Building up Conceptual Spaces: An ESOM Supported Strategy

Suelen Mapa de Paula, Micael Cabrera Carvalho, Ricardo Ribeiro Gudwin
School of Electrical and Computer Engineering
University of Campinas
{suelen,micael,gudwin}@dca.fee.unicamp.br

***Abstract*—Intelligent agents need robust knowledge representation schemes to model and solve complex real-world problems. A historical approach is the symbolic representation proposed in classic AI. Although symbolic representations have their appeal, the use of abstract symbols, representing general knowledge about the world, brings limitations to the way agents develop certain cognitive functions, as in the case of language. In the standard symbolic approach, there is no ground for the symbols used internally by the agents, creating a situation known as the symbol grounding problem, as explained by Harnad [1]. To deal with this problem, Gärdenfors [2] introduced a semantic theory named *conceptual spaces*, which attribute meaning to linguistic symbols. The geometry of such spaces forms a robust structure to conceptualize information. In this paper, we use an unsupervised classifier named Evolving Self-Organizing Maps (ESOM) to act as the computational implementation of conceptual spaces. Our results confirmed ESOM's capability to create concepts, aiding agents in reaching a linguistic consensus about different words exchanged during an objects naming game. Besides providing a way for symbols to get meaning on a biologically realistic way, these results also open possibilities for other characteristics of conceptual spaces to be applied on the study of artificial language, as e.g. grammatical language.**

Keywords*—intelligent agents; knowledge representation; conceptual spaces; evolving self-organizing maps; language games.

I. Introduction

Intelligent agents are software entities acting in an autonomous way in complex environments. Usually, they are designed to execute three cognitive functions: perceive the environment, build an internal model to represent learned knowledge and act in the environment modifying its conditions [3]. Building an internal model to represent knowledge plays an essential role among these three, because several cognitive tasks depend on it to be executed, e.g. making plans in order to solve problems, performing inferences in order to discover implicit knowledge, selecting actions and others.

One of the issues related to knowledge representation in intelligent agents is the development of realistic models of the world. A traditional approach is to use a symbolic representation, according to classic AI approaches. This involves the manipulation of chains of symbols following some formalism and inference rules in order to derive new chains of symbols. Usually, it comprises the use of knowledge from an expert to solve specific problems [4]. Although the applications developed under this paradigm have achieved relative success [5], the use of *ungrounded* symbols to represent knowledge brings limitations to the way agents develop certain cognitive functions, as in the case of language, in a wider scope [6].

According to Balkenius, Gärdenfors and Hall [7], the manipulation of symbols is fundamental to the acquisition of language. The way symbolic AI does it, however, is very far from how humans acquire and use language. In symbolic AI, all the information perceived by an agent during its sensory-motor experience is translated directly to linguistic symbols, which are supposed to encode the knowledge they represent. Thus, there is no specific ground associated to the meaning of symbols, besides the symbol itself, implying in a general problem considered in the literature as the symbol grounding problem, as explained by Harnad [1].

To deal with this problem, Gärdenfors [2] introduced a semantic theory (using what he calls "conceptual spaces") to attribute meaning to linguistic symbols. According to him, our mind organizes the information involving perception and memory in standard structures he calls "concepts", which can be modeled in a geometrical way, organized in what he calls "conceptual spaces" (to be addressed further in this article).

Many authors have used computational applications for investigating how meaning can co-evolve with language. A widely adopted approach for this purpose is the notion of language games, which was introduced by the philosopher Ludwig Wittgenstein. Its purpose is to emphasize that the meaning of words, and of linguistic constructions in general, lies in how they are used in concrete activities or "games". In language games, a population of agents must achieve linguistic consensus on how to refer to objects of a context. All agents start playing the games without a language, causing all games to fail in the beginning. From these failures, they can learn to communicate better if they manage to establish a shared language.

Some researchers have already explored the use of conceptual spaces for language games, aiming at representing the meaning of a shared language (e.g. [8], [9]). Although their results are promising, the adopted approaches demand the agent's internal representation to be trained *a priori*, not allowing new concepts to emerge during new experiences.

Therefore, the main contribution of this paper is the adoption of a real-time computational technique capable of building conceptual spaces from scratch. We expect it to provide a way for symbols to get meaning on a biologically realistic way, while allowing for new concepts to emerge during the agent's interactions with the world. Furthermore, we expect it to allow other characteristics of conceptual spaces to be applied on the study of artificial languages, as e.g. in the case of grammatical languages.

In our experiments, language games are used to evaluate the performance of Evolving Self-Organizing Maps (ESOM) [10] in building conceptual spaces. During the games, the agents represent their knowledge about the environment in terms of conceptual spaces and use it to ground the words exchanged during their communication. Our results indicate that conceptual spaces are very efficient for concept representation and learning, corroborating the results obtained by other studies (e.g.[11], [12], [13]).

This paper is organized in the following way: Section II briefly introduces conceptual spaces and discusses some of its main aspects. Section III shows characteristics of ESOM which are particularly relevant to their application as implementations of conceptual spaces. Section IV describes our approach to the problem, defining the language games and the different approaches used in the naming game. Section V provides details about our experiments and results, showing how ESOM can be used to implement conceptual spaces and its performance in the naming game. Finally, section VI provides insights about our findings and possible future applications.

II. Conceptual Spaces

According to Gärdenfors [14], *concepts* are mathematical structures which represent the meaning of words. The most common kind of concepts are object categories, but there might be concepts associated to qualities, actions, events and possibly to all categories and special combinations of words as well. Concepts are defined with the help of *conceptual spaces*, and conceptual spaces are constructed out of *quality dimensions*. The primary role of these dimensions is to represent various *qualities* of objects in different *domains*.

In cognitive psychology, a set of one or more different types of quality dimensions can be *separable* or *integral*. A set of dimensions is said to be integral when all the quality dimensions in the set are equally necessary to characterize a given quality. Color, for instance, is an integral set formed by hue, saturation and brightness quality dimensions. We can not assign hue to an object without assigning both brightness and saturation. A set of dimensions which don't obey that rule is said to be separable.

A domain, then, corresponds to a set of integral dimensions that are separable from all other dimensions. Many domains, such as *temperature* and *weight*, consists of only one dimension. Other domains, such as *color* and *location*, may require multiple dimensions. A conceptual space is a collection of one or more domains, which can be used to assign *properties* to an object.

The notion of property is used to denote information related to a single domain. More specifically, a property is defined as a convex region in some domain. To say that a region R is convex means that for any two points x and y in R, all points between x and y are also in R. Properties, when put together and correlated in a specific way, can be used to represent, for instance, the concept of an *object*. Objects are, therefore, identified as points within conceptual spaces, their properties are represented by regions in specific domains and the category of this object, which is also a concept, is denoted by a collection of regions (properties) and their relations in a conceptual space.

Categories of objects are special concepts which can be represented by *prototypes*. Prototypes are special objects of a category, because they can be used to derive the full region which comprises the category. In Figure 1, we present a conceptual space formed by a single domain, which is represented by the quality dimensions X and Y (axis), and the black points serve as prototypes for each category. From the full set of points, we can derive a Voronoi diagram, which creates partitions of the conceptual space, where each region comprises a property.

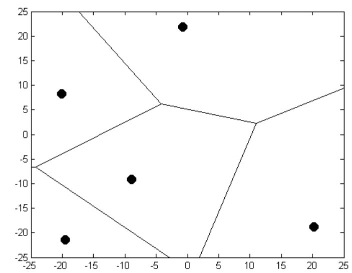


Fig. 1: Deriving properties from prototypes in a conceptual space, formed by a single domain, using a Voronoi diagram.

According to Gärdenfors [2], the use of conceptual spaces provide a different approach when compared to symbolic AI. In symbolic AI, the assignment of semantics to symbols requires an external interpretation. In the current approach, the semantics is implicit in the definition of conceptual spaces. Because of this, we can say that the use of conceptual spaces might be a possible solution to the symbol grounding problem. Furthermore, conceptual spaces provide a robust framework for learning concepts for language, because to have a space partitioned into a finite number of regions implies that a finite number of words can be used to refer to such regions.

III. Conceptual Spaces and Evolving Self-Organizing Maps

In machine learning, clustering tools are methods capable of detecting similarities, regularities and correlations in the input data, clustering it in groups (clusters) [15]. Conceptual spaces, as shown by Gärdenfors [2], work in a similar way, since created domains are used to categorize objects sharing similar properties, and can, therefore, be likewise implemented.

There are several clustering algorithms available in the literature that could be used for implementing conceptual spaces (e.g. [16], [17]). Our choice, however, was based on the evaluation of their compatibility to the conceptual space definition and their applicability to the grounding of linguistic symbols. Additionally, we have defined that it was desirable for the algorithm to have biological plausibility, besides being able to learn new concepts during the agent's interactions, without prior training.

We have selected a self-organizing artificial neural network named Evolving Self-Organizing Map (ESOM), proposed by Deng and Kasabov [10], as a candidate. ESOM is based on the Self-Organizing Map (SOM)[1], proposed by Teuvo Kohonen [18]. According to Silva, Spatti and Flauzino [15], SOM

[1]SOM is also known as Kohonen's Self-Organizing Map or, simply, Kohonen's network.

was inspired in the human cortex and is commonly applied to the resolution of problems involving pattern classification and data clustering. Although SOM can be used in different kinds of problems, it has a severe limitation which precludes its use in the learning of conceptual spaces: the need of *a priori* training.

The fact that ESOM is an *evolutionary* neural algorithm makes it particularly interesting for modeling conceptual spaces, mainly due to its ability to perform unsupervised online learning[2]. The network structure, formed during the learning phase, is self-adaptive and incremental. New clusters and possibly outliers may appear and disappear, creating a topology according to the data organization. These characteristics give ESOM flexibility and efficiency, allowing it to learn good representations for the input data. In this paper, the configuration of nodes will be used to segment the conceptual space in semantic regions (clusters), where each region can represent a category of object.

Deng and Kasabov [10] also compared the performance of ESOM with other neural networks, such as SOM and Growing Neural Gas, in challenging scenarios. Their results indicate that their approach is faster and more effective.

IV. Proposed Approach

In this section we introduce the details of the computational simulation we developed, a language game called naming game. ESOM was adopted for building conceptual spaces during the games (ESOM Conceptual Space – ECS). We have tested three different strategies for naming, further detailed in Subsection IV-C.

A. Naming games

The concept of language game was introduced by the philosopher Ludwing Wittgenstein as a tool to explore the characteristics of linguistic interactions. There are many different implementations of language games[3] and each one deals with a specific aspect of language, its emergence and evolution. For our experiments, we adopt the *naming games*, introduced by Steels and Loetzsch [19].

According to Steels and Loetzsch [19], a naming game is the simplest possible kind of language game imaginable. It is a game of references in which the speaker tries to draw the attention of the listener to an object in the context by naming a characteristic feature of the object. Thus, a population of agents must create and maintain a shared set of names to be assigned to a set of objects. Such names are exchanged during a game and can spread between the population only by interactions between the agents.

Our games are similar to the naming game implemented by Wellens [22], but with an extra stage for categorization on the game script, implemented by an ECS. This is a necessary change since, in Wellens' game, names have a meaning only when the referred object is physically present in the context.

Barsalou [6] explains that, on language, a meaning doesn't refer to physical entities, but to mental projections created by past experiences during cognitive activities. Conceptual spaces will, thus, allow this projections to be created and, consequently, the agent will recognize an object by the relation established between its name and its meaning, instead of the relation between the name and the object itself. We expect this change not to affect the results shown by Wellens [22], but to expand them instead, allowing other situations to be explored, like text interpretation (without the presence of physical objects) and object variability within the same class.

B. The grounding naming game

According to Steels and Loetzsch [19], a naming game demands a population of agents and a world consisting of a set of individual objects. For every game, a context is generated containing a subset of the world. Two agents, speaker and listener, are randomly chosen from the population and are confronted with the same context. In our implementation, agents extract object features to create conceptual spaces, which are used for categorizing the objects in the context.

Given its conceptual space and a purpose, the speaker conceptualizes a semantic structure, which will be used by the linguistic system to produce a name. The same will happen to the listener that, upon receiving a name, will use its conceptual space to find a corresponding semantic concept to interpret it. The following steps represent one iteration of our naming game:

1) Two agents (1 speaker and 1 listener) are chosen randomly;
2) A context is generated with a random set of world objects;
3) Both agents classify each object based on their conceptual spaces (categorization phase);
4) The speaker mentally picks one referent object and categorizes it; If there is a name (a single three syllable word) for that category object in its lexical dictionary[4], the speaker utters it; otherwise, the speaker creates a new name and utters it;
5) The listener hears the name and search for it in its dictionary; If there is an entry for such name in its lexical dictionary, the listener gets the features associated to that category and points them; otherwise, the listener points to invalid features to indicate it does not have a category associated to that name;
6) The speaker classifies the features, agreeing or disagreeing with the listener; if they are in the same class of the referent object, the speaker indicates a communication success and rewards the chosen name by increasing its weight; otherwise, the speaker indicates a communication failure, punishes the name and indicate the correct referent object to the listener;
7) The listener detects the speaker's feedback; if the communication succeeded, the listener also rewards the name used during the communication; otherwise, it classifies the referent object, retrieves the object's category and adds to its lexical dictionary the name used by the speaker.

[2]Online learning uses a data stream as input.

[3]Some examples of language games include: the naming games [19], the guessing games [20] and the description games [21].

[4]The lexical dictionary is a data structure created by each agent for maintaining pairs (category, set of names), associating each category to a possible set of names referring to it. To each name in the set, there is also an associated weight.

C. Strategies for the naming game

During the game, different names compete among themselves to describe an object/category. Agents will score each of these names based on its success (both agents agree on the same name) or failure (no agreement is reached) during the communication, according to different strategies, as shown by Wellens [22]. In this paper, we focus on three of them:

- **Minimal Naming Game (MNG)** the agent keeps multiple names for each category and in the case of communication success, both agents remove all the other names for that category, keeping only the winning name. In the case of failure, the listener agent adopts the name proposed by the speaker agent.
- **Basic Lateral Inhibition (BLI)** the agents keep multiple names for each category, scoring each name with a real value in the interval $(0, 1]$. In the case of a communication failure, the speaker decreases the score of the used name by δ_{dec} and the listener adopts it with the initial score of $s_{initial}$. In the case of a communication success, both agents increase the score of the used name by δ_{inc} and decrease the score of other names by δ_{inh}. Any name can be removed from the lexical dictionary, if its score reaches a value lower than or equal to 0.
- **Interpolated Lateral Inhibition (ILI)** is similar to the Basic Lateral Inhibition, incrementing (reinforcing) or decrementing (inhibiting) the scores (s) with a given δ that is interpolated between 0 or 1. For reinforcement: $s = s + \delta_{inc}(1 - s)$ or for inhibition: $s = s - \delta_{inh}s$.

V. Experimental Results

Our results are divided in two parts. In the first part, we implemented the ESOM algorithm, as proposed by Deng and Kasabov [10]. Before adding it to the naming game, we have verified whether it may be used as a conceptual space (named ECS). In the second part, we present results for the ECS in the naming game.

A. ESOM implementation

Our ESOM was implemented as a Java application, following the algorithm described by Deng and Kasabov [10]. The ESOM starts with no nodes and, during each learning iteration, it self-updates to categorize the input data, creating new nodes and new connections when necessary (i.e. if the network is empty or the distance - Euclidean distance - between input data and the nearest node is greater than the established threshold ϵ). Here, ϵ is very important, since it clearly controls the growing rate of the network, and was adjusted based on the tasks we were working on. When a new node is created, its prototype represents exactly the input data, which can be adjusted in later iterations. If the distance between the input data and the nearest node is smaller than ϵ, however, the connections of such node are updated, and no new node are added. The connections between nodes have a fundamental role of maintaining relationships between neighbor nodes. Their strength (relationship) is determined by their distance, which is updated by the learning rate $0 < \gamma < 1$. Every T steps, the connection between the two most distant nodes (i.e. the nodes with the weakest connection) is pruned.

Figure 2 shows the evolution of an ESOM. The initial neighborhood of each new node is defined to be the two nodes closest to it. In our example, node 3 was created and connected with nodes 2 and 1. After some time, the connection between nodes 3 and 2 was pruned. The pruning occurs when the distance between two nodes is too big (indicating a weak connection). Each node has a representative area bounded by ϵ.

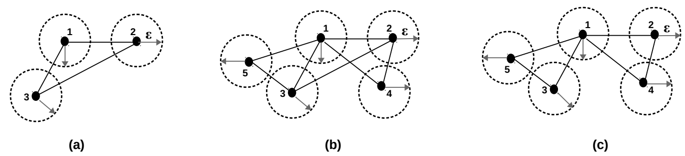


Fig. 2: The evolution of an ESOM. (a) The nodes 1, 2 and 3 are created; (b) Two new nodes, 4 and 5, are added; (c) The weakest connection is prunned. Adapted from Deng and Kasabov [10].

For more details on ESOM, we recommend the reader to refer to the work of Deng and Kasabov [10].

B. ESOM as conceptual spaces

In order to validate our ESOM implementation, and verify whether it can operate as a conceptual space, a synthetic dataset with 12 classes, composed of 2 features each, and 1000 data points was created. After defining the initial classes, we randomly sampled the points from such classes and, for each sampled point, we have added random noise following a normal distribution ($\mu = 0$, $\sigma = 2$). The resulting points are shown in Figure 3, and their ground-truth classes are represented by different colors in Figure 4. It is easy to see that near to their boundaries there is no clear distinction between neighbor classes, since the dataset was designed to allow small class overlapping and the ESOM should be able to generalize from the noisy data.

Fig. 3: Graphical plot of our synthetic dataset, composed of 12 classes (with 2 features each) and 1000 data points. Each point was randomly sampled from one of the 12 classes and a random noise was applied to it, following a normal distribution ($\mu = 0$, $\sigma = 2$).

After defining ESOM's initial parameters[5], the training phase began. At each iteration, a point was presented to the ESOM for learning and labeling. This way, the dataset

[5]For reproducibility: threshold $\epsilon = 9.0$ and learning rate $\gamma = 0.4$.

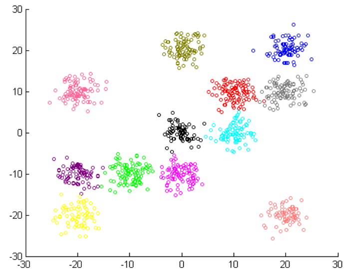


Fig. 4: Graphical plot of our synthetic dataset shown in Figure 3, with the original class for each point represented by its color.

representation is constructed in real-time, without any prior training.

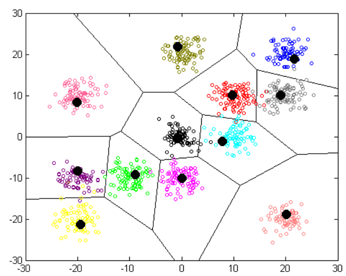


Fig. 5: Real-time labeled points by an unsupervised technique (ESOM), with the Voronoi tessellation for each class at the end of the 1^{st} training phase. Each color represents a different label and the large black dots represent prototype points (nodes/neurons of the map), determined by the ESOM.

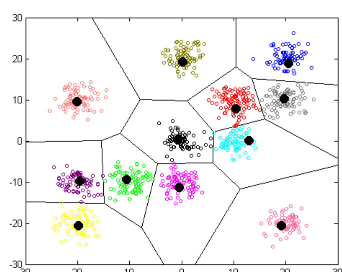


Fig. 6: Real-time labeled points by an unsupervised technique (ESOM), with the Voronoi tessellation for each class at the end of the 10^{th} training phase. Each color represents a different label and the large black dots represent prototype points (nodes/neurons of the map), determined by the ESOM.

In Figures 5 and 6, we show the real-time labeled points obtained through the training phases 1 and 10, respectively, and the corresponding tessellation at the end of each phase. The nodes act like prototype points, representing the dataset configuration. This way, each node is segmenting the space in a semantic domain (represented by a Voronoi tessellation), shaped as convex regions, determining that each point within one region shares the same property, i.e. the points are similar. According to Gärdenfors [14], the tessellation provides a geometric answer for how a similarity measure, with a set of prototypes, can determine a set of categories.

One can argue that, through time, ESOM tends to stabilize the regions. In Figure 5 each point is colored with the label assigned to it during the first time it was presented to the ESOM. Although the spatial division and the labels were mostly correctly defined, there were still some outliers, showing that the prototype points were moving during the classification of other points. Figure 6, on the other hand, shows the colored points after 10 training iterations, with less outliers. The difference in the regions between the first and the tenth iterations is also small, indicating that the ESOM could successfully build a good representation for the problem after seeing the data only once.

C. ECS's performance in the naming game

For the experiments we describe here, we have considered a population of 50 agents and a world with 5 types of abstract objects, each of them described by a vector composed of 4 quality dimensions, with random values in the interval $[0, 1]$. Each agent is capable of creating and maintaining its own lexical dictionary and conceptual space (ECS), as previously described.

During each game, two agents were randomly selected (speaker and listener) to interact and the context was composed of all the 5 objects. They then executed an iteration of the naming game following the steps 3-7 described in Subsection IV-B.

We ran a total of 100 experiments, each of them with 12500 iterations of the naming game[6]. The experiment measured the population alignment (alignment success), given by: (1) the communicative success, i.e., the listener's ability to correctly identify the name uttered by the speaker, pointing to the correct referent and (2) the agent's preference with respect to the name, i.e., if both the speaker and the listener would choose the same name to point to the object. If both criteria are met, the alignment success is 1, otherwise, 0.

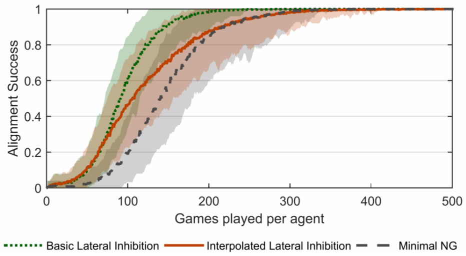
The average population alignment for our experiments was calculated for each naming strategy we have addressed (MNG, BLI and ILI) and our results are shown in Figure 7a. For comparison, in Figure 7b we also offer the results obtained by Wellens [22].

Both our and Wellens' [22] experiments indicate that the BLI and the ILI strategies have fast convergence at the beginning of the experiment, while the MNG strategy converges at a slower rate. A possible explanation, provided by Wellens [22], is that the MNG limits the agent's memory capacity, removing all the adversary names instead of a gradual inhibition. The BLI strategy, however, have the best results, since it doesn't remove all of the adversarial names and promotes linear reinforcements in the scores, instead of interpolated values[7].

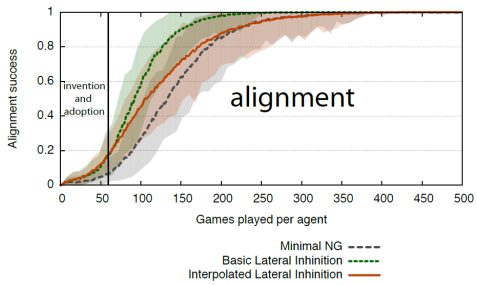
Our results (Figure 7a) are very similar to the results presented by Wellens [22] (Figure 7b), indicating that the use of ECS could match the performance of a simple dictionary-based implementation. We emphasize that because our agents

[6]For an average of 500 games per agent, we have $(2 * (number\ of\ games \div (size\ of\ population))$.

[7]For a detailed explanation, please refer to Wellens [22], subsection 2.4.1.



(a) Alignment success reached by our ECS implementation



(b) Alignment success reached by Wellens (2012). Reproduced from Wellens [22].

Fig. 7: Alignment success for the Lateral Inhibition and Minimal Naming Games strategies. 7b shows the alignment success reached by Wellens [22] and 7a shows the alignment success reached by our method. Both lateral inhibition strategies uses the parameters: $\delta_{inc} = \delta_{dec} = 0.1$, $\delta_{inh} = 0.5$ and $s_{initial} = 0.5$.

were in a dynamic environment with objects unknown *a priori*, their semantic domains were gradually formed while they participated in the games. The ECS, then, acted as an internal environment model for each agent, grounding the linguistic symbols used during the agents interactions. At the end of the games, each agent had 5 prototype points left in their ECS, representing the 5 objects from the environment.

VI. Conclusion

In this paper we have shown how an unsupervised online classifier, named Evolving Self-Organizing Map (ESOM), can formalize the computational implementation of conceptual spaces. Our approach could successfully identify the different clusters in our synthetic dataset and categorize them using prototype points (see subsection V-B), indicating that the ESOM can be used for building conceptual spaces on intelligent agents.

We also emphasize that our proposal (ECS) extends the model adopted by Wellens [22], allowing the meaning of the names to be learned in real-time, while the agents learn to identify the objects. This could be useful, for instance, in problems involving noisy data or category naming, when many different objects should be referred with the same name (e.g. toys, buildings, fruits etc).

These results could lead the way to the usage of conceptual spaces in more challenging tasks, as in the field of grammatical language. To Gärdenfors [14], there are strong arguments in favor of the use of conceptual spaces for such problems, as it allows us to study syntactic aspects of the word classes through their cognitive grounding. This may be a promising way to develop compositional linguistic expressions.

Acknowledgment

We would like to thank CAPES, CNPq and Samsung for partially funding this project.

References

[1] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335–346, 1990.

[2] P. Gärdenfors, *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[3] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[4] N. J. Nilsson, *Principles of artificial intelligence*. Morgan Kaufmann, 2014.

[5] S.-H. Liao, "Expert system methodologies and applicationsa decade review from 1995 to 2004," *Expert Systems with Applications*, vol. 28, no. 1, pp. 93 – 103, 2005.

[6] L. W. Barsalou, "Perceptual symbol systems," *Behavioral and brain sciences*, vol. 22, no. 04, pp. 577–660, 1999.

[7] C. Balkenius, P. Gärdenfors, and L. Hall, "The origin of symbols in the brain," in *Proceedings of the 3rd International Evolution of Language Conference, Ecole Nationale Superieure des Telecommunications*, 2000, pp. 13–17.

[8] P. Vogt, "The emergence of compositional structures in perceptually grounded language games," *Artificial intelligence*, vol. 167, no. 1, pp. 206–242, 2005.

[9] H. Eyre and J. Lawry, "Language games with vague categories and negations," *Adaptive Behavior*, pp. 1–15, 2014.

[10] D. Deng and N. Kasabov, "Esom: An algorithm to evolve self-organizing maps from on-line data streams," in *Neural Networks, IEEE-INNS-ENNS International Joint Conference on*, vol. 6. IEEE Computer Society, 2000, pp. 6003–6003.

[11] I. Lee and B. Portier, "An empirical study of knowledge representation and learning within conceptual spaces for intelligent agents," in *Computer and Information Science, 2007. ICIS 2007.* IEEE, 2007, pp. 463–468.

[12] T. Takala, "Preconceptual creativity." in *Proceedings of the Sixth International Conference on Computational Creativity*, June 2015, pp. 252–259.

[13] M. S. P. M. Agres, K. and G. Wiggins, "Conceptualizing creativity: From distributional semantics to conceptual spaces." in *Proceedings of the Sixth International Conference on Computational Creativity*, June 2015, pp. 118–125.

[14] P. Gärdenfors, *The geometry of meaning: Semantics based on conceptual spaces*. MIT Press, 2014.

[15] I. N. da Silva, D. H. Spatti, and R. A. Flauzino, *Redes Neurais artificiais para engenharia e ciências aplicadas curso prático*. Artliber, 2010.

[16] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[17] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[18] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[19] L. Steels and M. Loetzsch, "The grounded naming game," *Experiments in cultural language evolution. Amsterdam: John Benjamins*, 2012.

[20] J. De Beule and B. K. Bergen, "On the emergence of compositionality," in *Proceedings of the 6th International Conference on the Evolution of Language*, 2006, pp. 35–42.

[21] R. Van Trijp, "The emergence of semantic roles in fluid construction grammar," in *Proceedings of the 7th International Conference on the Evolution of Language (EVOLANG 7)*, 2008, pp. 346–353.

[22] P. Wellens, "Adaptive strategies in the emergence of lexical systems," in *Proceedings of the 21st Belgian-Dutch Conference on Machine Learning*, vol. 20. Springer Verlag, 2012, pp. 76–88.