Information Modelling and Knowledge Bases XXXIV
M. Tropmann-Frick et al. (Eds.)

doi:10.3233/FAIA220496

An Exploratory Research on the Expression of Knowledge and Its Generation Process Based on the Concept of "Dark-Matter"

Xing Chen[1]
Department of Information & Computer Sciences
Kanagawa Institute of Technology, Japan
chen@ic.kanagawa-it.ac.jp

Abstract. Research work on machine learning techniques has been going on since the invention of computers. With the development of machine learning techniques, researches on how knowledge is expressed in computers and the learning process of knowledge have become more important. Unlike other machine learning models such as artificial neural networks, in the previous work of this paper, a machine learning model based on the concept of "dark-matter" is presented. In this model, matrixes are used to represent temporal and non-temporal data. The term "matter" is used to denote non-temporal data. The term "dark-matter", on the other hand, is used to represent temporal data. In this paper, an exploratory research on the expression of knowledge and its generation process based on the concept "dark-matter" is presented. A case study is used to illustrate how knowledge is generated and expressed. The contribution of this paper is that new methods of knowledge generation and expression are proposed based on the concept of "dark-matter". In the paper, first, the concept of "dark-matter" is briefly reviewed. After that, the methods of knowledge generation and knowledge expression are illustrated with examples. The process of knowledge generation is also illustrated with examples. Finally, the relationship between knowledge and "dark-matter" is revealed.

Keywords. Knowledge, knowledge presentation, knowledge generation, machine learning, semantic space, spatiotemporal space

1. Introduction

The expression of knowledge on a computer is an essential requirement for implementing knowledge processing on the computer. Although many methods and models are proposed for expressing knowledge, it is still not enough to process knowledge on computers to achieve the level of the human brain. In addition, these methods are not sufficient to express knowledge generated based on machine learning. The machine learning indicates the calculation methods that allow people to achieve the desired function without programming code. That is, people can create processing models through the training process to achieve the desired function. Machine learning is divided into supervised learning, unsupervised learning, and reinforcement learning.

[1] Xing Chen, 1030 Simo-Ogino, Atsugi-shi, Kanagawa 243-0292, Japan; chen@ic.kanagawa-it.ac.jp

For supervised learning, supervised data set is required, which is the expected output data set for a given input data set. Unsupervised learning does not require supervised data. For reinforcement learning, an agent needs to be created and trained to get the maximum reward in order to get the output data for a given input data. In this paper, case studies on reinforcement learning are used to illustrate how knowledge is generated and expressed. It is an exploratory research on the expression of knowledge and its generation process based on the concept of "dark-matter" [1].

The concept of "dark-matter" is developed based on the research works of semantic computing models [2, 3, 4, 5]. In the semantic computing models, matrixes are used to present "meaning" of data. In the models, input data are mapped through mapping matrixes into semantic spaces and presented as points in semantic spaces. Another data which present different meanings, referred to as meaning data, are also mapped to the same space. Distances of the points among input data and meaning data are performed. In this way, the semantic calculation is transmitted to calculate Euclidean distances of those points. For example, in the case for implementing semantic query, query data presenting query keywords are mapped into a semantic space and summarized as a point in the space. Retrieval candidate data are also mapped into the semantic space as other points. Euclidean distance is calculated between the query point and each retrieval candidate point. When the distance of a retrieval candidate is shorter than a given threshold, its relative retrieval candidate is extracted as the query output.

Two methods, Mathematical Model of Meaning (MMM) [4, 5] and Semantic Feature Extracting Model (SFEM) [2, 3], are developed on semantic space creation. In MMM, a common data set, for example an English-English diction is used to create the semantic space. In SFEM, the semantic space is created based on special defined data sets according to the requirement of applications.

After the semantic space is created, input data will be mapped to the space and the Euclidean distance calculation between the mapped data points in the space will be performed. Mapping matrixes are required to map input data into the semantic space. In SFEM, mapping matrixes are defined according to the applications of the models [6-14]. Many methods are developed to create mapping matrixes for applying the model in the areas of semantic information retrieving [9, 10, 14], semantic information classifying [11], semantic information extracting [12], and semantic information analyzing on reason and results [13], etc. Furtherly a method is developed to create the mapping matrixes through deep-learning [15]. Elements of the matrixes presenting semantic spaces are previously defined, which are different from those other models, such as the artificial neural network model and deep-learning artificial neural network model [16-19].

A mechanism is furtherly developed based on the semantic space model to implement the basic logic computation to implement true and false judgement which is the foundational mechanism required by machine learning [20]. The mechanism is applied to simulate unmanned ground vehicle control [21]. Temporal data processing is required for the ground vehicle control. In paper [1], a new model is presented for the temporal data processing. In the model, the word "matter" is used to represent features of spaces which are related to the non-temporal data. The word "dark-matter" is used to represent of spaces which elements are temporally changed. The word "energy" is used to represent matrixes which are used to generate output data. "Dark-matter" is a matrix which elements are generated during training processing.

In this paper, an exploratory research is presented on knowledge representation and its generation process based on the concept of "dark-matter." Case studies are used to illustrate how knowledge is generated and expressed. The contribution of this paper is that new methods of knowledge expression and generation are proposed based on the concept of "dark-matter." The concept of "dark-matter" is reviewed in Section 2 of this paper. In Section 3, the relationship between "knowledge" and "dark-matter" is described. The method of knowledge expression is also illustrated with examples in this section. Then, the knowledge generation method based on training processing is illustrated. Finally, the conclusions of this paper are presented.

2. The machine learning model created based on the concept "dark-matter"

In this section, the machine learning model created based on the concept "dark-matter" is briefly reviewed through an example as shown in Figure 1. This model is called "dark-matter learning model" in the following. In this model, matrix multiply is referred to as "space mapping" or simply "mapping." For example, the following equation (1) is described as to mapping a matrix **X** by a matrix **E** to a new space **Y**. That is, a space represented by **X** is mapped to a new space represented by **Y**.

$$\mathbf{Y} = \mathbf{X}*\mathbf{E} \qquad (1)$$

The matrix **X** is further divided into two parts as shown in Figure 1. The first part is the first column of the matrix **X,** which is referred to as "***matter***". The second part of **X** is referred to as the "***dark-matter***", which is constructed by the second to the last column of the matrix **X**. In Figure 1, "***matter***" is an index matrix correlated with sensor data. As sensor data are "visible", it is referred to as matter. Elements in the matrix of "***dark-matter***" are randomly filled. As the matrix of the dark-matter is created randomly filled, it is referred to as "chaotic space".

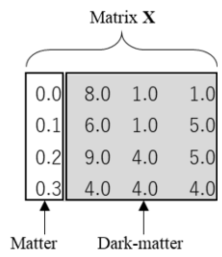


Figure 1. Creating a "chaotic space"

The learning process is to change the "chaotic space" to "ordered space." The learning model is considered as a state machine. A state in the state machine will be changed from one to others, which is referred to as state transition. State transition diagrams are used to present the state transition. Figure 2 is an example of state transition diagram. In the figure, a circle and the number in the circle represents a state, an arrow represents a state transiting from one to another one. The numbers by the arrows represent the condition required for the state transition. For example, the state will be transited from state "0.0" to the state "0.1" if the input data is "0.0".

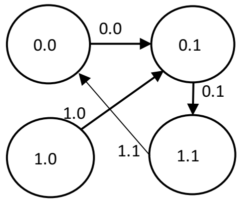


Figure 2. An example of state transition

Figure 3 presents how to create the "ordered space" from the "chaotic space" based on the state transition. The example of the state transition diagram shown in Figure 2 is used to illustrate it. In Figure 2, state "0.0" is the start state and the state transition from "0.0" to "0.1" is the first step of the state transition. Figure 3 (a) shows a chaotic space and Figure 3 (b) shows the first step state transition. In the second step, the state "0.1" transits to the state "0.3". as shown in Figure 3 (c). When all the state transition diagram shown is represented as Figure 3 (d), we say that we created an "ordered space" from the "chaotic space." In the figure, the "matter" is the first row of the matrix, and the "dark-matter" is the matrix constructed by the second, third and fourth row, painted in gray.

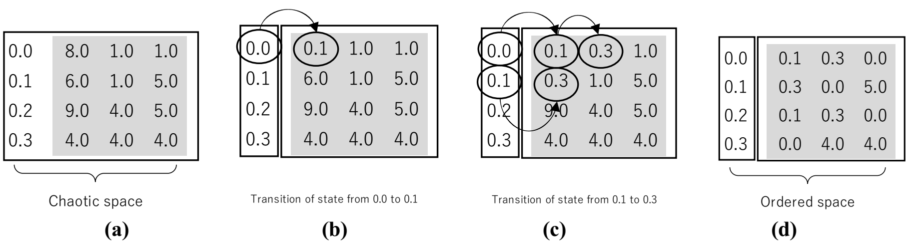


Figure 3. Creating an "ordered space" from a "chaotic space"

If $\mathbf{X}^{-1}$ is an inverse matrix of the matrix **X**, the matrix $\mathbf{X}^{-1}$ is referred to as an "***antimatter space***." An example of the "antimatter space" is shown in Figure 4 (b), which is the inverse matrix of X shown in Figure 4 (a).

If **Y** is a matrix of the desired output result, by applying antimatter space to the matrix **Y**, a new matrix **E** is created, which is referred to as "***energy***" as shown in equation (2). The calculation presented by the equation (2) is referred to as "*learning*" or "*training*" calculation.

$$\mathbf{E} = \mathbf{X}^{-1} * \mathbf{Y} \qquad (2)$$

As the example shown in Figure 4 (c), matrix **Y** is the results of logical calculation "and," "or" and "xor" represented as three vectors $\mathbf{Y}_{\text{and}}$, $\mathbf{Y}_{\text{or}}$ and $\mathbf{Y}_{\text{xor}}$. The "***energy***" **E** is represented as three vectors $\mathbf{E}_{\text{and}}$, $\mathbf{E}_{\text{or}}$ and $\mathbf{E}_{\text{xor}}$ as shown in Figure 4 (d).

X	Matter	Dark-matter		
	0.0	0.1	0.3	0.0
	0.1	0.3	0.0	5.0
	0.2	0.1	0.3	0.0
	0.3	0.0	4.0	4.0

Space

(a) Space

X^{-1}			
-5.0	0.0	5.0	0.0
7.8	0.5	0.7	-0.6
0.7	-0.2	-0.2	0.2
-0.4	0.2	-0.1	0.0

Antimatter space

(b) Antimatter space

Input	Y_{and}	Yor	Y_{xor}
00	0	0	0
01	0	1	1
10	0	1	1
11	1	1	0

Calculating results

(c) Input and output

$E_{and}=X^{-1}*Y_{and}$	$E_{or}=X^{-1}*Y_{or}$	$E_{xor}=X^{-1}*Y_{xor}$
0.0	5.0	5.0
-0.6	0.6	1.2
0.2	-0.2	-0.4
0.0	0.1	0.0

Energy

(d) Energy vectors

Figure 4. The concept of the "matter", "dark-matter", "anti-matter" and "energy"

When energy **E** is applied to the space **X**, the desired output result **Y** is calculated as shown in equation (3).

$$\mathbf{Y} = \mathbf{X}*\mathbf{E} \tag{3}$$

For the example shown in Figure 4, a single regression analysis function is used to map input data on the matter-space. As shown in Figure 4, the input data presented as binary number "00", "01", "10" and "11" are mapped respectively to "0.0", "0.1", "0.2" and "0.3" on the matter-space. The mapped value on the matter-space is used as index to extract a row vector of matrix X. For example, if the input data is given as "10", it is mapped to the value "0.2". After that, the mapped value "0.2" is used as the index to extract the row vector **V**, [0.2, 0.1, 0.3, 0.0] from X. When the energy matrix $\mathbf{E}_{and}$ is applied to the extracted vector **V** as **V*E**, the calculation result "0" is obtained which is the logic value of "0 and 1".

3. A knowledge representation method with "dark-matter"

In this paper, a new method is proposed to express knowledge by matrixes. Based on Oxford English dictionary [22], "knowledge" is defined as "facts, information, and skills acquired through experience or education". In the following, an example is used to illustrate how an agent to acquire skills through experience.

In this example, there is a maze, represented by a 4 x 4 matrix, as shown in Figure 5 (a). An agent is created in this case study which should move from a start position to a goal position. The start position of the agent is at row 2 and column 1, which is marked with the character "S". A position will be represented by row and column number as (*row number, column number*) in the following. Thus, the start position of the agent is represented as (2, 1) with the mark "S". The goal position of the agent is (2, 4) with the mark "G". The agent will go along the path shown by the arrow-mark "→". From the start position to the goal position. That is, the agent goes through the points (2,1), (2, 2), (3, 2), (4, 2), (4, 3), (4, 4) and (3, 4) and reaches to (2, 4).

By defining states of the agent as its positions on the maze matrix, a space matrix can be defined. As there are 16 possible positions for the agent on the maze matrix, 16 values from 0.0 to 1.5 are used to represent the index as shown in Figure 5 (b).

	1	2	3	4
1	0	0	0	0
2	S	1	0	G
3	0	1	0	1
4	0	1	1	1

(a) A maze matrix

	1	2	3	4
1	0.0	0.4	0.8	1.2
2	0.1	0.5	0.9	1.3
3	0.2	0.6	1.0	1.4
4	0.3	0.7	1.1	1.5

(b) The position index values of the maze matrix

Figure 5. A maze matrix shown for an agent moving from "S" to "G"

Using the values of index as the matter, and a 16 x 15 matrix with the random values as the dark-matter, a 16 x 16 space matrix is defined as shown in Figure 6 (a). The path from the starting position to the goal position is indicated in Figure 5 (a). Using the path indicated in Figure 5 (a), the agent can get its next position of the current position. For example, at the start position, (2, 1), its next position is (2, 2). By using the index values, the current position index value is 0.1 and its next position index value is 0.5. Therefore, the index value 0.5 is recorded at the dark-matter matrix at row 2 and column 2 of the space matrix. Same as that, the next position index value 0.6 of is recorded at row 2 and column 3 and row 6 and column 2 as shown in Figure 6 (b), if the current position index value is 0.5.

0.0	49.0	90.0	77.0	96.0	27.0	25.0	39.0	33.0	17.0	56.0	24.0	57.0	26.0	98.0	14.0
0.1	20.0	19.0	41.0	26.0	61.0	10.0	34.0	66.0	82.0	31.0	59.0	71.0	48.0	90.0	84.0
0.2	44.0	31.0	38.0	50.0	57.0	45.0	52.0	48.0	47.0	32.0	67.0	45.0	47.0	61.0	92.0
0.3	63.0	18.0	48.0	61.0	10.0	74.0	79.0	42.0	42.0	23.0	94.0	95.0	17.0	32.0	22.0
0.4	13.0	47.0	78.0	22.0	16.0	31.0	29.0	95.0	94.0	56.0	69.0	87.0	22.0	79.0	35.0
0.5	47.0	35.0	86.0	78.0	11.0	81.0	55.0	26.0	91.0	59.0	12.0	93.0	95.0	57.0	81.0
0.6	38.0	31.0	100.0	80.0	62.0	29.0	72.0	11.0	14.0	41.0	53.0	23.0	17.0	35.0	81.0
0.7	41.0	22.0	44.0	27.0	18.0	50.0	49.0	15.0	71.0	70.0	44.0	66.0	95.0	40.0	90.0
0.9	23.0	77.0	71.0	36.0	50.0	50.0	48.0	22.0	47.0	47.0	70.0	10.0	25.0	88.0	49.0
0.9	82.0	58.0	47.0	61.0	79.0	57.0	27.0	45.0	20.0	10.0	29.0	48.0	71.0	33.0	86.0
1.1	54.0	58.0	30.0	98.0	61.0	88.0	62.0	81.0	31.0	54.0	52.0	44.0	49.0	27.0	86.0
1.2	37.0	77.0	40.0	80.0	23.0	10.0	10.0	15.0	58.0	94.0	25.0	79.0	55.0	83.0	53.0
1.3	42.0	78.0	24.0	19.0	55.0	55.0	51.0	14.0	68.0	52.0	29.0	31.0	45.0	41.0	55.0
1.4	50.0	59.0	32.0	62.0	31.0	32.0	92.0	60.0	28.0	26.0	79.0	76.0	42.0	80.0	16.0
1.5	22.0	50.0	51.0	28.0	21.0	79.0	98.0	43.0	64.0	90.0	93.0	29.0	46.0	14.0	24.0
1.6	19.0	68.0	66.0	79.0	57.0	75.0	25.0	76.0	85.0	84.0	75.0	90.0	45.0	57.0	84.0

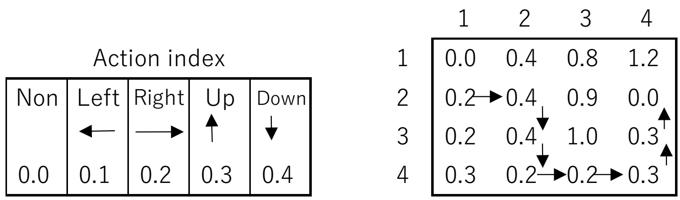
(a) The space matrix with the matter and dark-matter matrix

0.0	49.0	90.0	77.0	96.0	27.0	25.0	39.0	33.0	17.0	56.0	24.0	57.0	26.0	98.0	14.0
0.1	0.5	0.6	0.7	1.1	1.5	1.4	1.3	66.0	82.0	31.0	59.0	71.0	48.0	90.0	84.0
0.2	44.0	31.0	38.0	50.0	57.0	45.0	52.0	48.0	47.0	32.0	67.0	45.0	47.0	61.0	92.0
0.3	63.0	18.0	48.0	61.0	10.0	74.0	79.0	42.0	42.0	23.0	94.0	95.0	17.0	32.0	22.0
0.4	13.0	47.0	78.0	22.0	16.0	31.0	29.0	95.0	94.0	56.0	69.0	87.0	22.0	79.0	35.0
0.5	0.6	0.7	1.1	1.5	1.4	1.3	55.0	26.0	91.0	59.0	12.0	93.0	95.0	57.0	81.0
0.6	0.7	1.1	1.5	1.4	1.3	29.0	72.0	11.0	14.0	41.0	53.0	23.0	17.0	35.0	81.0
0.7	1.1	1.5	1.4	1.3	18.0	50.0	49.0	15.0	71.0	70.0	44.0	66.0	95.0	40.0	90.0
0.9	23.0	77.0	71.0	36.0	50.0	50.0	48.0	22.0	47.0	47.0	70.0	10.0	25.0	88.0	49.0
0.9	82.0	58.0	47.0	61.0	79.0	57.0	27.0	45.0	20.0	10.0	29.0	48.0	71.0	33.0	86.0
1.1	1.5	1.4	1.3	98.0	61.0	88.0	62.0	81.0	31.0	54.0	52.0	44.0	49.0	27.0	86.0
1.2	37.0	77.0	40.0	80.0	23.0	10.0	10.0	15.0	58.0	94.0	25.0	79.0	55.0	83.0	53.0
1.3	42.0	78.0	24.0	19.0	55.0	55.0	51.0	14.0	68.0	52.0	29.0	31.0	45.0	41.0	55.0
1.4	1.3	59.0	32.0	62.0	31.0	32.0	92.0	60.0	28.0	26.0	79.0	76.0	42.0	80.0	16.0
1.5	1.4	1.3	51.0	28.0	21.0	79.0	98.0	43.0	64.0	90.0	93.0	29.0	46.0	14.0	24.0
1.6	19.0	68.0	66.0	79.0	57.0	75.0	25.0	76.0	85.0	84.0	75.0	90.0	45.0	57.0	84.0

(b) Positions recorded in the dark-matter

Figure 6. The 16 x 16 space matrix with the matter and the dark-matter

Actions at different positions are defined as shown in Figure 7 (a). There are five actions are defined. These are "Non", "Left", "Right", "Up" and "Down", representing "no-action required", "to move left", "to move right", "to move up" and "to move down", respectively. Action index values are also defined as shown in Figure 7 (a). The values 0.0, 0.1, 0.2, 0.3 and 0.4 are defined as the action index values of "Non", "Left", "Right", "Up" and "Down", respectively. The action index value of the agent at each position on the path is shown in Figure 7 (b). For example, the required action at the start position (2, 1) is "to move right", therefore, its index value 0.2 is written at (2, 1) as shown in Figure 7 (b).



(a) Actions and action index of the agent

(b) Actions of the agent at each position

Figure 7. Actions and action index of the agent

Defining the action vector as a vector **Y** by using the action index value as shown in Figure 7 (b), an energy vector **E** can be calculated using equation (2) described in Section 2.

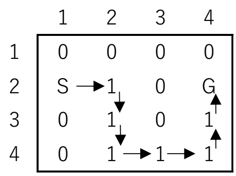
The action of the agent at each position can be calculated by using the space matrix **X** and energy vector **E** by using the equation (3). In order to calculate the action index value, a retrieval mechanism is required. In the retrieval mechanism, the index value of the agent position is used to retrieve the relative row from the space matrix **X.** The retrieval result is a row vector of matrix X corresponding to the position of the agent. Therefore, it is required that the agent must has a position sensor and the accuracy of the sensor output value is sufficient to meet the requirements of the retrieval mechanism.

If the retrieval mechanism is used, it can be used directly to retrieve the action index value based on the position index value. A correspondence table or a retrieval function can be used to implement the retrieval calculation. Table 1 is an example which can be used to implement the retrieval. It has two columns, one is the position index value column and the other is action index value column. Using the table, for given position, an action index value can be found directly from the table. That is, the action of the agent can be calculated without the dark-matter matrix.

Table 1. A table used to output the action index correlated to the position index

Position Index	Action Index
0.0	0.0
0.1	0.2
0.2	0.0
0.3	0.0
0.4	0.0
0.5	0.4
0.6	0.4
0.7	0.2
0.9	0.0
0.9	0.0
1.1	0.2
1.2	0.0
1.3	0.0
1.4	0.3
1.5	0.3
1.6	0.0

However, it is not always sufficient to calculate the action index value without the dark-matter. If the agent does not have the position sensor, but it has a laser sensor which output shows which direction that the agent can move as shown in Figure 8. The output values of the sensor are defined as follows: As shown in Figure 8 (b), the agent can move in four directions, moving to the "Down", "Left" and "Right" direction, the outputs of the sensor are defined "0100", "0010" and "0001", respectively. The direction, in which the agent can move at each position, is shown in Figure 8 (a). The sensor's output values when the agent is in different positions are shown in Figure 8 (c). For example, at the position (3, 2), the agent can move to "Up" direction, therefore, the output value of the sensor is "1000". The agent can go back from the current position to its previous position. If the agent is at the point (2, 2), the agent can move to two different positions. The agent can move back to the start position (2, 1) when it move "Left". It can also move "Down" to the next position (3, 2). The output value of the sensor at the position (2, 2) is "0110", which is the summary of the two direction "0010", "Left" and "0100", "Down".



(a) The maze matrix

Up	Down	Left	Right
1000	0100	0010	0001

(b) The output of the laser sensor

	1	2	3	4
1	0	0	0	0
2	0001	0110	0	0100
3	0	1100	0	1100
4	0	1001	0011	1010

(c) The output of the laser sensor in the maze

Figure 8. The output of the laser sensor of the agent

As shown in Figure 8 (c), when the agent is at the position (3, 2) and (3, 4), the output values of the sensor are the same "1100". That is, the agent can move "Up" or move "Down". If a table like Table 1 is used to retrieve the action of the agent, two different actions, "Moving Down" and "Moving Up" will be the output values. That is, if the retrieval values for a giving a sensor data correspond to different actions, without dark-matter， it is impossible to find a unique action for the agent. Therefore, it is necessary to use dark-matter in the calculation for deciding which action should be taken.

It is not always possible to know the path at the start position as that shown in Figure 6 (b). For example, when reinforcement learning is applied to find the path from the start position to the goal position, the path is unknown at the start position. The path will be found after many trys are performed. Sometimes, many paths are found. Rewards are assigned to found paths. If the length of a path is shorter than the others, higher reward than the others is assigned to the path. During the reinforcement learning, the agent is trained to obtain the maximum reward. In this way, the optimal path from the start position to the goal position can be found. At the same time, actions of the agent at the positions on the path are determined.

When the path is unknown at the start position, it is impossible to create the space matrix as that shown in Figure 8 (a). Here, ***a new method is proposed for generating the space matrix***. At the same time, ***a new method for knowledge expression is also proposed***. In the methods, passed positions are recorded instead of the next positions. For example, when the agent moves from the start position (2, 1) to the position (2, 2), the passed position (2, 1) is recorded. That is, dark-matter matrix can be created based on events that have occurred instead of the events which will occurred in the future. In the following, an example is used to illustrate the method in detail. In the example, the agent has a laser sensor.

In the example, a matrix is used to show a maze as shown in Figure 9 (a). The agent can be in any positions where the “1” is marked in the matrix. The agent cannot be in the positions where “0” is marked in the matrix. Figure 9 (b) shows the output of the sensor of the agent in each position, and Figure 9 (c) shows the index values of the sensor outputs at each position. For example, when the agent is at the position (2, 2), as the agent can move to “Up”, “Down” and “Left”, output of the sensor is “1110”, which is the summary of the value “Up”, “1000”, “Down”, “0100” and “Left”, “0010”. As the agent can not be in the position (3, 1), (4, 1), (2, 3), (3, 3) and (1, 4), the output of the sensor at those positions is marked with an “X”.

	1	2	3	4
1	1	1	1	0
2	1	1	0	1
3	0	1	0	1
4	0	1	1	1

(a) The maze matrix

	1	2	3	4
1	0101	0111	0111	X
2	1001	1110	X	0100
3	X	1100	X	1100
4	X	1001	0011	1010

(b) The output of the laser sensor

	1	2	3	4
1	0.5	0.7	0.7	X
2	0.9	1.4	X	0.4
3	X	1.2	X	1.2
4	X	0.9	0.3	1.0

(c) The index of the laser sensor

Figure 9. The output of the laser sensor of the agent

The reinforcement learning is performed as the follows. At the position (2, 1), the agent can move to two different directions “Up” and “Right”. The probability value 0.5 is assigned to the two directions. That is, the agent has a 50% chance of moving “Up” and a 50% chance of moving to the “Right”. When the agent moves to the position (2, 2), it has three moving directions, “Up”, “Down” and “Left”. Therefore, 0.33 is assigned as the probability value to the three directions. The maze is set up as follows. The target position is set at position (2, 4), and the starting position can be any position marked “1", as shown in Figure 9 (a). When the agent moves from a position to its neighbor position, it is said to have "moved a step". The number of the steps, which the agent is used to move from the start position to the goal position, is used for the reward calculation. The fewer steps the agent used, the higher the score the agent got. If the start position is at the point (2, 1), the minimum number of steps required is seven for the agent moving from the start position the goal position. When the agent moves to a direction where higher reward were obtained than the other directions, higher probability value is assigned to the direction. For example, at the position (2, 1), as moving to the “Right” can obtain higher reward than moving “Up”, 0.01 is added to the probability value of moving to that “Right”. Experiments were performed to find the path from the start position to the goal position. Based on the experimental results, trying about 200 times, the best path from the start position (2, 1) to the goal position (2, 4) can be found by the agent. After 2000 times trying, the probability values to the high reward directions are increased to 0.99.

In order to record sensor values, ***a working memory mechanism is proposed***. In the mechanism, a vector is created which initial value is randomly assigned. The length of the vector is as same as the number of the steps required by the agent from the start position to the goal position. For example, when the agent started at position (2, 1) which is set as the start position, and the agent moved through positions, (2, 2), (3, 2), (4, 2), (4, 3), (4, 4), (3, 4), and reached the goal position (2, 4), there are seven steps are required. Thus, the vector is defined with seven elements with random values. Then, the value of the current sensor value is combined with the vector, thus a vector with eight elements is created, as shown in Figure 10. In Figure 10, index values of the sensor values are used. The background of the element recording the current sensor value is painted white, and the background of the elements recording the past sensor values are painted gray. For example, at the start position (2, 1), the current sensor value is 0.9, which was recorded at the first element of the vector, and the background of the first element was painted white, as shown in Figure 10 (a). The background of all the other elements were painted gray. The values of the elements from the second to the eighth were remained the random values because no sensor data were recorded. When the agent moved to position (2, 2), 1.4 which was the index value of the sensor at the position, was recorded to the first element and the previously recorded data 0.9 was moved to the second element as shown in Figure 10 (b). In the same way, when the agent moved to position (3, 2), 1.2, which was the index value of the sensor output, was recorded to the first element, and the previously recorded data were moved to the second and third elements, respectively, as shown in Figure 10 (c). After the agent moved to the goal position (2, 4), all index values of the sensor output were recorded as shown in Figure 10 (d). The index value of the sensor output at the goal position was recorded to the first element, and the index value of the sensor at the start position was recorded to the eighth element.

1	2	3	4	5	6	7	8
0.9	**94.0**	**84.0**	**12.0**	**78.0**	**58.0**	**84.0**	**23.0**

(a) At position (2,1)

1	2	3	4	5	6	7	8
1.4	**0.9**	**84.0**	**12.0**	**78.0**	**58.0**	**84.0**	**23.0**

(b) At position (2,2)

1	2	3	4	5	6	7	8
1.2	**1.4**	**0.9**	**12.0**	**78.0**	**58.0**	**84.0**	**23.0**

(c) At position (3,2)

1	2	3	4	5	6	7	8
0.4	**1.2**	**1.0**	**0.3**	**0.9**	**1.2**	**1.4**	**0.9**

(d) At position (2,4)

Figure 10. Working-memory and recorded index value of the sensor output

Actions of the agent are also recorded by action index value. For example, if the agent moved to the "Right" at the start position, the index value 0.1, which is the index value of moving to the "Right", was recorded. A vector records all the action index values from the start position to the end position is used as the **Y** vector, as shown in Figure 11 (c).

As shown in Figure 11 (a), the space **X** is a collection of the working-memory of the agent moved from the start position to the goal position. Its inverse matrix $\mathbf{X}^{-1}$ is shown in Figure 11 (b). By multiplying $\mathbf{X}^{-1}$ by **Y**, an energy vector **E** is defined.

$$\begin{pmatrix} 0.3 & 0.9 & 1.2 & 1.4 & 0.9 & 58.0 & 84.0 & 23.0 \\ 0.4 & 1.2 & 1.0 & 0.3 & 0.9 & 1.2 & 1.4 & 0.9 \\ 0.9 & 94.0 & 84.0 & 12.0 & 78.0 & 58.0 & 84.0 & 23.0 \\ 0.9 & 1.2 & 1.4 & 0.9 & 78.0 & 58.0 & 84.0 & 23.0 \\ 1.0 & 0.3 & 0.9 & 1.2 & 1.4 & 0.9 & 84.0 & 23.0 \\ 1.2 & 1.4 & 0.9 & 12.0 & 78.0 & 58.0 & 84.0 & 23.0 \\ 1.2 & 1.0 & 0.3 & 0.9 & 1.2 & 1.4 & 0.9 & 23.0 \\ 1.4 & 0.9 & 84.0 & 12.0 & 78.0 & 58.0 & 84.0 & 23.0 \end{pmatrix} \quad \begin{pmatrix} -0.03 & 2.77 & -0.03 & 0.03 & 0.01 & -0.03 & -0.06 & 0.00 \\ 0.00 & 0.01 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & -0.01 \\ 0.00 & -0.01 & 0.00 & 0.00 & 0.00 & -0.01 & 0.00 & 0.01 \\ 0.00 & -0.08 & 0.00 & -0.09 & 0.00 & 0.09 & 0.00 & 0.00 \\ -0.01 & -0.02 & 0.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.02 & 0.03 & 0.00 & 0.00 & -0.02 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.01 & 0.00 & 0.00 & 0.01 & 0.00 & -0.01 & 0.00 \\ 0.00 & -0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 & 0.00 \end{pmatrix} \quad \begin{pmatrix} 0.1 \\ 0.0 \\ 0.1 \\ 0.1 \\ 0.8 \\ 0.4 \\ 0.8 \\ 0.4 \end{pmatrix} \quad \begin{pmatrix} -0.06 \\ 0.00 \\ 0.00 \\ 0.03 \\ 0.00 \\ -0.01 \\ 0.00 \\ 0.04 \end{pmatrix}$$

(a) X **(b) $\mathbf{X}^{-1}$** **(c) Y** **(d) E**

Figure 11. Working-memory and recorded index value of the sensor output

When the energy vector **E** is defined, each action of the agent at each relative position can be calculated by multiply the vector of the working-memory by **E**, as expressed by equation (3) introduced in Section2. The calculation result of the energy vector is shown in Figure 11 (d). When the dark matter matrix is utilized, a unique action index value can be calculated even if the output values of the sensor are the same. For example, at the position (3, 2) and (3, 4), the index values of the sensor output value are the same 1.2, as shown in Figure 12 (a) and (b), where the index values are recorded at the first elements of the two vectors. In the working-memory, the values of the dark-matter values in the second to the eight elements of the two vectors are different, as shown in Figure 12 (a) and (b).

1	2	3	4	5	6	7	8
1.2	1.4	0.9	12.0	78.0	58.0	84.0	23.0

(a) Values of the working-memory at position (3,2)

1	2	3	4	5	6	7	8
1.2	1.0	0.3	0.9	1.2	1.4	0.9	23.0

(b) Values of the working-memory at position (3,4)

Figure 12. Working-memory and recorded index value of the sensor output

By multiplying the two vectors of the working-memory by the energy vector **E**, two different action index values 0.4 and 0.8 are obtained, as shown in equation (4) and (5). The index value 0.4 means that the agent should take the action "Moving Down" at the position (3, 2), and the index value 0.8 means that the agent should take the action "Moving Up" at the position (3, 4).

$$[1.2, 1.4, 0.9, 12.0, 78.0, 58.0, 84.0, 23.0] \times \begin{bmatrix} -0.06 \\ 0.00 \\ 0.00 \\ 0.03 \\ 0.00 \\ -0.01 \\ 0.00 \\ 0.04 \end{bmatrix} = 0.4 \tag{4}$$

$$[1.2, 1.0, 0.3, 0.9, 1.2, 1.4, 0.9, 23.0] \times \begin{bmatrix} -0.06 \\ 0.00 \\ 0.00 \\ 0.03 \\ 0.00 \\ -0.01 \\ 0.00 \\ 0.04 \end{bmatrix} = 0.8 \tag{5}$$

During the above process, the agent obtained skills to take appropriate actions at different positions leading it to move from the start position to the goal positions. That is, skills are acquired through experience of the agent. Considering the definition of *knowledge*, which is defined as "facts, information, and skills acquired through experience or education", it can be found that ***knowledge is expressed in the dark-matter matrix*** of the space matrix **X**. ***The energy vector* E *is used to change knowledge into actions***.

In summary, the ***learning process*** of the agent can be divided into five steps:

1. Finding a path from the start position to the goal position,

2. Recording the sensor output value in the working memory during the movement of the agent from the starting position to the goal position,
3. Creating the space matrix **X** based on the working-memory,
4. Calculating the inverse matrix of the matrix **X**,
5. Calculating the energy vector **E**.

4. Discussion

In EJC2022 [23], the concept of "dark-matter", "energy" and "action" presented in this paper are discussed. Professor Marie Duží, an expert in natural language processing and artificial intelligence [24], questioned about the "energy" and "actions" while considering the famous concept of "mass energy equivalent" in physics $E = mc^2$. The question is very helpful to clarify the concepts presented in this paper. Let's review the "matter", "action" and the "energy" equation, equation (1) presented in Section 2,

$$\mathbf{Y} = \mathbf{X}*\mathbf{E} \qquad (1)$$

Equation (1) can be found to resemble the "mass energy equivalent" equation, if "Y" and "E" in equation (1) are rewritten to "E" and "C^2", as in equation (6).

$$\mathbf{E} = \mathbf{X}*\mathbf{C}^2 \qquad (6)$$

In the equation "mass and energy equivalent," "**mass**" is a measure of the amount of matter that an object contains. That is, it is visible, or it can be sensed by sensors. As explained in Figure 1, "**matter**" is defined as an index matrix that correlates with sensor data. That is, "**matter**" can be considered the same as "**mass**." It can also be seen that the concept of energy in physics is the "action" that we introduced in Section 2. In equation (1), the values of the elements of the matrix **Y** and the mass in the matrix **X** are measurable. That is, the measurable values of **Y** are the energy in physics. In other words, the concept of energy in physics is the "action" that we referred in this article, and the concept of mass in physics is the "matter" that we referred to.

Let's do a transformation of equation (6), and we can get equation (7).

$$\mathbf{X}*\mathbf{C}^2 = \mathbf{E}$$
$$\mathbf{X}^{-1}*\mathbf{X}*\mathbf{C}^2 = \mathbf{X}^{-1}*\mathbf{E}$$

$$\mathbf{C}^2 = \mathbf{X}^{-1}*\mathbf{E} \qquad (7)$$

Next, we will take an example to analyze the relationship between equation (6) and the "energy mass equivalent" equation. Suppose that space X is a five-by-five matrix, as shown in Figure 13 (a). This assumption means that there are five different types of matter in space. The elements in the first column of the matrix are the mass of matter. In the example, the values of the "**matter**" and "**dark-matter**", which are the elements of the matrix **X**, are shown in Figure 13 (a). The matrix **E** in equation (6) is set to a vector whose elements are shown in Figure 13 (b). The inverse matrix of **X**, $\mathbf{X}^{-1}$ is show in Figure 13 (c).

$$\underbrace{\begin{pmatrix} 6.00 & 14.16 & 72.84 & 49.42 & 26.72 \\ 3.00 & 83.37 & 75.00 & 36.21 & 28.21 \\ 9.00 & 27.55 & 32.70 & 59.96 & 20.56 \\ 2.00 & 28.83 & 34.27 & 35.39 & 74.98 \\ 7.00 & 55.43 & 86.68 & 38.21 & 87.91 \end{pmatrix}}_{\text{mass} \quad \text{dark-matter}} \qquad \underbrace{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}}_{\mathbf{E}} \qquad \underbrace{\begin{pmatrix} -0.09 & -0.07 & 0.11 & -0.15 & 0.16 \\ -0.02 & 0.01 & 0.01 & 0.00 & 0.00 \\ 0.02 & 0.00 & -0.01 & -0.01 & 0.00 \\ 0.01 & 0.01 & 0.01 & 0.02 & -0.03 \\ -0.01 & -0.01 & 0.00 & 0.01 & 0.01 \end{pmatrix}}_{\mathbf{X}^{-1}}$$

(a) Space X: created with five types of matter **(b) Energy vector** **(c) Inverse matrix of X**

Figure 13. An example of a space with five different types of matter

The values of the element of vector $\mathbf{C}^2$ can be calculated based on equation (7), by multiplying $\mathbf{X}^{-1}$ by $\mathbf{E}$, as shown in Figure 14.

$$\underbrace{\begin{pmatrix} 100.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}}_{\mathbf{C}^2} = \underbrace{\begin{pmatrix} -0.09 & -0.07 & 0.11 & -0.15 & 0.16 \\ -0.02 & 0.01 & 0.01 & 0.00 & 0.00 \\ 0.02 & 0.00 & -0.01 & -0.01 & 0.00 \\ 0.01 & 0.01 & 0.01 & 0.02 & -0.03 \\ -0.01 & -0.01 & 0.00 & 0.01 & 0.01 \end{pmatrix}}_{\mathbf{X}^{-1}} * \underbrace{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}}_{\mathbf{E}}$$

Figure 14. The calculated result of $\mathbf{C}^2$ with the values of Figure 13

As shown in Figure 14, only the first element of vector $\mathbf{C}^2$ is a non-zero value, and all the other values of the vector are zero. When the vector $\mathbf{C}^2$ is given, vector $\mathbf{E}$ can be calculated based on equation (6), as shown in Figure 15 (a). Let c^2 represent the first value of $\mathbf{C}^2$, the vector $\mathbf{E}$ can be calculated by multiplying the first column of $\mathbf{X}$ by c^2, as shown in Figure 15 (b).

$$\underbrace{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}}_{\mathbf{E}} = \underbrace{\begin{pmatrix} 6.00 & 14.16 & 72.84 & 49.42 & 26.72 \\ 3.00 & 83.37 & 75.00 & 36.21 & 28.21 \\ 9.00 & 27.55 & 32.70 & 59.96 & 20.56 \\ 2.00 & 28.83 & 34.27 & 35.39 & 74.98 \\ 7.00 & 55.43 & 86.68 & 38.21 & 87.91 \end{pmatrix}}_{\text{mass} \quad \text{dark-matter}} * \underbrace{\begin{pmatrix} 100.00 \\ 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}}_{\mathbf{C}^2} \qquad \underbrace{\begin{pmatrix} 600.00 \\ 300.00 \\ 900.00 \\ 200.00 \\ 700.00 \end{pmatrix}}_{\mathbf{E}} = \underbrace{\begin{pmatrix} 6.00 \\ 3.00 \\ 9.00 \\ 2.00 \\ 7.00 \end{pmatrix}}_{\mathbf{mass}} * \underbrace{100.00}_{c^2}$$

(a) Calculating E with vector $\mathbf{C}^2$ **(b) Calculating E with the scalar value c^2**

Figure 15. The calculated result of $\mathbf{E}$ with the vector $\mathbf{C}^2$ and the scalar value c^2

In this example, if E is the i-th element of the vector **E** and m is also the i-th element of the vector **mass**, it can be found that $E = m\mathrm{c}^2$. For example, for the first element of E and the first element of mass,

$E = 600$;
$m = 6$;
$\mathrm{c}^2 = 100$, where, $\mathrm{c} = 10$;
$600 = 6 * 10^2$.

That is,

$E = m\mathrm{c}^2$.

In summary, if the vector $\mathbf{C}^2$ is defined such that the value of its first element is a non-zero value c^2 and all the other values are zero, as shown in Figure 15, the elements values of vector **E** can be calculated by multiplying the vector **mass** elements by the scalar value c^2. That is, $E = m\mathrm{c}^2$. The vector $\mathbf{C}^2$ is a special vector indicating that "dark-matter" is not used in the calculation. That is, the equation, $E = m\mathrm{c}^2$, can only be used if "knowledge" or "state transition" is not considered during the computation process.

5. Conclusion and future work

In this paper, the concept of "***dark-matter***" is reviewed and further developed to express experience and knowledge. The concept of "***space***" is also reviewed and developed into a matrix that can be created based on experience. In addition, the characteristic of the concept of "***energy***" is revealed, indicating that it can be used to transform knowledge into the desired calculation. Based on the concept of dark-matter, examples of an agent moving in mazes are used for the exploratory research on knowledge expression and its learning process. In the paper, a working memory mechanism is proposed for recording the agent's experience in the actions of the agent's movement from the start position to the goal position in the maze. A new method is proposed for creating the space matrix **X** by using the working-memory. During the exploratory research of the paper, the process of acquiring knowledge is revealed. The most important contribution of this paper is that new methods are proposed for acquiring and expressing knowledge based on the concept of dark-matter. In the paper, it is revealed that knowledge can be expressed in a matrix and energy vectors can be used to turn knowledge into actions of the agent. In the paper, the learning process is illustrated during the agent's movements from a starting position to a goal position. Based on the illustration, the learning mechanism for the agent to acquire knowledge is revealed. As our future work, more features of the proposed methods and the mechanism will be further investigated, and application systems based on the proposed methods and the mechanism will be developed.

References

[1] Chen, X. and Kiyoki, Y., "*On Semantic Spatiotemporal Space and Knowledge with the Concept of "Dark-Matter"*," Information Modelling and Knowledge Bases XXXIII, IOS Press, pp.110-128, 2021.

[2] Chen, X. and Kiyoki, Y., "*A query-meaning recognition method with a learning mechanism for document information retrieval*," Information Modelling and Knowledge Bases XV, IOS Press, Vol. 105, pp.37-54, 2004.

[3] Chen, X. and Kiyoki, Y., "*A dynamic retrieval space creation method for semantic information retrieval*," Information Modelling and Knowledge Bases XVI, IOS Press, Vol. 121, pp.46-63, 2005.
[4] Kiyoki, Y. and Kitagawa, T., "*A semantic associative search method for knowledge acquisition*," Information Modelling and Knowledge Bases, IOS Press, Vol. VI, pp.121-130, 1995.
[5] Kitagawa, T. and Kiyoki, Y., "*A mathematical model of meaning and its application to multidatabase systems*," Proc. 3rd IEEE International Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems, pp.130-135, April 1993.
[6] Chen, X., Kiyoki, Y. and Kitagawa, T., "*A multi-language oriented intelligent information retrieval system utilizing a semantic associative search method,*" Proceedings of the 17th IASTED International Conference on Applied Informatics, pp.135-140, 1999.
[7] Chen, X., Kiyoki, Y. and Kitagawa, T., "*A semantic metadata-translation method for multilingual cross-language information retrieval,*" Information Modelling and Knowledge Bases XII, IOS Press, Vol. 67, pp.299-315, 2001.
[8] Kiyoki, Y., Kitagawa, T. and Hitomi, Y., "*A fundamental framework for realizing semantic interoperability in a multidatabase environment*, " International Journal of Integrated Computer-Aided Engineering, Vol.2, No.1(Special Issue on Multidatabase and Interoperable Systems), pp.3-20, John Wiley & Sons, Jan. 1995.
[9] Kiyoki, Y., Kitagawa, T. and Hayama, T., "*A metadatabase system for semantic image search by a mathematical model of meaning*, " ACM SIGMOD Record, Vol.23, No. 4, pp.34-41, Dec. 1994.
[10] Kiyoki, Y, Chen, X. and Kitagawa, T., "*A WWW Intelligent Information Retrieval System Utilizing a Semantic Associative Search Method*," APWeb'98, 1st Asia Pacific Web Conference on Web Technologies and Applications, pp. 93-102, 1998.
[11] Ijichi, A. and Kiyoki, Y.: "*A Kansei metadata generation method for music data dealing with dramatic interpretation*," Information Modelling and Knowledge Bases, Vol.XVI, IOS Press, pp. 170-182, May, 2005.
[12] Kiyoki, Y., Chen, X. and Ohashi, H.: "*A semantic spectrum analyzer for realizing semantic learning in a semantic associative search space*," Information Modelling and Knowledge Bases, Vol.XVII, IOS Press, pp.50-67, May 2006.
[13] Takano, K. and Kiyoki, Y.: "*A causality computation retrieval method with context dependent dynamics and causal-route search functions*," Information Modelling and Knowledge Bases, ISO Press, Vol.XVIII, pp.186-205, May 2007.
[14] Chen, X. and Kiyoki, Y.: "*A visual and semantic image retrieval method based on similarity computing with query-context recognition*," Information Modelling and Knowledge Bases, IOS Press, Vol.XVIII, pp.245-252, May 2007.
[15] Nitta T, "*Resolution of singularities introduced by hierarchical structure in deep neural networks*," IEEE Trans Neural Netw Learn Syst., Vol.28, No.10, pp.2282-2293Oct. 2017.
[16] Wiatowski, T. and Bölcskei, H., "*A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction*," IEEE Transactions on Information Theory, PP(99) · Dec. 2015.
[17] Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J. "*Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*," In Kremer, S. C. and Kolen, J. F. (eds.), *A Field Guide to Dynamical Recurrent Neural Networks*, IEEE Press, 2001.
[18] Hochreiter, S. and Schmidhuber, J., "*Long short-term memory*,". Neural computation, Vol.9, No.8, pp.1735-1780, 1997.
[19] Kalchbrenner, N., Danihelka, I. and Graves, "*A. Grid long short-term memory*," CoRR, abs/1507.01526, 2015.
[20] Chen, X. and Kiyoki, Y., "*On Logic Calculation with Semantic Space and Machine Learning*," Information Modelling and Knowledge Bases XXXI, IOS Press, Vol. 321, pp.324-343, 2019.
[21] Chen, X. Prayongrat, M. and Kiyoki, Y., "*A Concept for Control and Program Based on the Semantic Space Model*," Information Modelling and Knowledge Bases XXXII, IOS Press, Vol. 333, pp. 26-44, 2020.
[22] Oxford English dictionary, "knowledge", https://www.lexico.com/definition/knowledge, accessed on Jan. 7, 2022.
[23] 32nd International Conference on Information Modelling and Knowledge Bases EJC 2022, Hamburg, Germany, May 30 – June 3, 2022.
[24] Marie Duží, "*Concepts, Language and Ontologies (from a Logical Point of View)*," Information Modelling and Knowledge Bases XV, IOS Press, pp.193-209, 2004.