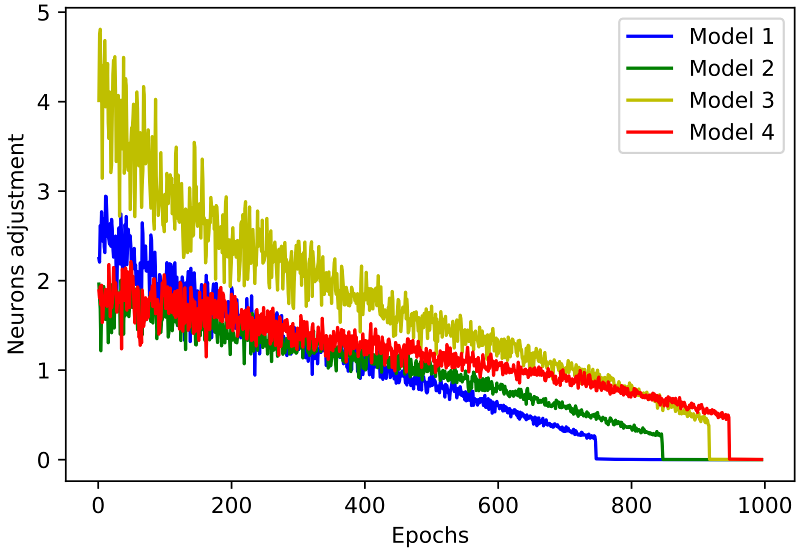
(a) Quantization error of Self-Organizing Maps

(b) Quantization error of Merge Self-Organizing Maps

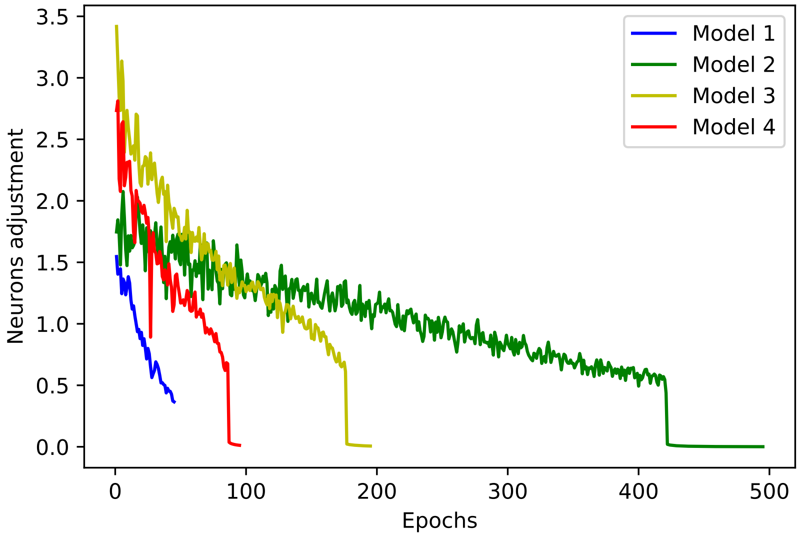
(c) Cross-entropy loss of Simple Recurrent networks

Figure 3.1: Training process of neural networks. Quantization error is shown for SOM and MSOM and Cross-entropy loss for SRN.

In SOM and MSOM networks can be also shown an adjustment of neurons during the training. Although this information is quite straightforward and we may assume that this adjustment would continuously decrease during the training, a plot showing this process can be used for purpose of training's diagnostic. This information about SOM and MSOM training is shown in figure 3.2. Apparently, there is no suspicious event showing in the plots. A rapid decrease of the adjustment of the neurons is caused by starting an adaptive phase of the training.



(a) Adjustment of neurons of the SOMs averaged across epochs



(b) Adjustment of neurons of the MSOMs averaged across epochs

Figure 3.2: Adjustment of neurons of SOM and MSOM

The parameters' settings

At the beginning of the section 3.2 we have summarized our steps in the process of choosing the best model. Firstly, we should find the best parameter settings for particular neural network. From the results shown in table 3.4 and figure 3.1a we can conclude that the best parameter settings of Self-Organizing Map for encoding the meaning of the sentences are those in model M_2 because the model achieved the lowest average quantization error. Similar criteria can be used in choosing the best parameters of SRN and MSOM models. Thus, the best setting of parameters of Simple Recurrent Network is that used in model M_3 and for the Merge Self-Organising Map the ones used in model M_2. Note, that these criteria are rather arbitrary and researcher can use other conditions. From the figure 3.1 we can also conclude that the training of models using self-organization (SOM and MSOM) was smoother[8] than of the SRN. There is also visible flagrant decrease of error in starting an adaptive phase of training of SOM and MSOM (figures 3.1a and 3.1b).

We use these information in the setting with the best parameters for our models. The best training performance from the SOM models shown the model M_2. However, since this model used bigger SOM lattice (20x20 neurons) and the difference between average error was not large, we decided to use parameters of SOM model M_4, also taking into consideration the Occam's razor. Furthermore, since the SOM and MSOM networks have to have the same size of lattice we considered the results of MSOM models in making appropriate combination. Secondly, for SRN we chose the parameters corresponding to SRN model M_3. Finally, we trained MSOM with the parameters used in MSOM model M_2.

Predicting the meaning of the sentences

As a measure of how well the model predicts the meaning of the sentence we have chosen Kullback Leibler divergence (more in section 3.1). An accurate computational model of sentence comprehension should not only gain sufficiently small discrepancy between comprehended meaning of the sentence but as well its actual meaning when the model knows the whole context[9]. It should achieve a good prediction power to comprehend the meaning of the sentences while only few words from the sentence have been presented to it. Therefore we have tested the prediction power of the models each time after a particular word from the sentence was introduced to the model. We summarize its computation into following steps:

1. Combine sentences with the similar length into groups. In our case we get 6

[8] The error decreased continuously in a direct way while the SRNs showed more saccades in fitting the model.

[9] In this case we mean that the model has already seen the whole sentence.

groups of text sentences while each group contains 50 sentences.

2. Take each group and compute the prediction of the meaning from each sentence after each word was presented.

3. Compare these predictions with actual meaning of the sentences by Kullback-Leibler divergence.

4. For every group of sentence compute the average KL divergence after each word. Specifically, take a group with sentences composed of 4 words. Average the predictions of the meaning after 1^{st} word then after 2^{nd} word etc. until we get the prediction from the whole sentence (4 words).

5. Repeat this process for all groups of sentences.

Tables 3.5 and 3.6 as well as figure 3.3 shows the results of the models' performance in prediction of the meaning of sentences. We can see that SOM-MSOM model did better at the beginning of the sentences. It is obvious that both models achieved much higher prediction accuracy after more words from the sentence were presented. Specifically, when the model "sees" the first word, its prediction of the meaning is highly dependent on the word that is presented and how often it occurred at the beginning of any sentence. For example, we can take a sentence "*Small Jacob sees Susan.*". When we consider only the first word of this sentence *small*, the sentence can continue in a very large number of ways. After we are reading further, we continuously discover its actual meaning, or the situation that this sentence describes. Elman (1990) explains this process in context of connectionist modeling. Thus, when the model sees "*small*" as a first word of the episode it predicts some combination of all possible episodes where the *small* stood at the first place in the training text data[10]. Therefore, after a first words of the sentence is the prediction error highest and it is decreasing after more words is presented to the network. Such case is clearly seen in the figure 3.3. Thus, we can conclude that we successfully simulated "online" process of sentence comprehension. However, in this case is only carrying on the mapping the meaning of the sentence to the words that it contains and its accurate prediction.

[10] In reality, there is infinity number of possible sentences started by word *small*. It is related to generativity of our language. However, since we constrained the language into the microlanguage or microworld, here the number of possibilities is discrete number but is also high.

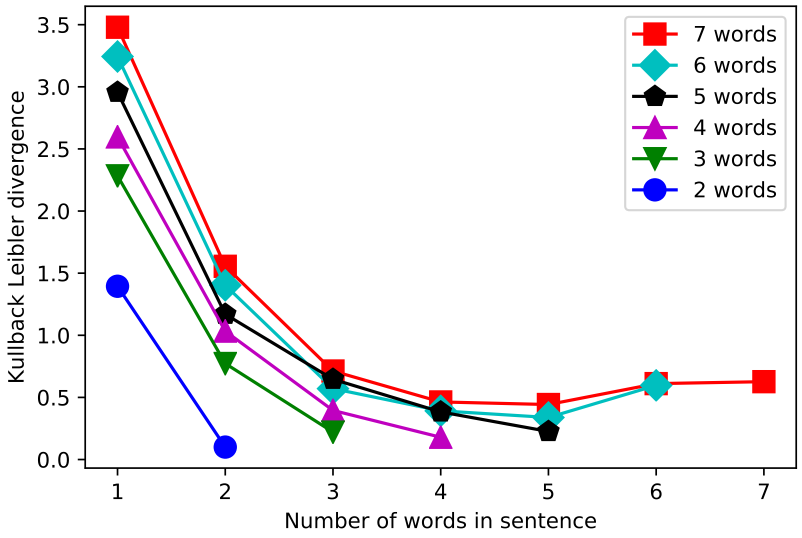
Table 3.5: KL DIVERGENCE AFTER WORD IS PRESENTING (SOM-SRN MODEL)

1st word	**2nd word**	**3rd word**	**4th word**	**5th word**	**6th word**	**7th word**
1.3945	0.1001					
2.2836	0.7728	0.2209				
2.5964	1.0333	0.3952	0.1791			
2.9562	1.1694	0.6461	0.3832	0.2258		
3.2440	1.4024	0.5707	0.3919	0.3383	0.5968	
3.4795	1.5553	0.7124	0.4634	0.4426	0.6109	0.6266

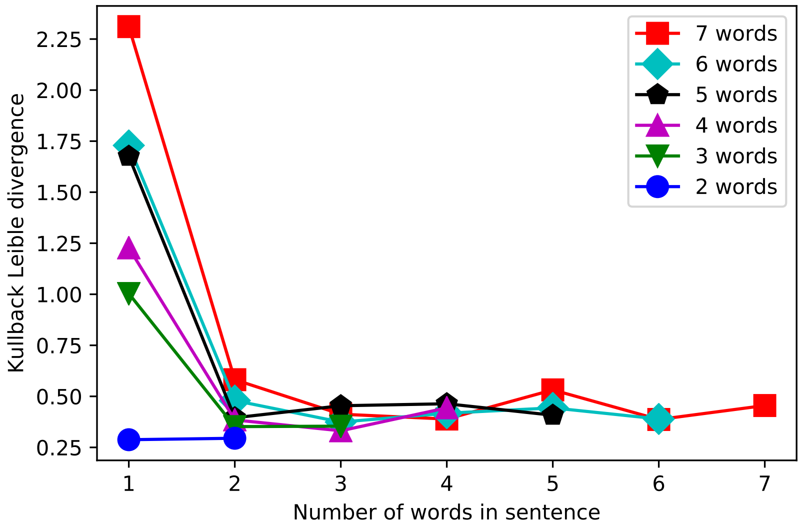
Table 3.6: KL DIVERGENCE AFTER WORD IS PRESENTING (SOM-MSOM MODEL)

1st word	**2nd word**	**3rd word**	**4th word**	**5th word**	**6th word**	**7th word**
0.2876	0.2948					
1.0036	0.3517	0.3540				
1.2271	0.3846	0.3313	0.4436			
1.6759	0.3952	0.4540	0.4637	0.4076		
1.7289	0.4794	0.3740	0.4175	0.4435	0.3894	
2.3110	0.5816	0.4127	0.3888	0.5317	0.3867	0.4550

Figure 3.3 shows that our models successfully mapped the meaning of the sentences to the text which they describe. It is obvious that the error between actual and predicted meaning is highest at the beginning of the sentences. This discrepancy decreases after more words are presented. Such an aberration can be considered as an uncertainty of the actual meaning. Thus, we can say that the meaning is changing while we know more about the context of a particular episode. These changes can be captured by the SOM since it encodes the meaning of the sentences in a distributed manner. In summary, we can conclude that both models showed good prediction accuracy of the meaning of the sentences. The SOM-MSOM model performed better in this task since it reduced the error in its prediction after second word and remained good performance.



(a) SOM - SRN model



(b) SOM - MSOM model

Figure 3.3: Kullback-Leibler divergence between predicted meaning of the sentences by SRN and MSOM and the actual meaning represented in activity of SOM neurons. KL was computed after presenting one word at the time. The particular KL values are averaged within the sentences with specific length, distinguished by the color and shape of the points. E.g. red line with square points averaged KL divergence across all sentences containing 7 words. Particular points represent KL values after presenting the number of words according to x axis. The legend refers to the length of the sentence.

3.2.1 Reconstruction of the meaning of the sentences

In previous section we have showed that our models performed sufficiently good in prediction of the meaning of the sentences. However, the meaning of the sentences is just encoded information in the activity of the SOM neurons. We would want to extract this information to find out what the numbers inside distributed situational vectors mean. To do so we spread the activation of neurons of MSOM and SRN down to the weights of the SOM. Similar method used Frank (2005) combining SOM with SRN for the serial processing of the words of sentences. Unlike his method, we based on an assumption that episodes are perceived as sensorimotor-routines with specific structure. On the other hand, combination of SOM and MSOM used Takac et al. (2012). However, in their model they used the top-down reconstruction in modelling of language production. The methodology of reconstruction of the meaning of sentences is explained into detail in section 2.4.

We show in figures 3.4 and 3.5 the example of reconstruction of the meaning of the particular episodes. We use example episodes described by 6 words for SOM-SRN model and by 5 words for SOM-MSOM model. Note, that the more words the sentence has the more certain the model is after presenting the last words. That is caused by the known context of the episode which is explained more detailed by more words. We can also see that there are predicted probabilities in unseen parts of the meaning vectors (e.g. at the beginning in the patient part of the vector). This probabilities refer to prior probability composed of what the network has seen during the training process. For example, in the first subplot of figure 3.5 can be seen, the activity in the place refers to word *white* instead of yellow. That means, that the network has seen the combination of words *small* - *white* in agent part of the vector more often than *small* - *yellow*. However, presentation the next word to the network updates the prediction referring to the right meaning, specifically *small* - *yellow*.

Figure 3.4 shows the reconstruction of the meaning of the episode "*Small Jacob Catches medium black dog*". The particular subplots show the probabilities of the particular semantic elements. In the first subplot is presented some uncertainty in prediction of these elements. The model have seen a little context and the prediction is not quite accurate. After next words are presented to the model the estimate becomes more accurate and the meaning of the episode is more clear. We can see that after the first element from the patient part "*medium*" is shown to the model the prediction becomes sufficiently precise.

Next figure (3.5) shows how the SOM-MSOM model reconstruct the meaning of the episode "*Small yellow man catches cat*". The behavior is similar as in SOM-SRN reconstruction. The estimate becomes more accurate after more words are presented to the model. In summary, we can conclude that both models fairly reconstruct the meaning of the episodes. We note that the sentences were chosen randomly and we shows these reconstructions as an example.

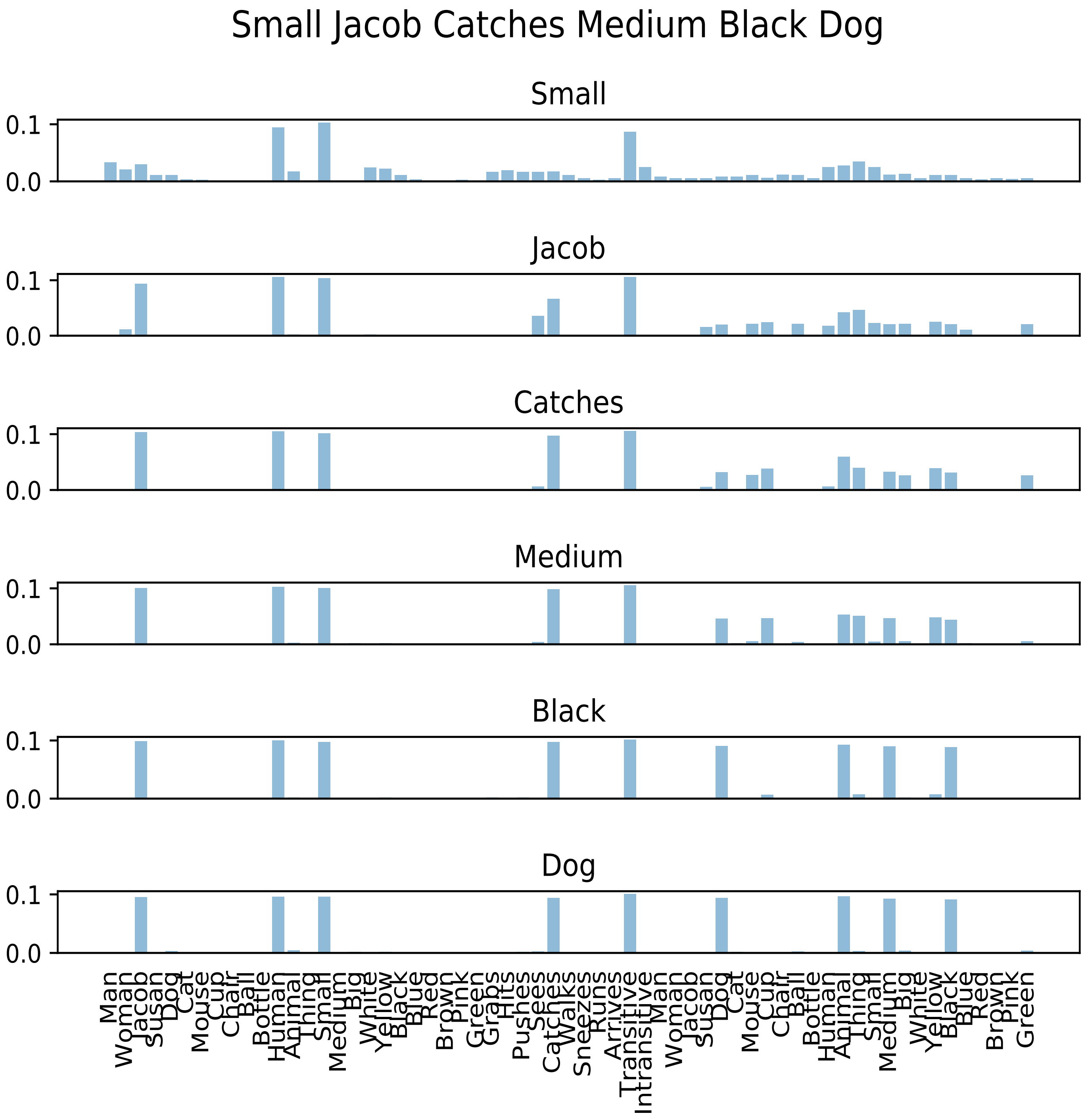


Figure 3.4: Example of reconstruction of the meaning - SOM-SRN model. The example episode is expressed by the sentence *Small Jacob catches medium black dog*. Particular words are presenting to the model and the reconstruction of the meaning of the whole sentence is computing. The methodology is explained in section 3.1.

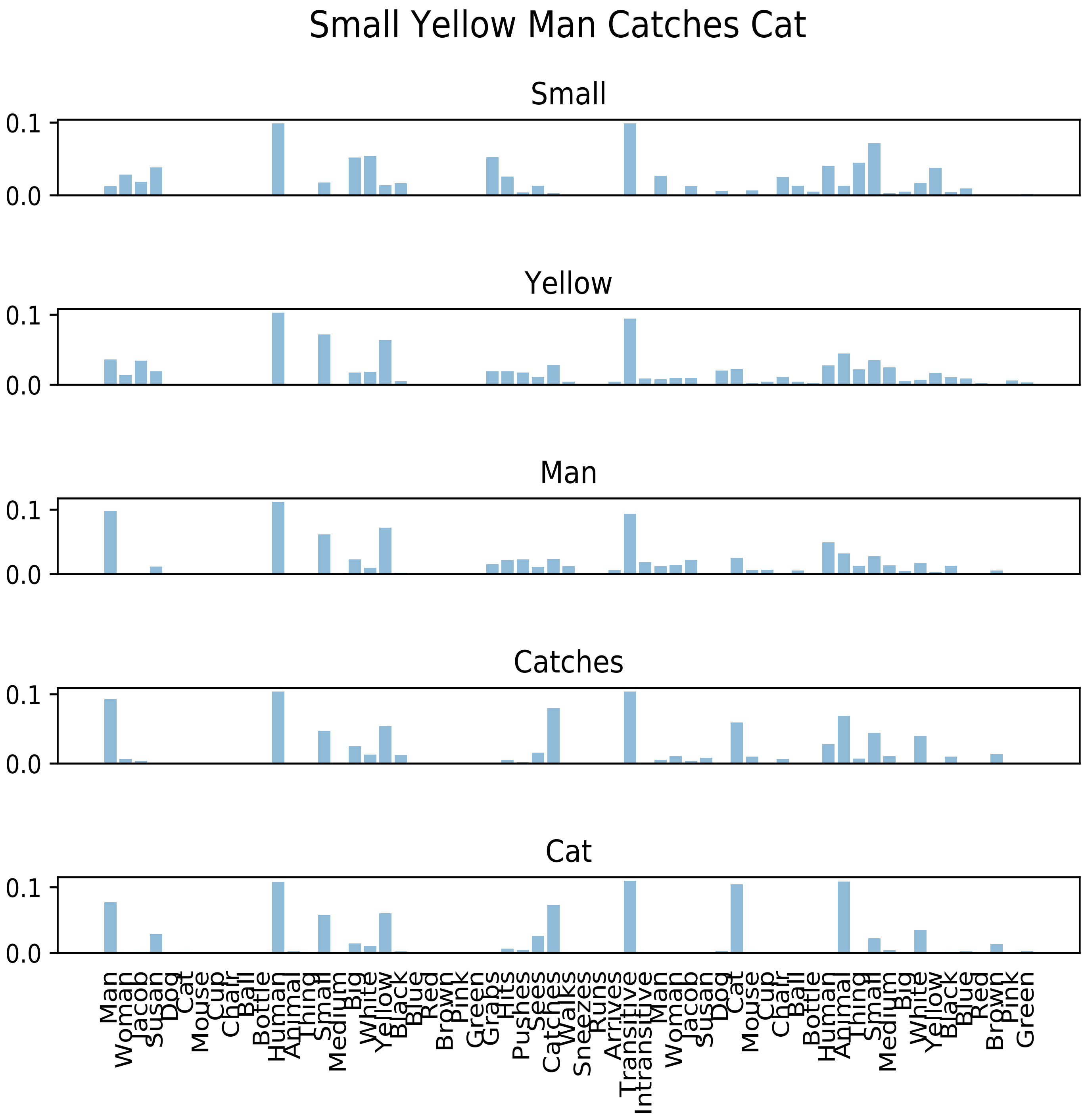


Figure 3.5: Example of reconstruction of the meaning - SOM-MSOM model. The example episode is expressed by the sentence *Small yellow man catches cat*. Particular words are presenting to the model and the reconstruction of the meaning of the whole sentence is computing. Again, the methodology is explained in section 3.1.

Conclusion

In this work we compare two connectionist models of sentence comprehension. Both models using Self-Organizing Maps for representation of the meaning of the sentences. They differed in a sentence processing component. First one used Simple Recurrent Network design by Elman (1990). This model was similar to the one built by Frank (2005). The second one contains Merge Self-Organizing Map introduced by Strickert & Hammer (2005). An inspiration for such choice was the work of Takac et al. (2012) where the authors trained a model for sentence production task. However, to our knowledge such a model has been not used for language comprehension so far. Therefore our work can be considered as an initial research in this area using this architecture.

First chapter was devoted to describing theoretical background of language comprehension, computational models that tried to simulate this process, and artificial neural networks which we used in our models. Second chapter goes into the detail of our models' architecture. Finally, in the third chapter we have shown the results of performance of our models in language comprehension task.

Our goal in this work was to create the model of sentence comprehension in biological plausible manner. Specifically, the models which use symbolic or localist system for representing the semantic concepts do not perform sufficiently good. Thus, comprehension has largely been the domain of distributed, connectionist models (Rohde & Plaut, 2003). However, even connectionist models sometimes lack of enough of biological plausibility. They are often include Simple Recurrent Network (e.g. Frank (2005); Rohde (2002)) which is trained by backpropagation algorithm. Therefore using other network as sentence processing component seems as a legit way in finding suitable biological appropriateness. Using network that uses self-organization as a training method may be a good direction in this area.

Self-organization is a process which can be seen in various brain locations (Singer, 1986; Kohonen, 1998; Gerstner & Kistler, 2002; Newman & Polk, 2008; Van Hulle, 2012; Kahn, 2013). There is evidence that this process is involved in processes that can proceed hand-to-hand with language processing (Chersi et al., 2014). Working memory also plays an important role in language comprehension, where the active reverberation circuits as well as a connection with long term memory storage helps to adequate functioning of processes needed in language processing (Gathercole &

Baddeley, 2014). These findings support the idea of using computational model of language comprehension based on self-organizing systems.

The results of our experiments suggest that the model architecture composed of systems on a basis of self-organization can be represented as an opponent against the traditional connectionist models using SRN. In our meaning prediction task the SOM-MSOM model performed even better than the SOM-SRN model. In addition, the SOM-MSOM model reconstructs the particular elements from the vector of meaning appropriately. Therefore, we can conclude that this architecture can be used in further research using connectionist modeling of language comprehension.

As we have mentioned previously, the self-organization based model can be considered as biologically plausible simulation of process of language comprehension. Another research is needed in examination of the role of other participating processes like working memory, long term memory or executive functions. Our research can therefore bring the a light in explanation of the language phenomena.

Bibliography

Baggett, P. (1979). Structurally equivalent stories in movie and text and the effect of the medium on recall. *Journal of Verbal Learning and Verbal Behavior*, *18*(3), 333–356.

Ballard, D. H., Hayhoe, M. M., Pook, P. K., & Rao, R. P. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, *20*(4), 723–742.

Britz, D. (2018). *WILDML artificial intelligence, deep learning, and nlp.* `http://www.wildml.com/`. ([Online; accessed 10-January-2018])

Carreiras, M. (1996). The use of stereotypical gender information in constructing a mental model: Evidence from english and spanish. *The Quarterly Journal of Experimental Psychology: Section A*, *49*(3), 639–663.

Chersi, F., Ferrari, P. F., & Fogassi, L. (2011). Neuronal chains for actions in the parietal lobe: a computational model. *PloS one*, *6*(11), e27652.

Chersi, F., Ferro, M., Pezzulo, G., & Pirrelli, V. (2014). Topological self-organization and prediction learning support both action and lexical chains in the brain. *Topics in Cognitive Science*, *6*(3), 476–491.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, *7*(2-3), 195–225.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological review*, *102*(2), 211.

Frank, S. L. (2004). Computational modeling of discourse comprehension.

Frank, S. L. (2005). Sentence comprehension as the construction of a situational representation: A connectionist model. In *Proceedings of AMKLC* (Vol. 5, pp. 27–33).

Frank, S. L., & Haselager, W. F. (2006). Robust semantic systematicity and distributed representations in a connectionist model of sentence comprehension.

Frank, S. L., Koppen, M., Noordman, L. G., & Vonk, W. (2008). World knowledge in computational models of discourse comprehension. *Discourse Processes*, *45*(6), 429–463.

Gathercole, S. E., & Baddeley, A. D. (2014). *Working memory and language.* Psychology Press.

Gerstner, W., & Kistler, W. (2002). *Spiking neuron models cambridge university press.* Cambridge.

Gray, C., & Singer, W. (1987). Stimulus-dependent neuronal oscillations in the cat visual cortex area 17. *Neuroscience (Suppl.)*, *22*, 434.

Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Pearson Upper Saddle River, NJ, USA:.

Jirsa, V. K., Fuchs, A., & Kelso, J. A. S. (1998). Connecting cortical and behavioral dynamics: bimanual coordination. *Neural Computation*, *10*(8), 2019–2045.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness* (No. 6). Harvard University Press.

Kahn, D. (2013). Brain basis of self: self-organization and lessons from dreaming. *Frontiers in psychology*, *4*.

Kelso, J. S. (1997). *Dynamic patterns: The self-organization of brain and behavior.* MIT press.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological review*, *95*(2), 163.

Knott, A. (2012). *Sensorimotor cognition and natural language syntax.* MIT press.

Kohler, E., Keysers, C., Umilta, M. A., Fogassi, L., Gallese, V., & Rizzolatti, G. (2002). Hearing sounds, understanding actions: action representation in mirror neurons. *Science*, *297*(5582), 846–848.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, *21*(1), 1–6.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., … Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning* (pp. 1378–1387).

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Münte, T. F., Schiltz, K., & Kutas, M. (1998). When temporal terms belie conceptual order. *Nature*, *395*(6697), 71–73.

Myers, J. L., & O'Brien, E. J. (1998). Accessing the discourse representation during reading. *Discourse processes*, *26*(2-3), 131–157.

Newman, L. I., & Polk, T. A. (2008). The computational cognitive neuroscience of learning and memory: Principles and models. *Advances in Psychology*, *139*, 77–99.

Perfetti, C. A., Britt, M. A., & Georgi, M. C. (2012). *Text-based learning and reasoning: Studies in history.* Routledge.

Ritter, H., & Kohonen, T. (1989). Self-organizing semantic maps. *Biological cybernetics*, *61*(4), 241–254.

Rohde, D. L. (2002). *A connectionist model of sentence comprehension and production* (Unpublished doctoral dissertation). School of Computer Science, Carnegie Mellon University.

Rohde, D. L., & Plaut, D. C. (2003). Connectionist models of language processing. *Cognitive Studies*, *10*(1), 10–28.

Schneider, W., & Körkel, J. (1989). The knowledge base and text recall: Evidence from a short-term longitudinal study. *Contemporary Educational Psychology*, *14*(4), 382–393.

Singer, W. (1986). The brain as a self-organizing system. *European archives of psychiatry and neurological sciences*, *236*(1), 4–9.

Strickert, M., & Hammer, B. (2005). Merge som for temporal data. *Neurocomputing*, *64*, 39–71.

Takac, M., Benuskova, L., & Knott, A. (2012). Mapping sensorimotor sequences to word sequences: A connectionist model of language acquisition and sentence generation. *Cognition*, *125*(2), 288–308.

Takac, M., & Knott, A. (2015). *A simulationist model of episode representations in working memory: Technical appendix* (Tech. Rep.). Tech. Rep. OUCS-2015-01, Dept of Computer Science, University of Otago.

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K. M., & Sedivy, J. C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, 1632–1634.

Vančo, P., & Farkaš, I. (2010). Experimental comparison of recursive self-organizing maps for processing tree-structured data. *Neurocomputing*, *73*(7), 1362–1375.

Van Dijk, T. A., & Kintsch, W. (1983). *Strategies of discourse comprehension.* Academic Press New York.

Van Hulle, M. M. (2012). Self-organizing maps. In *Handbook of natural computing* (pp. 585–622). Springer.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

Wikibooks, The Free Textbook Project. (2017). *Cognitive Psychology and Cognitive Neuroscience/Situation Models and Inferencing.* `https://en.wikibooks.org/wiki/Cognitive_Psychology_and_Cognitive_Neuroscience/Situation_Models_and_Inferencing/`. ([Online; accessed 28-December-2017])

Zwaan, R., Ericsson, K., Lally, C., & Hill, L. (1998). Situationmodel construction during translation. *Manuscript in preparation, Florida State University*.

Zwaan, R. A., & Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological bulletin*, *123*(2), 162.