Organized memory for natural text understanding and its meaning visualization by machine

Kandhasamy Sadhasivam Jayakumar

2007

Kandhasamy, S. J. (2007). Organized memory for natural text understanding and its meaning visualization by machine. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/6560>

<https://doi.org/10.32657/10356/6560>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:08:43 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**ORGANIZED MEMORY FOR NATURAL TEXT
UNDERSTANDING AND ITS MEANING
VISUALIZATION BY MACHINE**

**KANDHASAMY SADHASIVAM JAYAKUMAR
SCHOOL OF MECHANICAL & AEROSPACE ENGINEERING
2007**

Organized Memory for Natural Text Understanding And Its Meaning Visualization By Machine

Kandhasamy Sadhasivam Jayakumar

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

Abstract

This thesis presents the design and implementation of a mind or mental architecture for a machine to autonomously learn, analyze and synthesize natural language. The machine mind consists of mental processes and organized memory, and interaction between them creates the phenomenon of text understanding. The organized memory is the key component for the text understanding and it is a machine's brain memory, in which modeling of physical world and conceptual world elements are represented.

The implemented system is named KnowNet, and it learns and understands spatial language. A new direction is taken to model the spatial language in a deterministic way using entities properties and their interactions. The spatial words such as on, in, over, under, above, below, inside, outside, behind, in front of, back, front, left, right, beside, between, among, near and far are modeled and learned. The spatial concepts are learned through modeling the entities and their interactions. In contrast, current works use collocation information of words to learn the spatial concepts. Initially, sixty entities are learned by KnowNet. Only twenty five entities were needed to learn the above spatial concepts. The rigid body statics model takes the entities and constraints sequence as the input and then constructs the configuration network. It is also used to determine and verify the meaning of the sentences. The sentence generation experiments and different types of

sentences understanding experiments have been performed to evaluate the system's performance.

Acknowledgement

First of all, I would like to express my utmost appreciation and thanks to my supervisor, Dr. Xie Ming, Associate professor, Nanyang Technological University, for his invaluable guidance throughout the progress of this research work. Not only his vast knowledge of the subject, but also his dedication to research have been an inspiration to me.

I would also like to thank all my friends and staff at Robotics Research Center (RRC), for their sincere assistance and making my stay here so much more enjoyable.

Finally, I would like to express my gratitude to my parents for their support and concern for me.

Contents

Abstract	I
Acknowledgement	III
Contents	IV
List of Figures	VIII
List of Tables	XIII
1 Introduction	1
1.1 Significance	1
1.2 Applications	2
1.2.1 Conversational Dialogue Between a Humanoid Robot and a Human	2
1.2.2 Spatial Language to 3D Scene Conversion	3
1.3 Challenges	4
1.3.1 Learning Capability	4
1.3.2 Thinking Capability	5
1.3.3 Seeing Capability	5
1.3.4 Reading Capability	5
1.3.5 Listening Capability	5
1.3.6 Autonomous Capability	6
1.4 Organization of Thesis	6
2 Problem Statement	8
2.1 Literature Review	8
2.1.1 Intelligence	9
2.1.2 Meaning of a Word	11
2.1.3 Meaning Representation in Mind	12

CONTENTS	V
2.1.4 Symbol Grounding	13
2.1.5 Symbol Grounding Experiments	14
2.1.6 Knowledge-Based Natural Language Understanding	15
2.1.7 Spatial Language Understanding	18
2.2 Issues Related to Existing Research Works	19
2.2.1 Issues Related to Knowledge-Based Natural Language Un- derstanding	19
2.2.2 Issues Related to Symbol Grounding	20
2.3 Proposed Methodology	21
2.3.1 Machine Mental Architecture	21
2.3.2 Proposed Method for Spatial Language Learning, Under- standing and Generation	25
2.4 Objective and Scope	28
2.4.1 Scope of the Work	28
2.5 Contributions	29
2.6 Summary	29
3 Modeling of Elementary Meanings	31
3.1 Real World	31
3.1.1 Physical World	32
3.1.2 Conceptual World	33
3.1.3 Meaning and Knowledge	35
3.2 Physical Models	35
3.2.1 Properties and Constraints	36
3.2.2 Modeling of Mechanical Constraint for Rigid Bodies	38
3.3 Conceptual Models	42
3.3.1 Properties and Constraints	42
3.4 Summary	43
4 Construction of Organized Physical Meanings	45
4.1 Background	45
4.2 Physical Constraint Networks	46
4.3 Configuration Network	47
4.3.1 Rigid Body Statics Algorithm	48
4.3.2 Octree	50
4.3.3 Intersection Testing	51

CONTENTS	VI
4.3.4 Geometry Parameters Calculation	57
4.3.5 Transformation Matrix	58
4.3.6 Test for Stability	60
4.4 Scene Synthesis	61
4.4.1 Algorithm	62
4.5 Summary	62
5 Construction of Organized Conceptual Meanings	63
5.1 Conceptual Constraint Networks	63
5.2 Constraints for Phrase Formation	64
5.2.1 Algorithm for Phrase Construction	65
5.2.2 Phrase Construction Example	66
5.3 Constraints for Sentence Formation	67
5.4 Construction of Only Grammatically Correct Concept Network from Conceptual Models	67
5.4.1 Algorithm	68
5.4.2 Example for Concept Network Construction	69
5.5 Construction of Meaningful Concept Network from Configuration Network and Conceptual Models	70
5.5.1 Algorithm	71
5.5.2 Example	72
5.6 Summary	73
6 Implementation, Results and Discussion	74
6.1 KnowNet Architecture	75
6.2 KnowNet Graphical User Interface (GUI)	75
6.2.1 Programmed Interface	76
6.3 Learning of Entities and Their Associated Words	77
6.3.1 Example of Learning an Entity - Chesspawn	78
6.4 Learning of Entity's Usage Configuration	79
6.4.1 Graphical User Interface for Learning the Usage Configuration	80
6.5 Learning of Entities Static Interactions and Their Associated Concepts	81
6.5.1 Modeling Parameters for Spatial Constraints	86
6.5.2 Modeling and Learning of Spatial Constraints and Their As- sociated Concepts	88
6.6 Test on Predicting the Spatial Concept	105

CONTENTS	VII
6.7 Generation of Sentence(s) from Set of Entities	108
6.7.1 Sentence Generation Experiments	109
6.8 Understanding of Sentences	115
6.8.1 Grammatical Analysis of the Sentences	115
6.8.2 Understanding of Compound Sentences	117
6.8.3 Understanding of the Sentences Having Complex Preposi- tional Phrases	120
6.8.4 Understanding of the Sentences that Violate the Truthfulness of the Physical World	123
6.9 Discussion	126
7 Conclusions, Contributions and Future Work	128
7.1 Conclusions	128
7.2 Contributions	130
7.3 Future Work	131
Bibliography	133

List of Figures

1.1	The generic application scenario of conversational dialogue between a humanoid robot and a human.	2
1.2	The generation of 3D scene from the sentences ‘ <i>The bed is on the ground. The chair is beside the bed. The table lamp is on the chair. The table fan is behind the bed. The chair is on the ground</i> ’.	4
2.1	A machine’s mental architecture. The interaction between mental processes and organized memory creates the phenomena of text understanding	22
2.2	Architecture of an organized memory, in which modeling of meanings of the physical and conceptual world are represented. Organized memory acquires the meaning (i.e. properties and constraints) through learning process. More specifically, the properties are acquired through sensors and the constraints are learned by modeling the interactions among the entities.	23
2.3	Entities encoding and decoding processes. Entities are encoded or mapped to properties through the sensors or programmed interface. The properties can be decoded or mapped to virtual entities and their possible interactions with other entities are achieved through the constraint physics models such as computer graphics, rigid body statics and dynamics, etc.	25
2.4	Learning of words through entities properties and their interactions	26
2.5	Sentence generation using physical and conceptual models	27
2.6	Method for understanding the sentence	28
3.1	An illustration of the duality of the real world, in which the physical and conceptual world coexist. The properties and logics behind the physical and conceptual worlds form the body of knowledge	33

3.2	The properties and constraints of an entity form the basis for elementary meanings. They can be acquired by the machine through its sensory systems or programmed interface by realtime interaction with the environment.	37
3.3	Elementary meanings of conceptual symbols	42
4.1	Shows the octree decomposition for the ‘child’ object for different vertices count and octree depth	52
4.2	Shows the oriented boxes and formation of a separating axis in two dimensions	53
4.3	Illustration of rotation matrix and direction vector calculation . . .	59
5.1	Phrase formation constraints for English. The ordered sequence of lexical properties forms the constraints for the phrase.	65
5.2	Phrase construction for a set of symbols ‘cat’, ‘the’, ‘good’. Constraint matching plays an important role in phrase formation	66
5.3	Shows constraints for sentence formation.	68
5.4	Shows the processes in construction of the concept network ‘I want more cakes some’ from words such as ‘more’, ‘cakes’, ‘I’, ‘some’, ‘want’.	70
5.5	Illustration of concept network construction from configuration network, which helps to determine the subject, verb, object, and modifiers.	72
6.1	KnowNet architecture, in which physical and conceptual models are learned through programmed interface by real time interaction with the human master. Physical models are combined and organized by the rigid body statics model in order to construct the configuration network, from which scene and text are synthesized	76
6.2	KnowNet Graphical User Interface (GUI), which contains display pane, programmed interface, understanding pane and synthesized text display pane. Programmed interface is used to acquire the properties and constraints of entities and words.	77
6.3	Learning of an entity and its associated word	78
6.4	Graphical user interface for learning the usage configuration of the entity	80
6.5	Different stable configurations of the entity ‘mug’. Configuration number three is considered as the usage configuration.	81

LIST OF FIGURES	X
6.6 Usage configuration of different entities.	82
6.7 Usage configuration of different entities.	83
6.8 Usage configuration of different entities.	84
6.9 Usage configuration of different entities.	85
6.10 The graphical user interface for learning the spatial concept	90
6.11 Spatial configuration of spatial constraint-II for ‘On’.	91
6.12 Spatial configuration of spatial constraint-III for ‘On’.	92
6.13 Spatial configurations of spatial constraint-I for ‘In’.	92
6.14 Spatial configuration of spatial constraint-I for ‘Over’.	93
6.15 Spatial configuration of spatial constraint-II for ‘Over’.	94
6.16 Spatial configurations of spatial constraint-I for ‘Under’.	95
6.17 Spatial configurations of spatial constraint-II for ‘Under’.	95
6.18 Spatial configuration of spatial constraint-III for ‘Under’.	96
6.19 Spatial configuration of spatial constraint-I for ‘Above’.	97
6.20 Spatial configuration of spatial constraint-I for ‘Below’.	98
6.21 Spatial configuration of spatial constraint-I for ‘Inside’.	98
6.22 Spatial configuration of spatial constraint-I for ‘Outside’.	99
6.23 Bounds of the XZ plane of the reference entity’s inside empty space for locating ‘Left’, ‘Right’, ‘Front’, and ‘Back’.	100
6.24 Spatial configurations of spatial constraints-I for ‘Left’, ‘Right’, ‘Front’, and ‘Back’.	101
6.25 Spatial configurations of spatial constraints-II for ‘Left’, ‘Right’, ‘Front’, and ‘Back’.	102
6.26 Spatial configurations of spatial constraints-I for ‘In front of’, and ‘Behind’	103
6.27 Spatial configurations of spatial constraints-I for ‘Between’ and ‘Among’	104
6.28 Spatial configurations of spatial constraints-I for ‘Near’, and ‘Far’.	104
6.29 Graphical user interface for testing the prediction of spatial concept. It is carried out to see how effectively it predicts the spatial concept given the spatial configuration between the entities.	105
6.30 The Configurations of the entities and their predicted spatial con- cepts: part-I.	106
6.31 The Configurations of the entities and their predicted spatial con- cepts: part-II.	107
6.32 The Configurations of the entities and their predicted spatial con- cepts: part-III.	108

6.33	The sentences are generated from the configuration network. The thicker line shows that the entities intersect each other. The dotted lines are used to denote that the entities do not intersect.	110
6.34	The Sentences generation for the configuration-II	111
6.35	The Sentences generation for the configuration-III	112
6.36	The Sentences generation for the configuration-IV	113
6.37	The Sentences generation for the configuration-V	113
6.38	The Sentences generation for the configuration-VI	114
6.39	Understanding of the sentences ‘ <i>The table is on the ground. The desktop is left of the table. The monitor is right of the table. The keyboard is in front of the monitor</i> ’ using rigid body statics model.	118
6.40	Understanding of the sentences ‘ <i>The bed is on the ground. The chair is beside the bed. The table lamp is on the chair. The table fan is behind the bed. The chair is on the ground</i> ’ using rigid body statics model	120
6.41	Understanding of the sentences ‘ <i>The rack is on the ground. The bowl is in the rack. The teapot is in the rack. The paper is below the teapot. The paper weight is on the paper</i> ’ using rigid body statics model	120
6.42	Understanding of the sentences ‘ <i>The burger is in the bowl. The banana is in the plate. The plate is on the table. The bowl is left of the plate. The bucket is below the bowl</i> ’ using rigid body statics model	121
6.43	Understanding of the sentences ‘ <i>The clock on the table is over the bowl near the bucket</i> ’ using rigid body statics model	122
6.44	Understanding of the sentences ‘ <i>The ball near the pear/monitor is on the round table</i> ’ using rigid body statics model. There are two configurations for the sentence. The first one is with the pear and the other one is with the monitor.	123
6.45	Understanding of the sentence ‘ <i>The rose plant is on the ant</i> ’	124
6.46	Construction of modified configuration for the sentence ‘ <i>the bucket is on the ball</i> ’. It fails in the stability test.	125
6.47	Construction of modified configuration for the sentence ‘ <i>the bowl is inside the chair</i> ’. The chair does not have enclosed empty space so the constraint ‘inside’ is not applicable.	126

- 6.48 Construction of modified configuration for the sentence '*the desktop is in the bucket*'. The bucket empty space volume is lesser than the desktop volume so the constraint 'in' is not applicable between them. 127

List of Tables

3.1 Correspondence between conceptual world symbol to physical world
elements 36

Chapter 1

Introduction

1.1 Significance

Today's machines, though they have well developed physical abilities to perform actions, are not fully automated and they require manual programming or reprogramming by a human master to carry out intended tasks. As an alternative, is it possible to make the machines perform their intended tasks without human intervention and allow these machines co-exist with human beings and be treated as a human-like servants in our society. The pre-requisite for the machine to achieve the above is the autonomous capability to carry out the intended tasks. Then another important question arises, that is, how to enable the machines to become an autonomous or intelligent creature in our society.

A pre-requisite is to develop the mental ability for the machine to learn human language and understand the natural scenes in our environment. As a result, the processes underlying tasks can be encoded in human language such as English, Tamil, Chinese, etc and we can instruct the machine using our own language (i.e. English, Tamil, etc) instead of machine languages such as C, C++, Java, etc.

The machines which learn and understand the human languages can coexist with human beings and this is the primary requirement for the machine to become a sociable creature or agent in our society. The natural language text understanding system will pave the way to create potential applications such as linguistic programming (i.e. programming the autonomous machine or robot by human lan-

guage), linguistic based 3D modeling and dialogues with machines.

1.2 Applications

1.2.1 Conversational Dialogue Between a Humanoid Robot and a Human

Fig. 1.1 shows the application of conversational dialogue between a humanoid robot and a human. In this application, the robot and human interacts through speech. First, the speech is converted into text by a speech-to-text conversion system. Then, text is understood by text-meaning conversion system. As a result, the meaning is created. Next, the understood meaning is communicated to the other agent as speech through a speech synthesis process.

Figure 1.1: The generic application scenario of conversational dialogue between a humanoid robot and a human.

There are three possible outcomes as a result of the interaction between two agents using a natural language: (i) Deny: Speaker receives the utterance from Listener, who contradicts the composition of meanings. In this case, Listener is

most likely an expert and Speaker is a learner. (ii) Optimize: Speaker receives the utterance from Listener, who suggests a better composition of meanings. In this case, Listener is most likely an expert and Speaker is a learner. (iii) Confirm: Speaker receives the utterance from Listener, who asks for clarification. In this case, Listener is most likely a learner. If Listener is an expert, both outcomes of Deny and Optimize are possible. In this case, Speaker behaves like a learner. If Listener is not an expert, the most likely outcome will be Confirm. In this case, Listener behaves like a learner.

If the outcome is Deny, Listener contradicts Speaker's statement and Speaker learns from Listener. If the outcome is Optimize, Listener suggests a better way of representing the meaning or statement verbally as well as visually. (It is similar to the Auto-Correct feature of MS-Word, which gives suggestions, when a user commits a typo or a grammatical error.) If the outcome is Confirm, it indicates that Listener has difficulties in grasping the intention of Speaker. As a result, Listener further clarifies with Speaker.

1.2.2 Spatial Language to 3D Scene Conversion

The compound sentences such as *'The bed is on the ground. The chair is beside the bed. The table lamp is on the chair. The table fan is behind the bed. The chair is on the ground'* are considered for 3D scene creation. Fig. 1.2 shows conversion of spatial language to 3D scene conversion. The sentence 'the chair is beside the bed' can be interpreted in two ways. The first configuration would be the chair is placed on the bed's either right or left area. The second configuration would be where the chair is placed on the ground and the both should not touch each other. The sentence 'the chair is on the ground' will help the algorithm to select the second configuration. This is the actual output generated from this research work. The configuration network refers to position of one entity with respect to other entities in a certain pose in the graph form of representation.

Figure 1.2: The generation of 3D scene from the sentences ‘*The bed is on the ground. The chair is beside the bed. The table lamp is on the chair. The table fan is behind the bed. The chair is on the ground.*’.

1.3 Challenges

The increase in complexity of a machine (i.e. humanoid robot) results in the advantage to having complex, or even human-like, behaviors which otherwise are not possible with a simple arm manipulator. Most importantly, it gives rise to many challenging research issues which are still looking for engineering solutions. At a functional level, these challenging issues are to be addressed and they are as follows.

1.3.1 Learning Capability

A machine can be preprogrammed to repetitively execute instructions which solely reflect the intention of human masters. This mode of operation will not make a machine adaptable to a dynamically changing environment where unpredictable events may occur in both space and time. On the other hand, without learning capability, a machine will never be able to acquire new skills or behaviors in an autonomous manner (i.e. without human intervention in reprogramming).

1.3.2 Thinking Capability

Learning includes, but is much more than, memorizing. Most importantly, a learning machine must have the analytical skill in undertaking analysis and synthesis, which rely on the manipulation of conceptual entities (or knowledge). In other words, a machine without the innate mechanism of thinking and representation can never internalize any learning experience.

1.3.3 Seeing Capability

We are living in a physical world, which generically consists of entities (or objects), constraints and behaviors (i.e. ordered sequences of actions and interactions among entities). Without vision, a machine would not be able to perceive a physical world. This shortcoming will tremendously undermine a machine's ability to act and interact with others. Therefore, a machine must have a seeing capability.

1.3.4 Reading Capability

Our real world encompasses both natural and man-made entities (or objects). Among the man-made entities, a very important category of human inventions is the natural languages. It goes without saying that all meanings, which are the results of understanding, can be described in terms of texts of a written language. This fact also suggests that a natural language is a system for knowledge representation. To a great extent, we can say that all the published texts, which objectively describe meanings, form a conceptual world. Since the perception of conceptual world is equally important as the perception of physical world by any intelligent creature, a machine should master at least one natural language.

1.3.5 Listening Capability

When humans and intelligent machines co-exist, it is natural to expect that machines learn and speak human languages, instead of the current situation that humans have to learn and use a computer's languages, such as a high-level pro-

gramming language, in order to instruct machines to work for us. Therefore, it is a prerequisite for a machine to master the conversational speech with at least one natural language before it can become a sociable machine.

1.3.6 Autonomous Capability

With the advance achieved in control engineering today, it is not a difficult job to develop an automated machine, which repetitively produces a desired outcome, such as motion. As we know, it is relatively easy to specify a desired outcome of a single action, and to realize it with an automatic feedback control loop. However, it is a challenge to develop a machine in which the human invention remains at a higher level where the desired outcome, instructed by human, is not achievable by a single action, but a sequence of ordered actions (e.g. outcomes such as “to win a match”, “to clean a house”, etc).

1.4 Organization of Thesis

The thesis is organized as follows

- Chapter 2 discusses the existing works related to natural language understanding and their open issues, new methodology for developing the machine mental architecture, objective and scope of the research work, and contributions of the research work.
- Chapter 3 discusses what the meaning is, how to model the elementary meanings such as physical models and conceptual models.
- Chapter 4 discusses the construction of a configuration network by rigid body statics algorithm. It includes the discussion of intersection algorithms, a transformation matrix for the configuration network and a stability algorithm.
- Chapter 5 discusses the construction of a concept network from the configuration network, and constraints for a phrase and sentence formation.

- Chapter 6 discusses the learning of spatial concepts through modeling and learning the entities and their static interactions. First it discusses the architecture of KnowNet and its user interface. Then it describes the learning of the entities and their associated words, learning of a usage configuration, modeling and learning of spatial constraints, sentence generation experiments and understanding experiments.
- Chapter 7 ends with conclusions, contributions and future work.

Chapter 2

Problem Statement

This chapter discusses the existing works in the area of natural language understanding, issues related to the existing works and proposed methodology to overcome these issues.

2.1 Literature Review

Most of the works in natural language understanding to date is based on manipulating knowledge, which is pre-programmed by human master, by sophisticated algorithms. Such works can be called knowledge-based natural language understanding and they do not help the machine to autonomously learn and understand the human languages by interaction with the environment. Following that, symbol grounding theory has been proposed, which states that the meaning of the word or words should be represented in terms of machine sensory feature values. Using this theory, computational models have been developed for robots to learn language by real time interaction with the environment. However, they did not consider modeling the interaction between the entities and still rely on language semantic rules to interpret the meaning of the sentences.

This section discusses the definitions for intelligence, meaning theories and its representation in mind, symbol grounding theory and language learning robots, and works related to spatial language understanding and natural language processing.

2.1.1 Intelligence

Xie et al [1] discussed the definitions for intelligence and classified it into human intelligence and computerized human intelligence. Then he proposed the definition for robot or machine intelligence.

Human Intelligence

Based on common sense, various definitions for human intelligence have been emerged. Here are some example definitions (Russel and Norvig [2]):

- “Intelligence is an innate quality, as distinct from abilities acquired through learning” (Herbert Spencer).
- “Intelligence is the innate general cognitive ability” (Cyril Burt).
- “Intelligence is an inborn or innate quality, as distinct from abilities acquired through individual experience” (Encyclopedia Britannica).

Many people disagreed with these definitions because they do not consider the dimension of time. To remedy these deficiencies due to the lack of consideration of time (i.e. learning or experience), some other definitions have been put forward. These include:

- “Intelligence is the ability to carry out abstract thinking” (L. M. Terman).
- “Intelligence is the capacity to act purposefully, think rationally and deal effectively with the environment” (D. Wechsler).

Computerized Human Intelligence

With the advent of computers, researchers and scientists are excited about the possibility of computerizing human intelligence in the form of computer programs so as to make computers possess a certain degree of human-like intelligence. As a result of this human endeavor, a new technical term was born, that is : Artificial

Intelligence (AI). Interestingly enough, again, researchers and scientists could not agree on a common definition of AI. In the literature, some definitions are:

- “AI is the study of the computations that make it possible to perceive, reason, and act” (Winston).
- “AI is the study of mental faculties through the use of computational models” (Charniak and McDermott).
- “AI is the study of how to make computers do things at which, at the moment, people do better” (Rich and Knight).
- AI is the art of creating machines that perform functions that require intelligence when performed by people” (Kurzweil).

The study of AI historically focuses on problem-solving (i.e. analysis) and learning. The difficult issue of synthesis (i.e. creativity) has not yet received much attention. Although artificial intelligence literally means man-made intelligence in a machine, the content discussed under AI suggests that it actually deals with partly computerized human intelligence or computational rationality.

Machine or Robot Intelligence

Today’s computational tasks underlying computerized human intelligence are dictated by the results of programming undertaken by human beings. None of today’s computers have mastered a single language of its own or a natural language. Because of this limitation, a computer has not yet had the innate ability to simulate the human-like intelligence. In robotics, a computer is treated as the brain of a robot and new area of AI called Machine or Robot intelligence aims at equipping machines such as humanoid robots with the integrated ability to autonomously learn, understand, create and communicate meanings in the form of images and texts of natural language.

To achieve the above, Xie et al, in their paper, defined the intelligence from an engineering point of view as

- “Intelligence is the ability to link perception to actions for the purpose of achieving an intended outcome. Intelligence is a measurable attribute, and is inversely proportional to the effort spent in achieving the intended goal”.
- “Robot or machine intelligence is an attribute engendered by a robot’s brain, under the governance of causality and rationality”.

2.1.2 Meaning of a Word

It has been mentioned before that the meaning of a word is its (word’s) physical model and conceptual model. In philosophy, different theories for the meaning of the word have been proposed. But no theories suggest how to model the meaning of the word from an engineering point of view. Similar to that, in psychology, many theories have been put forward on how the meaning of the word(s) is represented in mind (Carrol [3] and Meaning in web [4, 5]).

Theories of Word Meaning

Denotation In this, the meaning of word is nothing but its (word’s) corresponding thing in the physical world. For example, by denotation, the meaning of the word *apple* is its corresponding object (*i.e. apple fruit*) in the physical world. Here the *apple fruit* is called the referent of the word *apple*. The referent is nothing but the physical world entity.

Connotation The meaning of word is its association or relationship with other words. Connotation is purely symbol systems and the relationships between the symbols form the meaning. For example, the meaning of *apple* is *apple is a fruit*, *apple is tasty* and *human eats the apple*. The sentences associating the words such as *fruit*, *tasty* and *human* with *apple* actually denote the meaning for *apple*. This theory of meaning is more dominant in the area of computational linguistics.

Structure The meaning of a word is its syntactical properties and rules. For example, as per structure theory, the meaning of *apple* is *noun* (*i.e. its syntactical*

property), and syntactical rules related to noun. The conceptual model follows this theory and it captures just one aspect of word meaning.

Semiotics Semiotics is the study of the meaning of signs. According to Pierce semiotic symbol is a sign, which in semiotics is a relation between representation, interpretant and object. Representation is a word, which is the shape of sign. Interpretant is the sense that is made of the sign. Object is the entity that stands for the sign and it may be abstractions, actions and other signs. According to Pierce, the sign becomes a symbol when its representation, in relation to its referent is arbitrary or conventionalized so that the relationship has to be learned; otherwise the sign is either icon or index [6]. Frege [7] also mentioned that the meaning of a word should consist of sense and reference.

2.1.3 Meaning Representation in Mind

Many theories of mind found in the literature give focus on how the knowledge should be represented in the memory of human (or machine) mind rather than how to manipulate the contents of the memory by mental processes. Here, three important meaning representation theories are going to be discussed and they are widely used in computational implementations. They are semantic networks (Collins [8, 9]), mental models (Craik [10] and Johnson-Laird [11, 12]), and episodic memory and semantic memory (Tulving [13], Kintsch [14], and Kintsch and Dijk [15]). Based on these, in computational linguistics, knowledge bases for text understanding have been developed. The knowledge bases, which are pre-programmed by the human master, store the language rules and world knowledge to understand the text. In contrast, the thesis focuses on how to model the contents of machine's memory so that machine can autonomously organize the knowledge about the real world through learning processes.

Semantic Network

A semantic network encodes the connotation meaning of a word and in it, the words are represented in nodes and the relationship between two words is represented in the edge, which connects two nodes.

Mental Models

The term *mental model* first was proposed by Kenneth Craik and he defined it as “small scale models of reality, which can be used to anticipate events”. Later Johnson-Laird criticized the semantic network on the grounds that it captures only the relationship between the words and it does not associate the word to its corresponding things or entities and events in the physical world. A mental model is a spatial representation of physical world entities. Humans construct the mental model while seeing the physical world or understanding the text.

Episodic and Semantic Memory

Human memory is divided into episodic memory and semantic memory. The episodic memory records the events and episodes of the physical world. The semantic memory contains the conceptual knowledge about the world in terms of words and their relations. The memories just reveal that there is an existence of physical world’s events and episodes and conceptual world’s concepts and topics in the human memory (Tulving [13], Kintsch [14] and Kintsch and Dijk [15]). But the works did not mention how both the memories help the human or machine to understand the text of natural language.

2.1.4 Symbol Grounding

In the last section, theories of meaning and its representation in mind were discussed. In this, works related to computational models of mind for language learning and understanding are discussed.

Since the invention of the computer, symbolic model of the mind has been the

dominant theory for language understanding. It states that the mind consists of symbols, syntax and semantics rules and manipulation of these symbols creates the thinking process in the mind (Fodor [16, 17, 18, 19], Newell [20], and Searle [21]).

Criticizing the symbolic model of mind, Harnad [22, 23] proposed the symbol grounding theory, in which the machine or robot should associate the word to the sensory feature values of the entity (i.e. that corresponds to the word) in the physical world. If the association of symbol and feature values is successful, then the robot is said to have understood or grounded the meaning of the word. Following that, computational models have been developed for the robot to learn and understand the natural language by real time interaction with the physical world.

2.1.5 Symbol Grounding Experiments

Steels [24, 25, 26, 27, 28] examined the hypothesis that language communication is bootstrapped in a social learning process based on the notion of language game. In the language game, a human acts as a mediator for assisting an AIBO robot to ground the meaning of the symbol and the AIBO learns concepts such as ball, red, etc and behaviors such as walk, stand and look.

Roy [29, 30, 31, 32] implemented a grounded language learning model called CELL (Cross-channel Early Lexical Learning). CELL learns the meaning of words from untranscribed acoustic and video input signal. As a result of learning, the word is represented in terms of sensory feature values of the entity correspond to the word.

Vogt [33, 34] and Cohen [35, 36] carried out similar experiments for the robot to learn the meaning of words that correspond to the entity and action from the sensory signals. Katarzyniak [37] discussed the theoretical aspects of internal structure for cognitive agents which ground the language by interaction with the physical world.

2.1.6 Knowledge-Based Natural Language Understanding

Most of the works in natural language processing belong to the domain of computer-aided natural language understanding, in which knowledge required for understanding the text is pre-programmed and then the knowledge is manipulated with computational algorithms. The manipulation is called understanding. Such knowledge based systems do not help the machine to learn and understand the natural language by interaction with the environment. This section discusses the works related to knowledge bases for text understanding, natural language understanding systems, natural language interfaces to robots and virtual reality systems and visualization of meaning in natural language.

Knowledge-Bases for Natural Language Understanding

This section discusses knowledge bases that are widely used in understanding of natural language texts.

WordNet: This is an on-line lexical knowledge base whose design is inspired by current psycholinguistic theories of human lexical memory and it is the most widely used semantic lexical source for the task of text understanding by artificial intelligence and computational linguistic communities (Miller [38]). Its lexicon contains verbs, nouns, adjectives, adverbs, and lexical information is organized in terms of word meanings that are represented by the relations such as hyperonymy, hyponymy, meronymy, synsets, and antonymy. One great advantage of the WordNet is that it has very nice classification of the physical world's entities, behaviors and events (i.e. hyperonymy) and it encoded different senses for each word in terms of its synsets. The gloss of the word is preprogrammed as strings of annotated words.

MindNet: This is a lexical knowledge base constructed automatically from the definitions and example sentences from the machine readable dictionaries such as Longman's Dictionary of Contemporary English (LDOCE) and American Heritage

Dictionary, 3rd edition (AHD3), using a broad coverage parser. Through parsing, it captures the semantic relation between the words in the dictionary definitions and these named semantic relations are used to populate the semantic network (Richardson et al [39]).

ConceptNet: This is a commonsense knowledge base encoded in a large scale semantic network. It is produced from the Open Mind Common Sense (OMCS) corpus. It contains the following categories of concepts: things, events, actions, spatial, goals, functions and generic. It captures the different aspects of the physical world through simple assertions, description of typical situations and stories (Liu and Singh [40]).

Cyc: This is a commonsense knowledge base that captures the physical world's facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The knowledge in Cyc is represented in a rich logical language called CycL. It has the ability to integrate with other knowledge bases to enhance its reasoning capability (Lenat [41]).

VerbNet: This is a verb lexicon that combines the thematic roles and semantic predicates with syntactic frames and selectional restrictions and it is constructed based on Levin's verb classes. Actually, it was developed to improve the capability of WordNet's verb lexicon. It contains the semantic roles such as agent, patient, manner, result, etc and is useful for shallow semantic representation of the input text (Kipper et al [42]).

Annotated Text Corpora: Corpus based natural language text understanding systems use statistical methods for lexicon learning, part of speech tagging, parsing, word sense disambiguation and semantic parser learning. An annotated corpus is used as the training data to learn the rules to analyze the raw or input text. Corpus annotation is a complicated task and requires lots of human effort

to annotate it. PropBank (Marcus et al [43]) and FrameNet (Collin et al [44]) corpora are annotated with semantic roles or predicate-agreement structure. And SemCor [45] and Open Mind Word Expert corpora (Tim and Rada [46]) are annotated with WordNet senses.

Natural Language Understanding Systems

Almost all the language understanding systems use the same procedure to interpret the meaning of sentences. Manipulating the knowledge with the computational algorithms forms the notion of text understanding. In these systems, first the input text is acquired through speech recognition system or key board and it is tokenized to form tokens or a set of words. The tokens are passed to a syntactic analyzer to determine the phrases and dependency relations. Then the output of the syntactic analyzer is passed to a semantic parser to determine the meaning of sentences. Finally, the meaning representation (Blackburn and Bos [47]) is passed as a query to domain dependent knowledge base to carry out the desired tasks. The tasks can be, for instance, information retrieval (Robert and Suzanne [48], Manning and Schutze [49], and Allen [50]), dialogues between machine and human (Dowding et al [51] and Allen [52]), story understanding (Mueller [53]), and spatial language understanding (Skubic et al [54], Adams et al [55] and Coventry [56]).

Natural Language Interfaces

In this, robots and agents in virtual systems are controlled by natural language commands. Works such as the Kairai System (Shinyama et al [57]), AnimNL (Badler et al [58]), CommandTalk (Moore et al [59]), and QuickSet (Cohen et al [60]) are some of the representative examples in the domain of natural language controlled animation systems. And works such as the HARIS Project (Veno [61]), the CARL Project (Lopes and Teixeira [62]), KANTRA (Lueth et al [63]), and the ROMAN Service Robot system (Fischer and Schmidt [64]) are some of the representative works in the areas of natural language controlled robotic systems.

Natural Language Visualization

In this kind of system, first text is understood by the text understanding system and the meaning representation is assigned with corresponding graphical models, which are then rendered to produce the static or dynamic scene. WordsEye (Coyne and Sproat [65]) is a such system, which converts natural language text input into static scenes. CarSim (Dupuy et al [66]) creates animated scenes of a typical accident report.

2.1.7 Spatial Language Understanding

Herskovits [67], Talmy [68] and Lakoff [69] gave detailed description of meanings of spatial prepositions in theoretical perspective. Regier [70] and Logan and Sadler [71] discussed the spatial prepositions prototypical meaning templates using the geometrical parameters of the entities such as distance, location, centroid, etc. To criticize these methods, Coventry et al [72] proposed that the spatial language should be modeled based on the geometry and extra-geometric variables of the entities. He means ‘geometry variables’ as distance between the entities, their location and orientation. The extra-geometrical variables are function of the entity, shape and size of the entity. The functional relations of the entities play the important role in modeling the spatial words. The functional relations mean the interaction between the entities and the usage of the entity. Martinez et al [73] proposed the hybrid neural network and virtual reality system to learn the meaning of prepositions such as over, above, under and below. The geometry and extra-geometry parameters are given as input and the neural network learns to associate the parameters with the spatial words. The simple modeling parameters such as bounding box information, entity shape approximated as hemispherical, conical and cuboid and the pre-defined function of the object were considered for the modeling.

The computational models presented in the existing works have the following limitations: a) the entities location and distance between them play an important role in modeling the spatial language. b) The computational models do not con-

sider the interaction between the entities. In other words, the number of possible ways how one entity can be positioned with respect to the other entities. c) The computational models are used to only predict the spatial relationship between the entities, and they lack the ability to construct the configuration or scene of the entities from the meanings of spatial prepositions.

2.2 Issues Related to Existing Research Works

In the current literature, works related to natural language text understanding can be classified into knowledge-based systems and symbol grounding systems. The main purpose of the research work is how to make the machine to be an autonomous agent so that it can coexist with human beings and it can be treated as human-like servants. For the machine to achieve the autonomous ability, it should have the mental ability to understand the human language (i.e. natural language) and to understand the physical world elements (i.e. our environment). In this thesis, the focus of the work is to develop the mental ability for the machine to understand the natural language text through interaction with the environment.

2.2.1 Issues Related to Knowledge-Based Natural Language Understanding

In the knowledge-based systems, the words syntactic and semantics information are stored in the database or are generated from the text corpora using statistical learning algorithms. The manipulation of the knowledge with help of algorithms such as semantic parser, Hidden-Markov Model, etc creates the text understanding. The knowledge-based system is purely a symbol based approach and it does not have any interaction with the environment. So, the symbol based systems can't be used for developing the autonomous machine.

2.2.2 Issues Related to Symbol Grounding

Criticizing the knowledge-based natural language understanding systems, Harnad [22, 23] proposed the symbol grounding theory, in which the machine or robot should associate the word to the sensory feature values of the entity (i.e. that corresponds to the word) in the physical world. If the association of symbol and feature values is successful, then the robot is said to have understood or grounded the meaning of the word. Following that, computational models have been developed for the robot to learn and understand the natural language by real time interaction with the physical world.

However, the above computational models for language learning and understanding have the following limitations: a) The feature values can be seen as another way of representing the entity. Just capturing the color and two dimensional shape parameters does not constitute the complete representation or modeling of the entity. So, the feature values extracted from the signal are incomplete. b) The feature values are helpful for the robot simply recognizing the entity in the signals. The above systems assume that recognition is understanding, but actually it is not so. c) They use preprogrammed semantic rules to understand the meaning of the set of words (i.e. sentence) and also preprogrammed action rules to execute the behaviors in robot. Because of these limitations, machines using the above computational models for language learning and understanding cannot become a autonomous agent in the society.

The feature values or properties of the entity in the physical world are a key part. This work will discuss how these properties help the machine to autonomously learn and understand the natural language through interaction with the physical world. The machine should understand the entities and their interactions before it starts to learn and understand the natural language related to spatial relations. This thesis focuses on developing a mind or mental architecture for the machine to autonomously undertake the learning, analysis and synthesis processes of text

understanding. To achieve that, the following issues are addressed from an engineering point of view:

- What would be the architecture of the machine's mind?
- How to model the content of the machine's brain memory or what is the design of organized memory?
- How important is our physical world or environment for understanding the text by the machine?
- What is the meaning? And how to model the meaning of the symbol and entity so that machine can learn the meanings through its sensory systems or programmed interface by real time interaction with the environment?
- how to enable the machine to understand or verify the meaning of the sentence(s) with help of constraint physics algorithm?

The following section discusses the solutions to the above issues.

2.3 Proposed Methodology

2.3.1 Machine Mental Architecture

Fig. 2.1 shows the machine mental architecture for text understanding (Xie et al [1]). The machine mind consists of mental processes and organized memory and they interact with each other in order to derive the meanings from the text. The mind is embodied in the body of the machine or vice versa. Real world consists of a physical world and a conceptual world. The physical world and conceptual world refer to our environment and texts of natural language respectively. Organized memory is the machine's brain memory, which contains the modeling of physical and conceptual world elements. It can also be referred as 'design of intelligence' or 'KnowNet' for the understanding of text. This organized memory, once designed, is enriched (i.e. it acquires its contents) through the learning processes by real-time interaction with the environment. The mental processes of the machine mind

Figure 2.1: A machine's mental architecture. The interaction between mental processes and organized memory creates the phenomena of text understanding

are learning, analysis, and synthesis. The interaction of these processes with the contents of the organized memory creates the phenomena of text understanding. The design of an organized memory is the key need for text understanding. Analysis refers to the process of understanding the text acquired from the real world and synthesis refers to conveying of the understood meaning in the form of text or images to the agents in the real world.

Organized Memory: KnowNet

The physical world or environment consists of man-made and natural-made entities, which exist in space and time. The entities are the basic building blocks of the physical world. The entities are never in isolation. Instead, they act and interact with one another through forces such as gravity, magnetic force, mechanical force, etc. The interaction between the entities creates the cause-effect phenomena in the physical world and forms the basis for creating action, behavior, events and episodes in the physical world. Most importantly, the interaction helps to determine the logics or truthfulness of text meaning. Logics is the understanding of the cause-effect phenomena among the entities in the physical world by the agent.

The interaction between entities is controlled by physical world physics or con-

straint physics. In physics, modeling the interaction between the entities helps us to study their cause-effect phenomena, for example, rigid body statics and dynamics. The entity in the physical world can be modeled by properties such as mechanical, geometrical, chemical, etc. Because of the interaction among the entities in the physical world, some entities may produce certain behaviors, some may receive the actions and some may transform one form of energy into the other. The capability of the entity to produce the above characteristics is called ‘constraint’. So, each entity in the physical world has two important components such as properties and constraints. The properties and constraints can be seen as representation of the entity in terms of certain measurable attributes. The constraint physics models can reconstruct the entity from its properties as well as it can determine and control the interactions among the entities. Chapter 3 discusses more about the entity’s properties and constraints.

Figure 2.2: Architecture of an organized memory, in which modeling of meanings of the physical and conceptual world are represented. Organized memory acquires the meaning (i.e. properties and constraints) through learning process. More specifically, the properties are acquired through sensors and the constraints are learned by modeling the interactions among the entities.

Similar to the physical world, the word is the basic element in the conceptual

world and words interact with one another through syntax or grammar rules in order to form phrases or sentences. In the conceptual world, properties refer to noun, verb, adjective, etc and constraints are the grammar rules.

Fig. 2.2 shows the architecture of the organized memory. The organized memory consists of elementary and composite meanings both for conceptual and physical world. Elementary meanings contain physical models and conceptual models. Physical models include entity's properties and constraints. Conceptual models refer to the word's grammatical properties and constraints. Composite meanings in the physical world (i.e. physical constraint networks) are constructed by combining the physical models through the mediation of the constraint physics algorithm. Similarly, conceptual constraint networks are nothing but natural language sentences, which are constructed by combining the conceptual models with help of grammar rules and physical constraint networks.

The physical constraint networks consist of configuration networks, behavior networks, event networks and episode networks. The configuration network refers to spatial arrangement of entities in space. The behavior network refers to chains of cause-effect phenomena in the configuration network or system of rigid bodies. Interactions of sets of events and episodes creates the event network and episode network respectively. The conceptual constraint network consists of a concept network, a topic network, a belief network and a value network. The concept network is a sentence or set of sentences, which is the description of a configuration network or behavior network. The topic network is a paragraph or set of paragraphs, which are the description of event or episode network. The value network is a concept or topic network, which tells the machine good and bad things about the activities in the physical world. The belief network allows the machine to accept or believe the concept or topic of other agents without any contradiction.

2.3.2 Proposed Method for Spatial Language Learning, Understanding and Generation

Entities Encoding and Decoding Processes

Fig. 2.3 shows the encoding and decoding processes of the entities in the physical world. This is the fundamental philosophy behind the development of mental architecture. The entities are transformed into properties through the sensors or programmed interface and this is called encoding of an entity. Then the properties can be mapped to virtual entities and their possible interactions with other entities are achieved through constraint physics models such as computer graphics, rigid body statics and dynamics, etc. This process is called decoding of an entity.

Figure 2.3: Entities encoding and decoding processes. Entities are encoded or mapped to properties through the sensors or programmed interface. The properties can be decoded or mapped to virtual entities and their possible interactions with other entities are achieved through the constraint physics models such as computer graphics, rigid body statics and dynamics, etc.

Framework for Learning of Words Through Entities Properties and Their Interactions

Fig. 2.4 shows the learning of words through entities properties and their interactions. First entities and their interactions are modeled by properties and constraints respectively. Assigning certain values to properties and constraints based on the entity and its interaction with other entities is called learning. The properties and constraints of the entity are called physical models. For each entity and constraint

a name or word is assigned. In other words, entities and constraints are associated with certain words. Then for each word, certain lexical properties and constraints are assigned. The word's lexical properties and constraints are called conceptual models.

Figure 2.4: Learning of words through entities properties and their interactions

Framework for Sentence Generation with Physical and Conceptual Models

Fig. 2.5 shows the generation of sentences using the physical and conceptual models. First, a set of entities are given as input. For each entity its corresponding physical model is retrieved. Then the physical models are combined and organized by rigid body statics model in order to construct the configuration network. Then the configuration network and the conceptual models are input to the text synthesis algorithm to generate the sentence(s). From the configuration network, the information such as subject, object, verb and spatial relationship between the subject and object are obtained. The conceptual models are then used to form phrases and sentence(s).

Figure 2.5: Sentence generation using physical and conceptual models

Framework for Understanding of the Sentence

Fig. 2.6 shows the procedure for understanding the sentences related spatial relations. First, using syntactical analysis, phrases such as noun phrase, verb phrase, prepositional phrase are identified in the sentence.

For each phrase, the corresponding physical models are retrieved and then passed to the rigid body statics model in order to construct the configuration network. While constructing the configuration network, two possible scenarios arise: a) the configuration network as per the sentence information, which does not violate the truthfulness of the physical world and b) the modified configuration network, which arises when the sentence information violates the truthfulness of the physical world. The violating information in the sentence are mainly due to incorrect subject and object and wrong spatial information between the subject and object. The modified sentence will be generated by the text synthesis algorithm given the input the modified configuration network and the conceptual models.

Figure 2.6: Method for understanding the sentence

2.4 Objective and Scope

The objective of the work is to design and develop a mental architecture for the machine to autonomously learn, analyze and synthesize the natural language related to spatial relations.

2.4.1 Scope of the Work

- The machine learns the elementary meanings through a programmed interface and it does not use any sensory systems for learning.
- The learning and understanding capability of the implemented system is limited to entities and spatial positions and their associated words. The spatial position words such as *in*, *on*, *over*, *under*, *above*, *below*, *inside*, *outside*, *behind*, *in front of*, *back*, *front*, *left*, *right*, *beside*, *between*, *among*, *near* and *far*

are considered for learning and understanding. The entities are assumed to be *rigid bodies* for modeling their interactions.

- The work focuses on static interaction among the entities that involve only spatial position relationship among the entities. In other words, the construction of configuration network by rigid body statics model is the main focus of the research work.

2.5 Contributions

The contributions of the thesis are as follows.

- Proposed and implemented mental architecture for the machine to understand natural language related to spatial relations. The implemented system is called ‘KnowNet’.
- Organized memory is designed, in which modeling of the physical and conceptual world elements are represented. It consists of elementary meanings and composite meanings. The elementary meanings are acquired or learned by a machine through a programmed interface by real time interaction with the human agent.
- Designed and developed rigid body statics algorithm for the machine to understand the meaning of sentence(s) that are related to entities and spatial relations. KnowNet does not use semantic knowledge of the words for the understanding.

2.6 Summary

In Summary,

- Different definitions for intelligence, meaning theories and theories related to meaning representation in the mind have been discussed.

- Since the invention of computer, symbolic model of mind was the dominant theory for text understanding. The theory states that mind consists of symbol systems and manipulation of symbols based on syntax and semantic rules form the thinking process in the mind. Harnad criticized the theory of symbolic model of mind on the grounds that the meaning of the symbols are pre-programmed by human. Then he proposed the symbol grounding theory. This states that machine should represent the meaning of symbols in terms of sensory feature values of the entity, that corresponds to the symbol, in the physical world.
- Based on the symbol grounding, computational models have been developed for the robot to learn the word meanings. The feature values or properties that are acquired from the sensory signal do not constitute the complete representation or encoding for the entity.
- The research work is related to symbol grounding works. A new machine mental architecture is developed to overcome the issues related to symbol grounding computational models of natural language learning and understanding.
- Just given the properties of the entities, a machine has to determine the possible interactions between the entities and possible actions or behaviors that can be generated by the entity. This is the important pre-requirement for the machine to learn any natural language.
- Frameworks for spatial language learning, understanding and synthesis are proposed.

Chapter 3

Modeling of Elementary Meanings

An intelligent agent should produce composite meanings and undertake autonomous mental simulation by combining the learned elementary meanings using constraint physics model. Elementary meanings in the physical and conceptual world refer to physical and conceptual models respectively. The agent should learn and understand elementary meanings before it undertakes the creation of composite meanings or the task of text understanding. Then, another important question to ask is how to model the elementary meanings so that the machine learns or acquires them through programmed interface or sensory systems by real-time interaction with environment. This chapter discusses the importance of the physical world in understanding text by machine, how meaning and knowledge are related, how to model the elementary meanings in physical and conceptual worlds and how to model the mechanical constraints of the rigid bodies.

3.1 Real World

We live in the real world, which encompasses both natural-made and man-made things. For text understanding, the real world can be divided into physical and conceptual worlds. The physical world refers to our environment and the conceptual world refers to texts of natural language, which has existence in the form of books, articles, reports, etc.

3.1.1 Physical World

The main elements of the physical world is entities. The entities can be divided into natural-made and man-made objects. Humans and machines perceive the physical world through their sensory systems. Each entity in the physical world exists in space and time. The existence of the entities in the space that forms a certain spatial arrangement is called a configuration. Change of configuration with respect to time forms the basis for creating the actions, behaviors, events and episodes.

The entities in the physical world are never in isolation. Instead, they act and interact with one another. The interactions create the cause-effect phenomena in the physical world and form the basis for creating the action, behavior, event and episode in the physical world. In physics, modeling these interactions help us to study their cause-effect phenomena, for example, rigid body statics and dynamics.

We all agree that action is the source of results (or outcomes). In general, a specific outcome is only achievable by a sequence of ordered actions (or operations). This engenders the notion of behavior (or process). By definition, behavior (or process) is a sequence of ordered actions (or operations) which result in an intended or unintended outcome. Accordingly, the actor of behavior (or process) is called the agent (e.g. machine). Since an actor can exhibit behaviors (e.g. lift, run, hit, etc) and is situated in a configuration (i.e. exist in the space), the dynamics (or behavior) of a configuration, in other words, change of configuration with respect to time by the influence of forces creates the notion of an event, which can last for any time. When an agent produces a behavior or is affected by the behavior, we call this an event. For example, in 'he runs' 'he' is called the agent, 'run' is called the behavior and 'he runs' is an event and it can last for any time. When a set of events act, interact and evolve over a certain range in space and a certain (long) period in time, this phenomena is referred to an episode and examples for the episode are manufacturing activity, football game, etc. So, the constituents of the physical world are as follows

$$entities \rightarrow constraints \rightarrow behaviors \rightarrow events \rightarrow episodes$$

and Fig. 3.1 shows the elements of the physical world.

Figure 3.1: An illustration of the duality of the real world, in which the physical and conceptual world coexist. The properties and logics behind the physical and conceptual worlds form the body of knowledge

3.1.2 Conceptual World

Among natural creatures, human beings are very special in that they know how to observe results, do analysis, plan actions, and make new things. These types of behavior is normally called *experiments* (or *games* in layman's terms). Moreover, the actions involved in experiments are encapsulated in the notion of a *process* as follows:

$$input \rightarrow process \rightarrow output$$

For example, an experiment can be the invention of new drugs, new materials, or physical theories.

Interestingly enough, it is the activity of undertaking experiments, which results in the notion of *understanding*, which is characterized by another important notion of *meaning*, as follows:

- The relationship between input and output, which forms part of *meaning*
- The properties of input, which also form part of *meaning*

- The properties of output, which also form part of *meaning*

The meaning of operations inside a process is called logics and its representation in the form of text results in the notion of a *concept*. For example, in manufacturing, we have the concepts of heating, solidification, material removal, etc. And, the accumulation of concepts relevant to the same process gives rise to the notion of *topic*, such as welding, machining, assembly, etc.

For example, let us consider the story of invention of gravity by Newton. In the physical world, there was an apple tree, the apple fell down from the tree and everybody noticed the *event* of ‘apple falling on the ground’. This is the event that normally happens in the physical world. Only, Newton who identified the reason why the apple fell down from the tree and invented the concept of gravity. In this ‘apple falling on the ground’ is the physical world event. The understanding on the event ‘why the apple is falling to the ground’ is called logics. Another good example is the invention of theory of buoyancy by Archimedes.

How should we convey or communicate the understood meaning or logics to others? Natural language text is used to describe or represent the logics. A natural language is a man-made thing, which is also a result of evolution and perfection over a long period of time. It serves the purpose of communicating concepts and topics in both spoken and written forms.

It is clear that text in natural language is a vehicle which carries logics of the physical world elements. The basic element of the text or conceptual world is the *word*. Each word can be combined by syntax or grammar rules in order to form grammatically correct sentences. Most importantly, grammatically formed sentences should reflect the *logics* of the physical world and otherwise the sentences won’t be meaningful. Hence the conceptual world consists of the following constituent elements:

$$word \rightarrow syntax \rightarrow logics \rightarrow concepts \rightarrow topics$$

as shown in Fig. 3.1

It can be seen that interaction of word, logics and syntax forms the concepts and topics. The logics are acquired by understanding the physical world elements.

3.1.3 Meaning and Knowledge

The properties and constraints of physical world elements are identified through experiments and the understood properties and constraints (i.e. meanings) can be represented in text, which forms concepts. Accumulation of concepts forms the topics. The accumulation of topics forms what we call *knowledge*. This body of knowledge motivates us to enhance our skills and invent new things.

The words are basic elements of natural language, each of which may consist of (a) a label or name, (b) conceptual models (i.e. syntax), (c) physical models. The next important question is how to model these elementary meanings so that a machine can autonomously learn, analyze and synthesize the texts of natural language related to spatial relations.

Correspondence between conceptual symbols and physical world elements and meaning ambiguity: In the physical world, entities and their interactions exist with respect to space and time, and due to this the elements of the physical world each have their own unique meaning. As a result, there is no meaning ambiguity in the physical world. But, in the conceptual world one symbol can refer to different things in the physical world. For example, the symbol ‘ball’ refers to an ‘object’ for playing with and ‘dance’ refers to a dance event in the physical world. Table. 3.1 shows how the physical world meanings correspond to different conceptual symbol(s).

3.2 Physical Models

The properties and constraints of entities in the physical world refer to physical models. An entity is represented or modeled by its properties, and the constraints are learned by modeling the interactions among the entities. A property reflects the

Table 3.1: Correspondence between conceptual world symbol to physical world elements

Conceptual world Symbol	Physical World Elements
Car, Table	Entity
on, below	Constraints
Hold, Move	Action
Walk, Run	Behavior
‘He Runs’, Flight	Event
Raining, Manufacturing	Episode

attributes or features of the entity. A constraint refers to capability of the entity to produce certain behavior, to produce certain interactions with other entities, and to transform one form of energy into the another.

We have seen that an entity can generate the force(s) or it can be acted upon by the force(s). The force(s) can also do another great work in the physical world; it causes the two or more entities interact with one another. The conversion of chemical energy into mechanical energy by a car’s engine mechanism is a good example for the interactions. Then, another important question arises, that is, how to control these interactions to produce the desired effect. A *Constraint physics* does that job. In the above example, the constraint physics called ‘rigid body dynamics and combustion physics’ plays the important role in a car’s engine mechanism, which converts the chemical energy into mechanical energy.

3.2.1 Properties and Constraints

The basic element of the physical world is an entity. An entity can be a natural-made or man-made things. The entities can be divided into single-body entities or primitive entities and multi-body entities based on their structure. In a multi-body entity, each primitive entity is connected by certain spatial or kinematic constraint in order to form an active or passive mechanism. This research work focuses on modeling the properties and constraints of rigid bodies. Fig. 3.2 shows the different properties and constraints of rigid bodies. In the physical world, the properties of the rigid bodies are as follows: chemical properties, thermal properties, dimen-

Figure 3.2: The properties and constraints of an entity form the basis for elementary meanings. They can be acquired by the machine through its sensory systems or programmed interface by realtime interaction with the environment.

sional or geometric properties, mechanical properties, acoustic properties, electrical properties, optical properties and magnetic properties (Callister [74]).

Each entity may have the following constraints: spatial constraints, mechanical constraints, electrical constraints, thermal constraints, acoustic constraints, optical constraints and chemical constraints. The spatial constraints are the static interactions among the entities, which determine the spatial relationships between the entities. The mechanical constraints deal with modeling the dynamic interactions among the entities. In other words, the entities ability to transform one form of energy to mechanical form of energy is called mechanical constraints. The same definition can also be applied to other constraints except spatial constraints.

3.2.2 Modeling of Mechanical Constraint for Rigid Bodies

The mechanical constraints come into picture when mechanical forces (i.e. translational force and rotational force) act upon or are produced by the rigid bodies. They play an important role to spatially arrange the rigid bodies to form an active or passive mechanism (i.e. kinematic constraints) and also to produce the desired motions by the application of force or torque on the rigid bodies (i.e. dynamic constraints). The kinematic and dynamic constraints are useful for learning the symbols that correspond to configurations and actions.

Cause-Effect Relationship Between Mechanical Forces and Rigid-body

When mechanical forces such as translational force and torque act on the rigid body, it causes the body to produce the motions (Craig [75] and Murray et al [76]). This can be captured by the well known Newton-Euler equation of motion for rigid bodies and can be written as:

$$\begin{aligned}
 \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} &= m(q)\ddot{q} + v(q, \dot{q}) + g(q) + f(q, \dot{q}) \\
 q &= \begin{bmatrix} X \\ Y \\ Z \\ \alpha \\ \beta \\ \gamma \end{bmatrix} \\
 Q &= \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix}
 \end{aligned} \tag{3.1}$$

where Q is the generalized forces, in which F is the translational force and M is the moments. q is the generalized coordinates. $\dot{q}$ and $\ddot{q}$ are velocity and acceleration of generalized components respectively. $m(q)$ is mass matrix of the rigid body. $v(q, \dot{q})$ is the velocity components. $g(q)$ is the gravity components and $f(q, \dot{q})$ is the frictional components.

Using the parameters of the Newton-Euler equation such as generalized force and generalized motion variables, kinematic constraints and dynamic constraints are modeled and they are very important for the machine to learn the action and behavior of an entity. If the kinematic and dynamic constraints and properties of entities passed to Newton-Euler equation correspond to a system of rigid bodies, the model will decide the interactions between the entities. Adams dynamics simulator [77] and MatLab SimMechanics [78] are the good examples for modeling the interactions between the rigid bodies.

Kinematic Constraint

To produce the desired motion or force from the entity, components (i.e. a set of rigid bodies) in the entity should be placed in a certain configuration. The

kinematic constraint captures the position and orientation of rigid body A with respect to rigid body B. The kinematic constraints of a system of rigid bodies produce the configuration network, which can be represented by a graph data structure and chapter 4 discusses the configuration network in more detail. The kinematic constraint is applicable not only to configurations of machine components but also to configurations of any rigid bodies in the physical world. The generalized expression for the kinematic constraint of rigid body A with respect to rigid body B is given by

$$\begin{aligned} {}^B_A\mathbf{K} &= mTnR(X, Y, Z, \alpha, \beta, \gamma : {}^B_A\mathbf{T}) \\ {}^B_A\mathbf{T} &= \begin{bmatrix} {}^B_A\mathbf{R} & {}^B_A\mathbf{P} \\ 1 & 1 \end{bmatrix} \end{aligned} \quad (3.2)$$

Where ${}^B_A\mathbf{K}$ is the kinematic constraint expression. m is the number of translational motions (i.e. translational DOF) between the bodies and n is the number of rotational motions between the bodies, and m and n each will have maximum value of three. $(X, Y, Z, \alpha, \beta, \gamma)$ refers to relative motion variables. ${}^B_A\mathbf{T}$ is the transformation matrix of body A with respect to body B, in which ${}^B_A\mathbf{R}$ refers to the rotation matrix and ${}^B_A\mathbf{P}$ refers to the translation matrix.

The kinematic constraint captures the following information:

- It captures the degrees of freedom (DOF) of rigid body A with respect to rigid body B (i.e. relative motion of body A with respect to body B) and motion variable. This determines the capability of the entity to produce certain action and behavior by the application of force(s).
- It captures the position and orientation (i.e. pose) of body A with respect to body B.

The pose parameters are used for the machine to learn meanings of prepositions such as on, above, below, near, ahead, etc.

Dynamic Constraint

The dynamic constraint models the action and behavior of the entity. The generalized expression for the dynamic constraint of the body A is given by:

$${}_AD = m\ddot{T}n\ddot{R}(\ddot{X}, \ddot{Y}, \ddot{Z}, \ddot{\alpha}, \ddot{\beta}, \ddot{\gamma}) : rFsM(F_x, F_y, F_z, M_x, M_y, M_z) \quad (3.3)$$

where ${}_AD$ is the dynamic constraint of body A. F is the translational force and M is the rotational force or moments. r and s refer to the number of force components and each has maximum value of three. $m\ddot{T}n\ddot{R}$ is the direction of application of force by the body A on body B (i.e. configuration of end-effector in terms of acceleration).

The dynamic constraint captures the following information:

- It captures the direction (i.e. configuration) of application of force on a rigid body and its (force) components
- It also captures the effect of the application of force on a body.
- It models the action and behavior in any natural language.

The dynamic constraint is a useful property for the machine allowing it to learn that the meaning of a symbol corresponds to static and dynamic actions. For example, consider the dynamic constraint $0\ddot{T}0\ddot{R} : 3F0M$ from which it can be inferred that the end effector of the rigid body A applies the translational force to the body B, and it is important to note that there is no relative motion between the bodies, which forms the passive mechanism. This can be represented in the conceptual world as *hold*. The motion of body B as a result of application of force by a body A can be represented as $3\ddot{T}3\ddot{R}$. This means that the motion has six degrees of freedom and in conceptual world it can be represented as *fly*. It is important to note that the constraint expression for the flying does not have force components. If there is no force components the constraint will refer to the motions of a rigid body.

The dynamic constraint is a useful property to model the behavior network or cause-effect phenomena in the physical world and using this, dynamic simulation of entities can be performed.

Figure 3.3: Elementary meanings of conceptual symbols

3.3 Conceptual Models

The basic constituent of the conceptual world is the word. The words are arranged or grouped together by the grammar rules in order to form phrases, sentences and paragraphs. A machine can produce the meaningful sentences only if it knows the logics of the physical world.

3.3.1 Properties and Constraints

Noun, verb, adjective, adverb, preposition, determiner, and conjunction refer to properties of words in the English language (Allen [50]). In the conceptual world,

there are two types of constraints: pre constraint and post constraint. A pre-constraint refers to a property which always comes before the current word's property and a post-constraint refers to the property which will be placed after the current word's property. For example, the property *noun* will have *determiner*, *adjective*, and *noun* as pre-constraints and *noun* as post-constraint as in Fig. 3.3.

3.4 Summary

In summary,

- Natural language is nothing but projection of physical world meanings onto the conceptual world in the form of text.
- Physical world plays an important role in giving meaning to the text. We use text to represent the properties and constraints of the entities in the physical world. Interestingly enough, there is no meaning ambiguity in the physical world.
- In the physical world, entities are the basic elements and they act and interact with one another by the application of forces. In the conceptual world, words are the basic elements and they are grouped by grammar rules in order to form grammatically correct sentences.
- In general, entity in the physical world can be represented or encoded by its properties and constraints. The properties refer to attributes or features of an entity and the constraints refer to the capability of the entities to produce certain behaviors, to produce certain interactions with other entities, and to transform one form of energy into the another.
- Mechanism is used to convert one form of energy into the another and constraint physics controls the conversion process.
- In this thesis, only properties and constraints related to the rigid body are discussed. In particular, only mechanical constraints are considered, from

which generalized expression for kinematic constraint and dynamic constraint are derived. The kinematic constraint is useful for the machine to spatially arrange the system of rigid bodies and this is the main prerequisite for the machine to simulate the interaction between the entities.

- From an engineering point of view, the meaning or lexicon of a word has two parts: physical model and conceptual model, and a machine can learn these through programmed interface and represent these in its organized memory.

Chapter 4

Construction of Organized Physical Meanings

4.1 Background

Physical meanings play an important role in determining the logics of interactions between the entities. The main focus of this chapter is to model the static interactions between the entities. The static interaction tells us about how one entity can be spatially positioned with respect to another entity or ground (i.e. whether one entity can be positioned on/under another entity). Such capability of the entity is called ‘spatial constraint’. The spatial arrangement of the entity in certain pose with respect to the other entity gives rise to configuration of entities. The configuration of system of rigid bodies and its representation in the graph is called ‘configuration network’.

The following sections discuss the physical constraint networks, in which the configuration network is the main focus of this chapter, along with the rigid body statics model which is the main algorithm that converts entity properties into configuration network, and the algorithm includes many sub-algorithms such as octree representation for the entity, grouping of similarly oriented faces, geometry parameters calculation, intersection testing, and test for stability.

4.2 Physical Constraint Networks

Interactions among entities in the physical world create the physical constraint networks. They are modeled and represented in organized memory. The constraint physics controls the interaction between the entities in order to produce the desired outcome. Because of these interactions, action, behavior, events and episodes are represented in the machine mind. Before any interaction takes place between the entities, the entities should be placed in the space with certain positions and orientations with respect to one another and we call this a configuration of entities. When a system of rigid bodies are placed in space in certain configuration with respect to one another this creates the notion of a *configuration network*. For example, the assembly of car parts for making a car; or the positioning of the parts such as walls, roofs, windows, tables, floors, etc in certain poses with respect to one another creates a configuration network, which can be named as ‘house’. In mechanics, configuration network includes mechanisms (Nortan [79]).

The purpose of the configuration network is to produce an active or passive mechanism in order to produce a desired outcome. As we have seen, if force acts on the entity, it will produce an effect in the form of motion. As we know a configuration network consists of a set of rigid bodies (i.e. components) and each component is placed in a certain pose with respect to the others in order to produce the desired configuration and motion. If the force acts on the first component, the first component’s effect will act on the next component and this component’s effect will act on another and so on. The assumption here is that components are connected. This chain of cause-effect phenomena in the mechanism or configuration network creates the *behavior network*. For example, transmission of force or motion, or conversion of one form of energy into other in the mechanism creates the behavior network, and this behavior network creates the notion of an event. When elements in a set of events interact with one another to produce certain cause-effect phenomena, this creates the notion of an *event network* and the event

network produces the concept of an episode. Football games, and manufacturing activity are examples of episodes. When elements in a set of episodes interact with one another in order to produce certain cause-effect phenomena, this creates the notion of an *episode network*. War, manufacturing, and interaction of different characters in a movie are examples for the episode network. Fig. 2.2 shows the constituents of physical constraint networks.

4.3 Configuration Network

Among the physical constraint networks, the configuration network is the basic element. The spatial arrangement of rigid body A in a certain pose with respect to rigid body B and the relative motion between them can be captured by the kinematic constraint. The parameter values of the kinematic constraint are the useful property for the machine to learn the meaning of words that are related to spatial configurations such as on, above, below, etc. More importantly, only the gravity constraint and mechanical properties are needed to construct the configuration network. When systems of rigid bodies are arranged based on gravity (i.e. attachment of entities based on the geometrical constraints and it should not involve any artificial attachment between the entities), the equilibrium and stability of rigid bodies play an important role. The rigid body statics model captures these aspects. It is important to note that the kinematic constraint captures the two important parameters such as pose and relative motion between the entities and in constructing a configuration network, only pose plays a vital role and relative motion comes into the picture only when a behavior network is constructed between the entities. The construction of the configuration network is the main focus of this thesis.

What the rigid body statics model does is it takes as the input entities properties such as geometrical and mechanical and spatial constraints parameter values (optional), and it calculates the pose for each entity with respect to the world co-

ordinate system. The configuration network is represented by a graph data structure. From the configuration network, a machine can synthesize an image as well as text to convey its understood meaning. Text synthesis from the configuration network is discussed in chapter 5.

4.3.1 Rigid Body Statics Algorithm

Input: Geometrical properties and mechanical properties of the entities and spatial constraint parameter values

Output: Configuration network

Procedure:

1. *Put the physical properties of the entities and constraints in a list*
2. *Initialize the graph and create root node and add the properties of the ground*
3. *Initialize the stack*
4. *Start iterating on the list and put elements on the stack, and check whether the stack sequence is any one of the following forms: ‘the reference entity - constraint values - the target entity’, ‘the reference entity - the target entity’, and ‘the reference entities - constraint values - the target entity’*
5. *If the stack has the sequence ‘the reference entity - constraint values - the target entity’ or ‘the reference entities - constraint values - the target entity’, do the following.*
 - (a) *If the constraint parameter value has $(E_r \cap E_t) = \text{true}$, (i.e. it means that the reference entity intersects the target entity), perform the following procedure*
 - i. *Check the condition: (maximum strength of the reference entity > weight of the target entity)*
 - ii. *If false, Exchange the entities and again check the condition, If this fails, try a different combination of entities using step 4*

- iii. *If graph has no children (i.e. nodes), place the reference entity on the ground at some arbitrary location P_{Er} . The reference entity is placed on the ground based on usage configuration.*
- iv. *If the reference entity exists already in the graph, go to next step*
- v. *Get the constraint parameter values. Based on the constraint information, analyze the empty space, surface patch and bounding box parameters of the reference entity. Then determine the position vector of the target entity (P_{Et}). Then place the target entity at the desired position till it intersects the reference entity.*
- vi. *Determine the contact points of the intersection*
- vii. *Perform the stability test to check whether the configuration is stable or not*
- viii. *If stable, then attach the reference entity to the root node of the graph and attach the target entity to the reference entity node. Then remove the reference entity and the target entity properties, and constraints from the list.*
- ix. *If not stable, try different combination of entities, then go to step 4*
- (b) *If the constraint parameter value has ($E_r \cap E_t = false$), it means that there is no intersection between the reference entity and the target entity.*
 - i. *If the reference entity already exists in the graph, get its bounding box information and its position vector (P_{Er}). If the reference entity does not exist in the graph, place it on the ground, then get its bounding box information and its position vector (P_{Er})*
 - ii. *Get the constraint parameter values. Based on the constraint information, analyze the empty space, surface patch and bounding box parameters of the reference entity. Then determine the position vector of the target entity (P_{Et}). Then place the target entity at the desired location.*

6. *If the stack sequence is ‘the reference entity - the target entity’ , assume that there is intersection between the entities and construct the configuration network using step 5a*
7. *If the list is empty, exit the loop*

4.3.2 Octree

Octree is a spatial data structure, which is used to represent the entity in terms of discrete cubic elements (i.e. voxels). Each node in the octree is subdivided into eight equal parts (i.e. cubes) till certain conditions are satisfied. A leaf node is a node which does not have any subdivision. Each leaf node in the octree contains some vertices of the entity, width and the center of the cube. The entity can be recreated by just retrieving the information from each leaf node in the octree. Octree reduces the number of intersection tests needed during the collision detection (Meagher [80]).

Algorithm for Octree

Input: Vertices of the entity, maximum number of vertices per node and maximum octree depth.

Output: Octree data structure. And each node in the octree contains vertices corresponding to some portion of the entity, cube center and width, boolean variables such as whether the current node is subdivided or not

Procedure:

1. *Using the entity’s vertices, calculate the bounding cube which encloses the vertices. The bounding cube is represented by its center and width. Create the octree list and put the vertices, cube center and width into the list. This octree list is called as root node of the octree.*
2. *Given the input octree node (i.e. it can be root node or any subdivided node), subdivide that node space into eight equal parts and as a result, under the root*

node, eight new octree nodes will be created and it uses the following procedure to create eight new nodes.

- (a) *Initialize eight octree list*
 - (b) *Calculate the center and width for each octree node*
 - (c) *Determine the vertex or vertices belong to each node and put them in the node*
3. *Now, we have subdivided the root node into eight sub-nodes. Recurse each sub-node for further subdivision into eight sub nodes using step 2 till the following conditions are satisfied: (Vertex count of each octree > Maximum vertex per node) and (Current depth < Maximum octree depth)*

Example

Fig. 4.1 shows the octree representation of the ‘child’ object. There are four sub figures and each has the octree representation for a different octree depth and maximum number of vertices per node. They are generated from the KnowNet system and used for construction of the configuration networks. Two important cases can be noticed: (a) for higher value of vertices count, whatever the values of octree depth, the algorithm does not produce the proper discretization of the object into voxels and the reason is that the higher number of vertices count causes the algorithm to stop the subdivision of nodes before it reaches the actual octree depth. (b) keeping the vertices count to zero and octree depth level more than five produces fine discretization.

4.3.3 Intersection Testing

For a machine to undertake the mental simulation of physical world entity interaction, intersection testing should be performed in order to avoid collision between the objects (i.e. penetration of one rigid body into the other). This helps the machine to perform the mental simulation of interactions between the rigid bodies

Figure 4.1: Shows the octree decomposition for the ‘child’ object for different vertices count and octree depth

in its mind and this simulation exactly reflects the interactions of entities in the physical world. Intersection is the most studied topic in computational geometry and it is currently applied in gaming, dynamic simulation of rigid bodies and so on. Currently there are many well known and widely used intersection algorithms for primitive shape intersection to be found in the literature. For the construction of the configuration network, this section considers the following primitive shape intersection testing: oriented box-oriented box intersection (Gottschalk et al [81] and Gamasutra [82]), triangle-triangle intersection (Moller [83]) and ray-triangle intersection (Moller and Trumbore [84]).

Oriented Box-Oriented Box Intersection

Oriented box as its name implies that there is a box or cube which can be in any orientation in the space. The oriented box can be represented by eight vertices or center point, orthonormal basis or axes, and extents such as width, height, depth. For intersection testing, the latter representation is used. The oriented box intersection is based on the separate axis theorem, which states that ‘Two convex polytopes are disjoint if there exists a separating axis orthogonal to a face of either

polytope or orthogonal to an edge from each polytope'. A separating axis is an axis in which the polytopes are projected in order to form the interval.

Figure 4.2: Shows the oriented boxes and formation of a separating axis in two dimensions

Fig. 4.2 shows the oriented boxes and formation of a separating axis in two dimensions. For simplicity of understanding the concept, the two dimensional representation is considered. There are two oriented boxes such as box A and box B and the box A has the center point (P_a), orthonormal basis (A^1, A^2, A^3), where A represents the rotation matrix of 3×3 size, A^1 represents the first column of A, A^2 represents the second column of A, A^3 represents the third column of A, and extents (a_1, a_2, a_3), where a_1 is box width, a_2 is box height and a_3 is box depth. Similarly, oriented box B is represented. The radius of projection of box A onto the separating axis L is given by

$$r_a = a_1|A^1 \cdot L| + a_2|A^2 \cdot L| + a_3|A^3 \cdot L| \quad (4.1)$$

Similarly, the radius of projection of box B onto the separating axis L is given by

$$r_b = b_1|B^1 \cdot L| + b_2|B^2 \cdot L| + b_3|B^3 \cdot L| \quad (4.2)$$

And, the condition to form the separating axis and no intersection between the boxes is given by

$$|T \cdot L| > r_a + r_b \quad (4.3)$$

Where $T = P_a - P_b$ represents the difference between the center point of the boxes. L is the basis vector and there are fifteen possible basis vectors for two

oriented boxes, so the above condition should be checked for fifteen possible basis vectors. L takes the following basis vectors: $A^1, A^2, A^3, B^1, B^2, B^3, A^1 \times B^1, A^1 \times B^2, A^1 \times B^3, A^2 \times B^1, A^2 \times B^2, A^2 \times B^3, A^3 \times B^1, A^3 \times B^2$, and $A^3 \times B^3$.

Algorithm: It is important to note that box B's orthonormal basis should be transformed into box A's coordinate frame.

Input: Center point, orthonormal basis and extents of box A and box B

Output: Boolean variable such as true or false for whether boxes intersect or not

Procedure:

1. Transform the box B's orthonormal basis into A's coordinate frame
2. For each L , do the following:
 - (a) Calculate the radius of projection for box A and box B
 - (b) Calculate $|T \cdot L|$
 - (c) Check the condition $|T \cdot L| > r_a + r_b$, if true, there is no intersection

Triangle-Triangle Intersection

In this, the Moller [83] algorithm for triangle-triangle intersection is discussed. A triangle is represented by three vertices such as V_0, V_1, V_2 . Let us denote the two triangles T_1 and T_2 ; their vertices are V_0^1, V_1^1, V_2^1 and V_0^2, V_1^2, V_2^2 respectively. The plane at which the triangles T_1 and T_2 lie are π_1 and π_2 respectively. The equation of the plane π_2 is given by

$$\begin{aligned}
 \pi_2 : N_2 \cdot X + d_2 &= 0 \\
 N_2 &= (V_1^2 - V_0^2) \times (V_2^2 - V_0^2) \\
 d_2 &= -N_2 \cdot V_0^2
 \end{aligned} \tag{4.4}$$

The signed distances between T_1 and plane π_2 are determined by inserting the vertices of T_1 in plane Equation 4.4. and denoted as

$$d_{v_i^1} = N_2 \cdot V_i^1 + d_2, i = 0, 1, 2. \tag{4.5}$$

If $d_{v_i^1} \neq 0$ for all i , there are no vertices lying in the plane and overlap is rejected. The above test is also done for π_1 and T_2 . These tests help to avoid many of the computations required for some triangle pairs.

The next two tests to be considered are the interval overlapping test and the coplanar test. In the interval overlapping test, the interval for T_1 can be calculated as

$$t_1 = p_{v_0^1} + (p_{v_1^1} - p_{v_0^1}) \frac{d_{v_0^1}}{(d_{v_0^1} - d_{v_1^1})} \quad (4.6)$$

$$t_2 = p_{v_0^1} + (p_{v_2^1} - p_{v_0^1}) \frac{d_{v_0^1}}{(d_{v_0^1} - d_{v_2^1})} \quad (4.7)$$

$$p_{v_i^1} = \begin{cases} V_{ix}^1, & \text{if } |D_x| = \max(|D_x|, |D_y|, |D_z|) \\ V_{iy}^1, & \text{if } |D_y| = \max(|D_x|, |D_y|, |D_z|) \\ V_{iz}^1, & \text{if } |D_z| = \max(|D_x|, |D_y|, |D_z|), i = 0, 1, 2. \end{cases} \quad (4.8)$$

Where $D = (N_1 \times N_2)$, similarly, the interval for T_2 is calculated. These intervals are compared to check whether the triangles intersect. For the coplanar test, a point-in-triangle test for one vertex of T_1 against T_2 and vice versa is performed.

Algorithm: *Input:* Vertices of triangles T_1 and T_2

Output: Boolean variable such as true or false for whether triangles intersect or not

Procedure:

1. Calculate the plane equation for T_2
2. Check for intersection by substituting the vertices of T_1 in the plane equation of T_2
3. Calculate the plane equation for T_1
4. Check for intersection by substituting the vertices of T_2 in the plane equation of T_1 .
5. Calculate the intervals for both the triangles and compare the intervals for intersection.

6. Check whether triangles are coplanar.

Ray-Triangle Intersection

The Moller and Trumbore [84] algorithm for ray and triangle intersection is discussed. A ray is represented by the origin point (O) and the direction (D) and triangles are represented by vertices such as V_0, V_1, V_2 . The parametric equation of the ray is given by

$$R(t) = O + tD \quad (4.9)$$

The parametric plane equation of the triangle is given by

$$T(u, v) = (1 - u - v)V_0 + uV_1 + vV_2 \quad (4.10)$$

where $T(u, v)$ is any point on the triangle, (u, v) are the barycentric coordinates, which must satisfy $u \geq 0, v \geq 0$ and $u + v \leq 1$. When ray intersects the triangle, the target or final point of the ray is equal to a point on the triangle and it is represented by

$$\begin{aligned} R(t) &= T(u, v) \\ O + tD &= (1 - u - v)V_0 + uV_1 + vV_2 \end{aligned} \quad (4.11)$$

Expanding the above equation yields the solution to u, v and t as follows

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix} \quad (4.12)$$

Where $P = (D \times E_2)$, $Q = (T \times E_1)$, $E_1 = V_1 - V_0$, $E_2 = V_2 - V_0$ and $T = O - V_0$.

Algorithm: *Input:* Ray and vertices of triangle T_1

Output: Boolean variable such as true or false for whether ray and triangle intersect or not

Procedure:

1. Calculate and assign the $P \cdot E_1$ to det , if $det < tolerance(0.0001)$, return false (i.e. no intersection)

2. Calculate and assign the $P \cdot T$ to u_1 , if $u_1 < 0$ or $u_1 > det$, return false,
3. Calculate and assign the $Q \cdot D$ to v_1 , if $v_1 < 0$ or $u_1 + v_1 > det$, return false,
else return true

4.3.4 Geometry Parameters Calculation

This section discusses the geometry parameters required for construction of the configuration network. Given as input the faces of an entity, the algorithm has to identify the similarly oriented triangular faces, center of mass, and surface area. These are the parameters the rigid body statics algorithm will use for constructing the configuration network.

Grouping of Similarly Oriented Triangular Faces of an Entity

This is important to place an entity A in a certain pose with respect to entity B. The grouping of similarly oriented triangular faces forms the set of regions or patches. The normal of the entities patches will decide the orientation of entity A with respect to entity B.

Algorithm: *Input:* Faces of an entity

Output: Set of similarly oriented triangular faces

Procedure:

1. Put the faces in the list
2. First, choose any face in the list and name it as f_1
3. Determine the connected faces, given the input face f_1 .
 - (a) For each connected face, calculate the angle between input face normal and the connected face normal
 - (b) Check whether the angle is equal to zero
 - i. If yes, put the connected face in the output list, and do the recursion by taking this face as input and continue step 3

- ii. *If no, go to other connected face for the recursion using step 3 if connected faces exist in the output list or start with a new face, which should not have been in the output list and should not have been iterated for angle calculation before.*

4. *Exit the loop if all the faces are iterated*

Centroid or Center of Mass

The mass of the body is considered to be concentrated on the single point, which can exist either outside the body or in the body and the mass concentration point is called centroid or center of mass (Hobbeler and Fan [85], Dunn [86] and Bovrg [87]). The centroid is a useful property to determine the stability between the rigid bodies. The centroid of the body that consists of N triangular faces with vertices of i^{th} face (x_i, y_i, z_i) is given by

$$C = \frac{\sum_{i=0}^{N-1} A_i R_i}{\sum_{i=0}^{N-1} A_i} \quad (4.13)$$

$$R_i = (x_i + y_i + z_i)/3 \quad (4.14)$$

$$A_i = |(b_i - a_i) \times (c_i - a_i)| \quad (4.15)$$

where C is the center of mass, A_i twice the area of i^{th} triangular face and R_i is the center point of the i^{th} triangular face. The surface area and center point of set of grouped triangular faces can be determined using the above equations.

4.3.5 Transformation Matrix

Another important question arises as to how to calculate the transformation matrix or pose of an entity A with respect to entity B. What we know from the entity is the set of grouped faces and the face normals corresponding to each face and these are the information we have at hand. So, the problem is how to calculate the rotation and translation of the transformation matrix using the above.

Let's denote a body as A which has six patches and one of their normals and centers are n_a and sc_a respectively. Similar to that, body B has normal n_b and center sc_b for one of its grouped faces as in Fig. 4.3.

Figure 4.3: Illustration of rotation matrix and direction vector calculation

Calculation of Rotation Matrix

Here, the rotation matrix is calculated by converting the axis angle form of rotation into a three by three rotation matrix (Dunn [86]). In axis angle rotation, the rotation axis and rotation angle are represented respectively by a vector of three elements and a scalar value in degrees or radians respectively. So, from normals n_a and n_b , the axis angle rotation of body A with respect to body B can be calculated by

$$N = n_a \times n_b \quad (4.16)$$

$$\theta = \cos^{-1}(n_a \cdot n_b) \quad (4.17)$$

$${}^b_aR = [N_x, N_y, N_z, \theta] \quad (4.18)$$

where N is axis of rotation and its components are $[N_x, N_y, N_z]$. θ is the rotation angle. b_aR rotation of body A with respect to body B. The axis angle rotation can

be converted into 3 by 3 matrix by

$${}^b_a\mathbf{R} = \begin{bmatrix} tN_xN_x + c & tN_xN_y - N_zs & tN_xN_z + N_ys \\ tN_xN_y + N_zs & tN_yN_y + c & tN_yN_z - N_xs \\ tN_xN_z - N_ys & tN_yN_z + N_xs & tN_zN_z + c \end{bmatrix} \quad (4.19)$$

where $c = \cos(\theta)$, $s = \sin(\theta)$ and $t = 1 - c$. For the intersection of an oriented box, this rotation matrix is used as the orthonormal basis for the box.

Calculation of Translation

Translation is determined by the surface center points of the bodies. The difference between the center point (sc_a) of the surface of body A and the center point (sc_b) of the surface of body B gives the direction vector. Using the direction vector, the translation of body A with respect to body B is calculated by

$${}^b_a\mathbf{P} = CG_a + (sc_b - sc_a)k \quad (4.20)$$

where ${}^b_a\mathbf{P}$ is translation of body A with respect to body B, CG_a is the initial position or centroid of the body A and k is a real number, which can be decided by intersection testing and the stability test of a rigid body statics algorithm. Combining rotation and translation, the transformation matrix is given by

$${}^b_a\mathbf{T} = \begin{bmatrix} {}^b_a\mathbf{R} & {}^b_a\mathbf{P} \\ 1 & 1 \end{bmatrix} \quad (4.21)$$

4.3.6 Test for Stability

In the physical world, the stability plays an important role in the arrangement of one entity with respect to other entity. This test helps the machine to decide automatically whether a certain configuration of entities is possible or not in the physical world. From physics, it is known that the body mass is concentrated at centroid and because of gravity the body weight (mass x acceleration due to gravity) is acting perpendicular to the ground. The stability test between the body A and body B can be done by projecting the centroid of A on the contact surface between body A and body B and if the projected centroid point lies on the contact surface, the body A is in stability with respect to B. More importantly, the

stability of the bodies A and B is tested by projecting the centroid of bodies A and B altogether on the contact surface between body B and the ground (Leipholz [88] and Stability [89]).

Algorithm

Input: Entities and their contact points, which are determined by intersection test

Output: boolean variables such as true or false to check whether the configuration is in stability

Procedure:

1. *Calculate the centroid of the body A*
2. *Get the contact points between the body A and body B*
3. *Calculate the convex hull using the contact points and convex hull algorithm produces the set of triangular faces that enclose the set of contact points*
4. *The projection of centroid on the contact surface can be formulated as the intersection between ray and triangle. The ray is represented by centroid as its initial point and direction vector perpendicular to the ground as $([0 \ -1 \ 0])$. If ray intersects any of the triangular faces, then body A is in stability with respect to body B.*

The algorithm uses the quick hull algorithm to determine the convex hull for a set of contact points (Barber et al [90]).

4.4 Scene Synthesis

The configuration network captures the spatial relationship between the entities and each entity has its geometrical properties such as vertices and color or material properties. Using the OpenGL [91] renderer, the configuration network is visualized in order to convert meaning of word or sentences into images or scenes.

4.4.1 Algorithm

Input: Configuration network

Output: Scene or image

Procedure:

1. *Get all the nodes from the configuration network (i.e. graph)*
2. *From each node, retrieve the vertex, vertex normal, color or material properties and transformation matrix*
3. *Render the above in OpenGL using `glVertex3f`, `glcolor3f`, `glLoadMatrix`, etc*

4.5 Summary

In summary,

- Static interaction of entities based on the gravity creates the configuration network
- Physical meanings assist the machine to learn words correspond to the entities and their interactions.
- The machine can understand the interaction of entities with help of rigid body statics algorithm.
- To construct the configuration network, geometrical properties, mechanical properties and spatial constraint parameters are considered as main elements. The configuration network helps the machine to learn the spatial concepts.
- The machine can synthesize the image and text from the configuration network.

Chapter 5

Construction of Organized Conceptual Meanings

Sociable machines should communicate their understood meanings to human beings in the form of natural languages such as English. This chapter discusses how to synthesize the sentences or text given the input configuration network and conceptual models. Text that is generated from the configuration network will be meaningful and that which is generated only from the conceptual models will be a grammatically correct but it will not be meaningful. The following sections discuss the conceptual constraint networks, constraints for phrase formation, constraints for sentence formation, and synthesis of sentences from conceptual models and physical constraint networks. This chapter focuses on the construction of the concept network or synthesis of simple sentences using conceptual models and configuration network.

5.1 Conceptual Constraint Networks

Organizing the set of words based on the grammar rules and logics of the physical world in order to form the meaningful texts is termed a ‘conceptual constraint network’. The constituents of a conceptual constraint network are shown in Fig. 2.2. The concept network, topic network, belief network and value network are the main elements. Arrangement of a set of symbols based on the grammar rules and logics of the physical world in order to form a meaningful sentence(s) is called a concept

network. The topic network refers to the arrangement of a set of coherent concepts based on certain relationships (i.e. logics). A value network can be a concept network or a topic network, which helps the machine to differentiate between good and bad activities or behaviors in the physical world. Teaching the activities to the machine such as giving the respect to elderly people, the fact that one should not rob banks, etc are the examples of value networks. A belief network is also a concept network or a topic network, which makes the machine to follow or accept certain customs, which can be religious or imposed by human beings or other agents in the real world.

Among all the networks, the concept network is the basic one. As we know, conceptual models have properties and constraints and the matching the constraints of one symbol with other symbols creates the syntactically correct sentences. The following sections discuss the constraints for phrase formation, the constraints for simple sentence formation and the algorithm for constructing the concept network.

5.2 Constraints for Phrase Formation

A phrase is constructed by grouping one or more symbols with the help of phrase formation constraints. Phrases are combined together with help of sentence formation constraints in order to form a sentence. They are divided into eight types: Noun Phrase (NP), Verb Phrase (VP), Adjective Phrase (ADJP), Adverb Phrase (ADVP), Noun Phrase subject (NPs), Noun Phrase object (NPo), Prepositional Phrase (PP), and infinite Verb Phrase (VPinf). Fig. 5.1 shows the phrase formation constraints.

The ordered sequence of lexical properties forms the rules for the phrase. For example, a Noun Phrase (NP) can be constructed from the sequence of following properties: ‘determiner + adjective + noun’, ‘determiner + noun’, ‘determiner + noun + noun’, ‘noun + noun’, and ‘noun’. The empty box on either side of the middle box tells us that the word’s lexical property can itself form a noun phrase.

Figure 5.1: Phrase formation constraints for English. The ordered sequence of lexical properties forms the constraints for the phrase.

For example, in noun phrase construction, only ‘noun’ can be used to form the phrase (Allen [50]).

5.2.1 Algorithm for Phrase Construction

Input: Set of words and their corresponding conceptual models.

Output: List of phrases.

Procedure:

1. Tokenize the set of words and attach the conceptual model to each token
2. Put the words and their conceptual models in the list
3. Get the first word and push into the stack
4. Get the next word and push into the stack
5. Check whether the constraints for the words in the stack match
 - (a) If yes, go to step 4

- (b) *If no, pop the stack and check whether the sequence of the words properties in the stack matches the phrase rules.*
- i. *If yes, form the phrase and empty the stack*
 - ii. *If no, go to step 4*
6. *If the list is empty, exit the loop*

Figure 5.2: Phrase construction for a set of symbols 'cat', 'the', 'good'. Constraint matching plays an important role in phrase formation

5.2.2 Phrase Construction Example

Constraint matching is nothing but checking whether the property of word A equals a pre-constraint (i.e. property) of the word B and the property of the word B equals the post-constraint of the word A. This is the main process which is used in phrase construction algorithm. Fig. 5.2 shows the construction of the phrase for a set of symbols 'cat', 'the', 'good'.

First, the words are tokenized and corresponding conceptual model is attached to each word. Then, the constraints are verified between the words ‘cat’, ‘the’. In this process, the property of ‘the’ matches with the pre-constraint of the ‘cat’ and the property of ‘cat’ matches with the post-constraint of ‘the’. So, the order ‘the cat’ is produced and put it into the stack. Then, algorithm gets the next word ‘good’ from the list and constraints are matched between ‘good’ and ‘the cat’. As a result, the ‘good’ is placed between ‘the’ and ‘cat’ and finally the phrase ‘the good cat’ is formed.

5.3 Constraints for Sentence Formation

Like words, phrases also have pre-constraints and post constraints. Combining the phrases through the constraint matching process forms a sentence. The sentence formation constraints consist of phrase name (i.e. the good cat), phrase property (i.e. noun phrase), pre-constraints and post-constraints. Constraints are also referred to phrase properties that can be matched with other phrase property. Different types of sentence formation constraints are shown in Fig. 5.3. For example, Noun Phrase (NP) has pre-constraints such as Noun Phrase (NP), Prepositional Phrase (PP), and Verb Phrase (VP) and post-constraints such as Noun Phrase (NP), Prepositional Phrase (PP), Verb Phrase (VP), Adjective Phrase (ADJP) and Adverb Phrase (ADVP). These constraints can produce simple sentence(s) from a set of phrases.

5.4 Construction of Only Grammatically Correct Concept Network from Conceptual Models

In this section, the construction of the concept network given the input conceptual models, the phrase formation constraints and the sentence formation constraints is discussed. The matching of properties and constraints between the words creates only grammatically correct concept network. Since the sentence is synthesized from

Figure 5.3: Shows constraints for sentence formation.

the conceptual models, the generated sentence will be grammatically correct but it will not be meaningful. For now, the purpose is just to show that machine has the capability to construct the sentences from a set of symbols. One good application for this is rearranging the words in order to form the correct sentence, which is normally found in primary school level English language learning.

5.4.1 Algorithm

Input: Set of words and their corresponding conceptual models.

Output: List of sentences.

Procedure:

1. Tokenize the set of words and attach the conceptual model to each token
2. Put the words and their conceptual models into the list
3. Initialize $i=0$
4. Increment i till it equals to size of the list

5. *Get the word from the list[i] and pass the list and the i^{th} element to the phrase formation algorithm in section. 5.2.1 and put the phrases in the new list and call this as 'phrase list'.*
6. *Initialize $k=0$*
7. *Increment k by 1*
8. *Get k^{th} element from the phrase list and push into the stack*
9. *Get the next element and push into the stack*
10. *Check whether the constraints for the phrases in the stack match*
 - (a) *If yes, go to step 9 and continue till the phrase list is empty*
 - (b) *If no, pop off the stack and go to step 9*
11. *Put the phrase sequence in the output list. If the phrase list is not empty, go to step 7. If it is empty, go to step 4*
12. *Filter or remove similar sequences or sentences in the output list and exit the program*

5.4.2 Example for Concept Network Construction

Words such as 'more', 'cakes', 'I', 'some', 'want' are considered for this example. In these, words such as 'more' and 'some' have more than one lexical property and they are 'determiner' and 'adverb'. Because of these properties, the machine can generate the list of possible sentences. Fig. 5.4 shows processes in the construction of the sentence *I want more cakes some* from the set of words. Further more, the algorithm can produce the sentences: *I want some cakes more*, *I want some more cakes*, *I want cakes some more*, and *I want cakes more some*.

Figure 5.4: Shows the processes in construction of the concept network ‘I want more cakes some’ from words such as ‘more’, ‘cakes’, ‘I’, ‘some’, ‘want’.

5.5 Construction of Meaningful Concept Network from Configuration Network and Conceptual Models

As discussed before, a machine can produce meaningful sentences or concept networks given the input configuration network and conceptual models. The configuration network is nothing but the spatial arrangement of rigid bodies based on the gravity with the help of the rigid body statics model. What we have seen before from the last example of constructing the concept network from a set of words is that (a) word may have more than one lexical property, (b) construction of more than one grammatical sentence. Obviously, the machine is not expected to do the above, but rather we are interested in having the machine to construct the meaningful sentences. To address these issues, the configuration network and conceptual models are used as the input for the construction of the concept network.

The configuration network captures the entities and spatial relations between

them. The spatial relations are represented in the configuration network as a transformation matrix, which consists of rotation and translation. The transformation matrix between the entity A and entity B, intersection between the entities, volume ratio, etc decides the spatial relation between them. From the above parameters values, the corresponding conceptual label or word is determined. The subject, verb, object, and modifiers and their dependencies are determined by the configuration network and detailed discussion is given in Chapter 6.

Here, the words sequence can be determined using the configuration network and the purpose of the phrase formation and sentence formation constraints are to just check whether the sequence is grammatically correct. Since the configuration network is static in nature and it has the information about entities' spatial relations, the word 'is' is used to denote the verb during the sentence synthesis.

5.5.1 Algorithm

Input: Configuration network.

Output: Meaningful and grammatically sentence(s).

Procedure:

1. *Get the configuration network*
2. *Traverse the graph from leaf node to the root node and any nodes between the depth $i+1$ and depth i form the subject and object respectively and in contrast, any nodes at the breadth level can act as the subject as well as the object.*
3. *Take any two nodes between the depth $i+1$ and depth i and the node name at $i+1$ is taken as 'subject' and node name at i is taken as 'object' and add verb 'is' between the subject and object. Then assign the determiner 'the' to subject and object if they do not belong to the lexical property 'name'.*
4. *Identify the spatial constraint parameter values between the the subject and object and get the appropriate name to that constraint.*

5. *Get the words corresponding conceptual model*
6. *Check the sequence by matching the constraints between the words and put the sentence in the list.*
7. *If traversed to root node, exit the loop.*

5.5.2 Example

Figure 5.5: Illustration of concept network construction from configuration network, which helps to determine the subject, verb, object, and modifiers.

Let's assume that the configuration network has the situation such that there is a table on the ground and there is a ball, which is on the table as in Fig. 5.5. The configuration network is represented by a graph data structure. The root node has the world coordinate system and 'ground' represents the root node. All other entities' transformation matrices are relative to the ground coordinate system. The leaf node of the graph has the entity 'ball' and the predecessor to the leaf node has the entity 'table'. The modeling parameters of the spatial constraint between 'table' and 'ball' will be $\{S_{top} = true; E_r \cap E_t = true; VD_{O-ce_r} \geq 1; ST = true; E_t \cup E_r = true\}$ and the word or concept which corresponds to this constraint is 'on'. This

word is inserted before the object or reference entity name. Any node having geometrical properties in the configuration network is recognized as an ‘entity’. The concept network algorithm forms the sequence ‘the ball is on the table’. The conceptual model for each word in the sequence is retrieved and the constraints are verified between the words to check whether the sequence is grammatically correct.

5.6 Summary

In summary,

- Conceptual constraint networks are nothing but descriptions of understood meaning or logics (i.e. meanings in physical constraint networks) in terms of sentences or text by a machine is called as text synthesis.
- A concept network can be generated given the input either conceptual models or configuration network and conceptual model. The sentences that are produced from configuration networks and conceptual models are meaningful.
- Sentence synthesis with conceptual models can be seen as a constraint matching process.
- The configuration network is the main component for synthesizing the sentence(s) and it helps the machine to determine subject, verb, object, and modifiers and their dependencies, and the conceptual constraints are used to check whether the word sequence is grammatically correct.

Chapter 6

Implementation, Results and Discussion

In previous chapters, modeling of elementary meanings, rigid body statics algorithm for constructing the configuration network, text synthesis from the configuration network, and algorithm for scene synthesis have been discussed. In this chapter, implementation of the above algorithms, and the importance of the entity's properties and its static interactions for learning and understanding the spatial language by the machine are discussed. The implemented spatial language learning and understanding system is named as 'KnowNet'. KnowNet has been implemented in Java and OpenGL. The following sections discuss KnowNet's architecture, and KnowNet's graphical user interface (GUI), which consists of a programmed interface, which supports the learning of words through entities properties and their interactions, and spatial language understanding and generation component. Following that, the chapter discusses experiments such as learning of words through the entities properties, learning of usage configuration of entities, learning of spatial concepts (i.e. spatial words) through modeling and learning the spatial constraints, testing of learned spatial concepts, generation of sentences, and understanding of sentences such as compound sentences, complex prepositional phrase sentences and the sentences that are violating the truthfulness of the physical world.

6.1 KnowNet Architecture

Fig. 6.1 shows the architecture of KnowNet. It consists of a programmed interface to acquire or learn elementary meanings, organized memory, and mental processes such as analysis and synthesis. First, entities and their constraints are learned through the programmed interface and then a name or word is assigned to each entity and each of its constraint. Then lexical properties and constraints for each word are acquired. Finally the properties and constraints of entities as well as words are stored in the organized memory.

Text is acquired through the text acquisition system, which can be either speech to text conversion or keyboard-based. Then the input text is tokenized in order to form tokens. Then phrases such as noun phrases, verb phrases, adjective phrases and prepositional phrases are identified from the tokens. A physical model for each phrase is retrieved from the organized memory. Physical models are combined and organized by the rigid body statics model in order to construct the configuration network, from which scene and text are synthesized. In scene synthesis, information about the configuration network is given to the OpenGL renderer in order to create an image. In the text synthesis process, subject, object, verb and name of the spatial position are determined from the configuration network and then using the conceptual models syntax for phrases and sentences are verified. Finally, the configuration network and the synthesized sentence(s) are stored in the organized memory.

6.2 KnowNet Graphical User Interface (GUI)

Fig. 6.2 shows the GUI of KnowNet. KnowNet contains a display pane, a programmed interface, a text understanding pane and a synthesized text display pane. The display pane supported by OpenGL is to show the rendered scene from the configuration network. In the understanding pane, a user can input the sentences as well as the entities for the understanding. If the input sentences are mean-

Figure 6.1: KnowNet architecture, in which physical and conceptual models are learned through programmed interface by real time interaction with the human master. Physical models are combined and organized by the rigid body statics model in order to construct the configuration network, from which scene and text are synthesized

ingful (i.e. follows the truthfulness of the physical world), KnowNet creates the configuration network and shows the scene of it in the display pane. If the input sentences are non-meaningful, the modified configuration will be constructed and from it, modified text and modified scene will be synthesized. The synthesized text will be displayed in the text display pane. If the entities are given as input to the understanding, KnowNet constructs different possible configuration networks from which sentences and scenes are synthesized.

6.2.1 Programmed Interface

The main purpose of the programmed interface is to create the interface between the user and KnowNet. It is used to acquire the properties of the entities, to acquire the conceptual properties and constraints, to learn the usage configuration of the entities, to learn the spatial concepts through interaction between the entities and to test the learned spatial concepts. The components of the programmed interface

Figure 6.2: KnowNet Graphical User Interface (GUI), which contains display pane, programmed interface, understanding pane and synthesized text display pane. Programmed interface is used to acquire the properties and constraints of entities and words.

is shown in Fig. 6.2.

6.3 Learning of Entities and Their Associated Words

Learning of the entity is the primary step for learning and understanding the spatial language. First, the entity is modeled by the properties. Then the properties are assigned with certain values. The process of assigning values to the properties is called ‘entity learning’. Then the entity’s properties are assigned with a name or word. This association of the word with the entity is called ‘word learning’.

Fig. 6.3 shows the acquisition of entities properties through programmed interface and their associated words. The machine should acquire the entities before it

starts to learn the usage configuration and the spatial concepts.

Figure 6.3: Learning of an entity and its associated word

In the acquisition process, first the entity's properties such as geometrical and mechanical are acquired. The geometrical properties are acquired through a 3DS file, which has the shape and color information. The mechanical properties such as weight and maximum strength are acquired. The capacity of the entity to withstand the weight of the other entities is called the maximum strength. If the external weight exceeds the maximum strength, the entity will undergo deformation. After the entity is learned, a name or word is assigned to it (i.e. the entity). For that word, the corresponding lexical property and constraints are acquired. The lexical property of the word is nothing but its part of speech tag such as noun, verb, adjective, preposition, etc. The constraints are lexical constraints, phrase formation constraints and sentence formation constraints. The properties and constraints of the word are referred to conceptual models. *Sixty entities have been learned through the programmed interface.*

6.3.1 Example of Learning an Entity - Chesspawn

The entity 'chesspawn' is considered for the learning. The geometrical properties are loaded through the 3DS file. The mechanical properties are assigned with the following values: weight - 0.1 kg and maximum strength - 50kg.

Then the entity is assigned with the word 'chesspawn' and its lexical property

is assigned with ‘noun’. The lexical constraints, phrase formation constraints and sentence formation constraints as shown in Chapter 5 are input by clicking their corresponding buttons. Finally, the ‘store’ button is pressed to store the entity’s properties and conceptual models.

6.4 Learning of Entity’s Usage Configuration

Spatially positioning the entity with respect to the other entity or the ground is termed the configuration of the entity. Normally the entity can have many configurations. These configurations are generated by the rigid body statics model using entity’s properties. The configuration, which people use for a specific purpose is termed the usage configuration. The usage configuration can be learned by either looking at the configuration of a similar entity in the physical world through sensors or by selecting the configuration of the entity among the many entered through the programmed interface by the human agent.

The following procedure is used to generate the different stable configurations of the entity.

- Take the properties of the entity and the ground.
- The ground is assumed to have infinite strength and surface area.
- Using the geometrical analysis, similarly oriented surface normals are identified and the normals form the patch (i.e. subregion in the outer surface of the entity).
- The above information is fed into the rigid body statics model. First the ground is placed. Then the entity is rested on the ground using its patch in order to get the different possible configurations. Then the stability algorithm filters out the unstable configurations. Finally, the similar configurations are removed.

6.4.1 Graphical User Interface for Learning the Usage Configuration

Fig. 6.4 shows the graphical user interface for learning the usage configuration of the entity. The usage configuration of the entity ‘mug’ is shown in the display pane. The ground color is taken as white.

In the GUI, the user first has to select the entity for learning. There are two options such as yes and no given for the stability test. If ‘yes’ is selected, KnowNet generates the possible stable configurations. Then the user is invoked to select the ‘usage configuration’.

Figure 6.4: Graphical user interface for learning the usage configuration of the entity

Fig. 6.5 shows the different stable configurations of the entity ‘mug’.

Figure 6.5: Different stable configurations of the entity ‘mug’. Configuration number three is considered as the usage configuration.

Fig. 6.6 shows the usage configuration of different entities.

6.5 Learning of Entities Static Interactions and Their Associated Concepts

Static interaction between the entities plays an important role in learning the spatial constraints. Spatial constraints involve capability of the entity to be positioned with respect to other entities. In other words, the number of possible spatial relationships that one entity has with an other entity. Modeling and learning the spatial constraints are more important for learning the spatial concepts. The main difference between the kinematic constraint and spatial constraint is that the former deals with dynamic interaction between the entities and latter deals with static interaction between the entities. The rigid body statics model is used to organize or spatially position the entity with respect to other entities.

Figure 6.6: Usage configuration of different entities.

Figure 6.7: Usage configuration of different entities.

Flat Monitor

Fly

Warehouse

Helmet

Hospital Bed

Plate

Knife

Paper

Shoe

Flat plate

Microscope

Table Lamp

Jeep

Monitor

Pear

Figure 6.8: Usage configuration of different entities.

Figure 6.9: Usage configuration of different entities.

6.5.1 Modeling Parameters for Spatial Constraints

The bounding boxes of the entities, distance between the entities, intersection between the entities, etc play the important role in modeling the spatial constraints. The parameters of the spatial constraints are given as input to the rigid body statics model. The following parameters are used to model the spatial constraints. Here, the following axis convention is used. Y and -Y axes are vertically above and below, X and -X are right and left and Z and -Z are front and back.

E_t - Target entity

E_r - Reference entity

P_o - Position vector of the observer $[X_o, Y_o, Z_o]^T$

W - World coordinate system $[X, Y, Z]^T$

BO_r - Bounding box of the reference entity $[L_r, W_r, H_r, CE_r]$, where L_r is the length of the bounding box, W_r is the width of the bounding box, H_r is the height of the bounding box and CE_r is the centroid of the bounding box $[X_{CE_r}, Y_{CE_r}, Z_{CE_r}]^T$.

BO_t - Bounding box of the target entity $[L_t, W_t, H_t, CE_t]$, where L_t is the length of the bounding box, W_t is the width of the bounding box, H_t is the height of the bounding box and CE_t is the centroid of the bounding box $[X_{CE_t}, Y_{CE_t}, Z_{CE_t}]^T$.

S_t - Surface area of the target entity's patch and this patch is got from the usage configuration (i.e. contact area between the entity and the ground).

B_t - Bounds of the target entity's patch $[l_t, w_t, ce_t]$, where l_t is the length of the patch, w_t is the width of the patch and ce_t is the centroid of the patch $[X_{ce_t}, Y_{ce_t}, Z_{ce_t}]^T$

S_r - Surface area of the reference entity's patch

B_r - Bounds of the reference entity's patch $[l_r, w_r, ce_r]$, where l_r is the length of the patch, w_r is the width of the patch and ce_r is the centroid of the patch $[X_{ce_r}, Y_{ce_r}, Z_{ce_r}]^T$

P_{Er} - Position vector of the reference entity $[X_{Er}, Y_{Er}, Z_{Er}]^T$

P_{Et} - Position vector of the target entity $[X_{Et}, Y_{Et}, Z_{Et}]^T$

N_{Er} - Surface normal of the reference entity along X, Y, Z, -X, -Y, and -Z directions $[N_{Er(x)}, N_{Er(y)}, N_{Er(z)}, N_{Er(-x)}, N_{Er(-y)}, N_{Er(-z)}]$

N_{Et} - Surface normal of the target entity along X, Y, Z, -X, -Y, and -Z directions $[N_{Et(x)}, N_{Et(y)}, N_{Et(z)}, N_{Et(-x)}, N_{Et(-y)}, N_{Et(-z)}]$

C_E - Centroid of the position vectors of the entities.

V_{Er} - Volume of the bounding box of the reference entity.

V_{Et} - Volume of the bounding box of the target entity.

d - Difference between the position vectors of the reference entity and target entity $(P_{Er} - P_{Et}) \Rightarrow [X_d, Y_d, Z_d]^T$.

d_t - Difference between the position vectors of the target entity and reference entity $(P_{Et} - P_{Er}) \Rightarrow [X_{dt}, Y_{dt}, Z_{dt}]^T$.

D - Distance between the entities $\sqrt{X_d^2 + Y_d^2 + Z_d^2}$

D_f - Distance between the faces of the bounding boxes.

D_{vf} - Vertical distance between the faces of the bounding boxes along Y direction with respect to the reference entity.

D_{-vf} - Vertical distance between the faces of the bounding boxes along -Y direction with respect to the reference entity.

VD_{O-ce_r} - Vertical distance between the observer and the centroid of the patch of the reference entity along the positive y axis.

$E_r \cap E_t$ - Intersection between the reference entity and the target entity

$R_t \cap E_r$ - Intersection between the ray of the target entity and the reference entity. R_t has six components along X, Y, Z, -X, -Y, and -Z directions $[R_{t(x)}, R_{t(y)}, R_{t(z)}, R_{t(-x)}, R_{t(-y)}, R_{t(-z)}]$

$E_t \square E_r$ - Whether the target entity and the reference entity are in the same plane.

$O \square E_r$ - Whether the observer and the reference entity are in the same plane.

S_{top} - Whether the patch of the reference entity is a top surface.

$E_t \diamond E_r$ - Whether the target entity is completely enclosed by the reference entity. To satisfy this, the following conditions should be satisfied: $\{R_{t(y)} \cap E_r =$

$true(must); R_{t(x)} \cap E_r = true; R_{t(-x)} \cap E_r = true; R_{t(z)} \cap E_r = true; R_{t(-z)} \cap E_r = true; \}$. Among the rays in X, -X, Z, and -Z, the intersection of any three rays with the entity should be true.

$E_t \triangle E_r$ - Whether the target entity is partly enclosed by the reference entity. To satisfy this, at least two rays should intersect the reference entity.

$E_t \cup E_r$ - Whether the target entity is rested on the reference entity based on the gravity.

$E_t \uplus E_r$ - Whether the target entity is rested on the reference entity by the attachment.

$E_{(space - below)}$ - Whether the empty space exists between the ground and the reference entity's immediate part

$E_{(space - between)}$ - Whether the empty space exists between the parts of the reference entity.

$E_{(space - inside)}$ - Whether the empty space exists inside the reference entity.

$E_{(space - outward)}$ - Whether the empty space exists in the reference entity in Y direction (i.e. vertically up) of the surface normal (N_{Ey})

$B_{(E-space-below)}$ - Bounding box of the empty space ($E_{(space - below)}$).

$B_{(E-space-between)}$ - Bounding box of the empty space ($E_{(space - between)}$).

$B_{(E-space-inside)}$ - Bounding box of the empty space ($E_{(space - inside)}$).

$B_{(E-space-outward)}$ - Bounding box of the empty space ($E_{(space - outward)}$).

n - Number of reference entities.

ST - Whether the stability exists between the target entity and the reference entity.

6.5.2 Modeling and Learning of Spatial Constraints and Their Associated Concepts

Using the above parameters, the spatial constraints are modeled and learned. Assigning the certain values to the constraints parameters is called learning of the spatial constraints, then which is associated to the certain word or concept. The

following section discusses how the concept ‘*on*’ is learned through the programmed interface with help of the interaction between the entities. Besides that the learning of the spatial concepts such as *in*, *over*, *under*, *above*, *below*, *inside*, *outside*, *behind*, *in front of*, *back*, *front*, *left*, *right*, *beside*, *between*, *among*, *near* and *far* are discussed.

Graphical User Interface for Learning the Spatial Concepts Using the spatial constraints parameters, the interaction or the spatial constraint of ‘one entity is on top of the other entity’ is modeled by

$$\{S_{top}; E_r \cap E_t; VD_{O-cc_r}; ST; E_t \cup E_r\}$$

The values of the parameters are obtained by the interaction between the entities.

Fig. 6.10 shows the graphical user interface for learning the spatial concepts. First, the user is prompted to select the entities. The spatial configuration between the entities is constructed in two modes: manual mode and autonomous mode. In manual mode, using the translation and rotation values the desired configuration is constructed. In autonomous mode, a number of possible configurations will be generated and the user can select the desired configuration.

Once the desired configuration is constructed, the values of the constraints parameters are displayed in the text pane by pressing the ‘Display Values’ button. Then, the user is asked to give the conceptual name or word to those parameters. The conceptual word is called ‘spatial concept’. Then, the part of speech tag is assigned to the word. Finally, pressing ‘Store’ button stores the spatial concept.

In this example, the reference entity is taken as ‘table’ and the target entity as ‘book’. The observer is assumed to be small ‘box’. The ground is assumed to be infinite surface area and infinite strength, and its color is white. In the list box, the entities ‘table’, ‘book’ and ‘observer’ are selected. The desired configuration is constructed by the manual mode. Then pressing the ‘Display Values’ button shows the values for the parameters. The displayed values are as follows:

Figure 6.10: The graphical user interface for learning the spatial concept

$$\{S_{top} = true; E_r \cap E_t = true; VD_{O-ce_r} = 67cm; ST = true; E_t \cup E_r = true\}$$

The vertical distance between the observer and the reference entity is 67cm. The units are arbitrary and the user can choose any units such as millimeter(mm), feet(ft), meter(m), etc. The concept is named as ‘on’ and its part of speech is assigned with ‘Preposition’. The above to be called ‘on’ the minimum vertical distance value should be greater than or equal to one.

Spatial Constraint - I for ‘On’: Based on the above, the constraint parameter values for the spatial concept ‘on’ is written as

$$\{S_{top} = true; E_r \cap E_t = true; VD_{O-ce_r} \geq 1; ST = true; E_t \cup E_r = true\} \implies \text{On}$$

Spatial Constraint - II for ‘On’: In the last example, the configuration of the entities is such that the target entity is placed on the top of the reference entity’s patch, which should be the top surface of the entity. The target entity is

rested on the reference entity based on the gravity.

In another configuration, there are plain surfaces in the reference entity, whose surface normals may be in X, -X, Z, -Z and -Y directions. In that surface, the target entity is placed on the surface by some attachment. Fig. 6.11 shows the configuration when the photo is attached to a wall of the house. The procedure mentioned above is applied here to learn the spatial constraint. The constraint parameter values for the spatial concept ‘on’ is written as

$$\{V_{Et}/V_{Er} \leq (3/4); S_r/S_t \geq 1; E_r \cap E_t = true; VD_{O-ce_r} \geq 1; E_t \uplus E_r = true; N_{Er} \in N_{Er(x)}|N_{Er(z)}|N_{Er(-x)}|N_{Er(-y)}|N_{Er(-z)}\}; \implies On$$

Figure 6.11: Spatial configuration of spatial constraint-II for ‘On’.

Spatial Constraint - III for ‘On’: Fig. 6.12 shows the configuration for ‘On’. In this case, the configuration of the entities is such that the target entity is placed on the top of the reference entity’s patch, which should be the top surface of the entity. Here, the vertical distance between the observer and the centroid of the reference entity’s patch is zero (i.e. $VD_{O-ce_r} = 0$). In this example, the ground is taken as the reference entity and the car is taken as the target entity. The spatial constraint for ‘On’ is written as

$$\{(S_{top} = true; S_t/S_r \leq 0.8; E_r \cap E_t = true; VD_{O-ce_r} = 0; ST = true; E_t \cup E_r = true)\} \implies On$$

Figure 6.12: Spatial configuration of spatial constraint-III for ‘On’.

Spatial Constraint - I for ‘In’: Fig. 6.13 shows the configurations for ‘In’. The first configuration shows that there is a bowl and the burger is placed in it. In this case, the reference entity is the bowl, which has the empty space in the Y direction of the surface normals. In the second configuration there is a rack and the book is placed in that. In this case, there is an empty space that exists between the entities. The spatial constraint for the above configurations can be written as

$$\{V_{Et}/V_{Er} \leq 1; S_t/S_r \leq 1; E_r \cap E_t = true; E_t \cup E_r = true; (E_{(space-between)} = true \ \& \ R_t(y) \cap E_r = true) | (E_{(space-outward)} = true \ \& \ E_t \triangle E_r = true); ST = true\}; \implies In$$

Figure 6.13: Spatial configurations of spatial constraint-I for ‘In’.

Spatial Constraint - I for ‘Over’: Fig. 6.14 shows the configuration for ‘Over’. This configuration comes when the target entity is vertically higher than the reference entity. The rays of the target entity should not intersect the reference entity. In this example, there is a table in the garage and the ceiling fan is attached to the ceiling. Here, the reference entity is the table and the target entity is the ceiling fan. The spatial constraint for ‘Over’ is given by

$$\{D_{vf} \geq 1; |Y_{dt}| > |X_{dt}|; |Y_{dt}| > |Z_{dt}|; E_r \cap E_t = false; R_t \cap E_r = false\} \implies Over$$

Figure 6.14: Spatial configuration of spatial constraint-I for ‘Over’.

Spatial Constraint - II for ‘Over’: Fig. 6.15 shows the configuration for ‘Over’. In this case, there is a table and on top of it there is a paper. The paper covers the table’s top surface fully. Here, the parameter S_r/S_t plays the important role. Here the reference entity is the table and the target entity is the paper. The spatial constraint for ‘Over’ is defined as

$$\{V_{Et}/V_{Er} \leq (3/4); S_r/S_t \approx 1; S_{top} = true; E_r \cap E_t = true; ST = true\} \implies Over$$

Figure 6.15: Spatial configuration of spatial constraint-II for ‘Over’.

Spatial Constraint - I for ‘Under’: Fig. 6.16 shows the configurations of spatial constraint-I for ‘Under’. In the first configuration, there is a plane tree on the ground and there is a ball, which is close to it. In the other configuration, there is a pole in the ground and the antenna is close to the pole. The spatial constraint for ‘Under’ is defined as

$$\{V_{Et}/V_{Er} \leq (1/4); H_t/H_r \leq (1/4); D_f \leq (1/4) \times H_r; E_t \sqcap E_r = true; R_{t(y)} \cap E_r = true(optional)\} \implies Under$$

Spatial Constraint - II for ‘Under’: Fig. 6.17 shows the configuration of spatial constraint-II for ‘Under’. In this case, there is a table on the ground. There is a ball, which is under the table. The empty space between the ground and the reference entity’s immediate part is the key parameter. The spatial constraint-II for ‘Under’ is defined as

$$\{E_{(space-below)} = true; R_{t(y)} \cap E_r = true; BO_t \leq B_{(E-space-below)}\} \implies Under$$

If the empty space exists between the entity’s parts, the above constraint becomes ‘In’.

Figure 6.16: Spatial configurations of spatial constraint-I for ‘Under’.

Figure 6.17: Spatial configurations of spatial constraint-II for ‘Under’.

Spatial Constraint - III for ‘Under’: This spatial constraint is opposite to the spatial constraint-I of ‘Over’. The vertical distance along the -Y direction is

the important parameter for modeling this spatial constraint. Fig. 6.18 shows the configuration of spatial constraint-III for ‘Under’. In this case, there is a table and on top of it there is a bottle. The table is on the ground. There is a bucket, which is on the ground and close to the table. Here the reference entity is the bottle and the target entity is the bucket. The spatial constraint between the bottle and the bucket is written as

$$\{D_{-vf} \geq 1; |Y_{dt}| > |X_{dt}|; |Y_{dt}| > |Z_{dt}|; E_r \cap E_t = false; R_t \cap E_r = false\} \\ \implies Under$$

Figure 6.18: Spatial configuration of spatial constraint-III for ‘Under’.

Spatial Constraint - I for ‘Above’: Fig. 6.19 shows the configuration for ‘Above’. This configuration comes when the target entity is vertically higher than the reference entity. The rays of the target entity should intersect the reference entity. In this example, there is a table in the garage and the ceiling fan is attached to the ceiling. Here, the reference entity is the table and the target entity is the ceiling fan. The spatial constraint for ‘Above’ is given by

$$\{D_{vf} \geq 1; E_r \cap E_t = false; R_t \cap E_r = true\} \implies Above$$

Figure 6.19: Spatial configuration of spatial constraint-I for ‘Above’.

Spatial Constraint - I for ‘Below’: This spatial constraint is opposite to the spatial constraint-I of ‘Above’. The vertical distance along the -Y direction is the important parameter for modeling this spatial constraint. The rays of the target entity should intersect the reference entity. Fig. 6.20 shows the configuration of spatial constraint-I for ‘Below’. In this case, there is a table and on top of it there is a bottle. The table is on the ground. There is a bucket, which is on the ground and under the table. Here the reference entity is the bottle and the target entity is the bucket. The spatial constraint between the bottle and the bucket is written as

$$\{D_{-vf} \geq 1; E_r \cap E_t = false; R_t \cap E_r = true\} \implies Below$$

Spatial Constraint - I for ‘Inside’: Fig. 6.21 shows the configuration for ‘Inside’. This configuration comes when the target entity is completely enclosed by the reference entity. In this configuration, there is a house and there is a table fan inside it. The spatial constraint for ‘Inside’ is given by

$$\{E_{(space - inside)} = true; BO_t \leq B_{(E - space - inside)}; E_t \diamond E_r = true; E_r \cap E_t = true\} \implies Inside$$

Figure 6.20: Spatial configuration of spatial constraint-I for 'Below'.

Figure 6.21: Spatial configuration of spatial constraint-I for 'Inside'.

Spatial Constraint - I for 'Outside': Fig. 6.22 shows the configuration for 'Outside'. The reference entity should have the property such that there is an empty space inside it. The target entity is placed outside the reference entity. The observer is located inside the reference entity and he views in Z-axis direction. In this configuration, there is a house and there is car, which is close to the house. The reference entity is the house and the target entity is the car.

$$\{E_{(space-inside)} = true; E_O \diamond E_r = true; E_r \cap E_t = false, D_f \leq \max(L_r, W_r, H_r) \times (1/2); d_t = [0, 0, 1]\} \implies Outside$$

Where d_t is unit direction vector of the target entity with respect to the reference entity. It is defined by the observer's viewing direction. Since the observer is viewing in along Z-axis direction the unit vector for Z component is one and others have zeros.

Figure 6.22: Spatial configuration of spatial constraint-I for ‘Outside’.

Spatial Constraints-I for ‘Left’, ‘Right’, ‘Front’, and ‘Back’ To model these constraints, the view direction of the observer plays an important role. In this case, modeling of the spatial constraints for the reference entity having the empty space inside or the top surface is considered. Fig. 6.24 shows the configurations for the constraints. In this case, the reference entity is taken as the garage. The target entity is taken as the table lamp. The observer views in Z-axis direction. The bounds of the XZ plane of the empty space is shown in Fig. 6.23 and it is represented by the centroid ce_{xz} and bounding points $P_{xz1}, P_{xz2}, P_{xz3}$ and P_{xz4} . The plane formed by the points $ce_{xz}, P_{xz1}, P_{xz4}$ represents the active region for ‘Front’. The plane formed by the points $ce_{xz}, P_{xz1}, P_{xz2}$ represents the active region for ‘Right’. The plane formed by the points $ce_{xz}, P_{xz2}, P_{xz3}$ represents the active

region for ‘Back’. The plane formed by the points $ce_{xz}, P_{xz3}, P_{xz4}$ represents the active region for ‘Left’. The most appropriate location for the target entity is the centroid of the triangle plane.

$$\{E_{(space-inside)} = true | S_{top} = true; E_t \cap \angle ce_{xz} P_{xz1} P_{xz4} = true; \\ P_{Et} \approx Centroid(ce_{xz}, P_{xz1}, P_{xz4})\} \implies Front$$

$$\{E_{(space-inside)} = true | S_{top} = true; E_t \cap \angle ce_{xz} P_{xz1} P_{xz2} = true; \\ P_{Et} \approx Centroid(ce_{xz}, P_{xz1}, P_{xz2})\} \implies Left$$

$$\{E_{(space-inside)} = true | S_{top} = true; E_t \cap \angle ce_{xz} P_{xz2} P_{xz3} = true; \\ P_{Et} \approx Centroid(ce_{xz}, P_{xz2}, P_{xz3})\} \implies Back$$

$$\{E_{(space-inside)} = true | S_{top} = true; E_t \cap \angle ce_{xz} P_{xz3} P_{xz4} = true; \\ P_{Et} \approx Centroid(ce_{xz}, P_{xz3}, P_{xz4})\} \implies Right$$

Figure 6.23: Bounds of the XZ plane of the reference entity's inside empty space for locating ‘Left’, ‘Right’, ‘Front’, and ‘Back’.

Spatial Constraints-II for ‘Left’, ‘Right’, ‘Front’, and ‘Back’ The spatial constraints-II are modeled for the reference entity, which may have empty space

Figure 6.24: Spatial configurations of spatial constraints-I for ‘Left’, ‘Right’, ‘Front’, and ‘Back’.

inside it or not. The view direction of the observer (i.e. X-axis), the position vector of the reference entity, and the difference between the position vectors of the target entity and the reference entity are considered for the modeling. Fig. 6.25 shows the configurations for the constraints. In the configurations, the reference entity is the water drum and the target entity is the table fan.

$$\{P_{Et} \approx [X_{Er} + (\max(L_r, W_r, H_r) \times (0.25 - 1)), Y_{Er}, Z_{Er}]; E_r \cap E_t = \text{false};\} \\ \implies \text{Front}$$

$$\{P_{Et} \approx [X_{Er} - (\max(L_r, W_r, H_r) \times (0.25 - 1)), Y_{Er}, Z_{Er}]; E_r \cap E_t = \text{false};\} \\ \implies \text{Back}$$

$$\{P_{Et} \approx [X_{Er}, Y_{Er}, Z_{Er} + (\max(L_r, W_r, H_r) \times (0.25 - 1))]; E_r \cap E_t = \text{false};\} \\ \implies \text{Left}$$

$$\{P_{Et} \approx [X_{Er}, Y_{Er}, Z_{Er} - (\max(L_r, W_r, H_r) \times (0.25 - 1))]; E_r \cap E_t = \text{false};\} \\ \implies \text{Right}$$

Spatial Constraints-I for ‘In front of’ and ‘Behind’ Fig. 6.26 shows the configurations for the constraints. In these, there is a car on the ground and the

Figure 6.25: Spatial configurations of spatial constraints-II for ‘Left’, ‘Right’, ‘Front’, and ‘Back’.

bucket is in front of the car. In the other configuration, the bucket is behind the car. It is assumed that the observer views in X-direction. The spatial constraints modeling is similar to the spatial constraints-II of ‘Front’ and ‘Back’.

$\{P_{Et} \approx [X_{Er} + (\max(L_r, W_r, H_r) \times (0.25 - 1)), Y_{Er}, Z_{Er}]; E_r \cap E_t = \text{false};\}$
 $\implies$ In front of

$\{P_{Et} \approx [X_{Er} - (\max(L_r, W_r, H_r) \times (0.25 - 1)), Y_{Er}, Z_{Er}]; E_r \cap E_t = \text{false};\}$
 $\implies$ Behind

Spatial Constraint-I for ‘Beside’ The spatial constraint for ‘Beside’ is similar to the spatial constraints-II of ‘Left’ and ‘Right’. Either one is correct for ‘Beside’.

Spatial Constraints-I for ‘Between’ and ‘Among’ This constraint has more than one reference entities. Fig. 6.27 shows the configurations for the constraints. In the first one, there is a table lamp and there is a banyan tree in front of the table lamp. There is an antenna between the table lamp and the banyan tree. Here, the table lamp and the tree are the reference entities. In the other

Figure 6.26: Spatial configurations of spatial constraints-I for ‘In front of’, and ‘Behind’

configuration, there are four entities randomly positioned on the ground. The bed is placed at the centroid of the reference entities positions.

$$\{n = 2; P_{Et} \approx \text{Centroid}(P_{Er1}, P_{Er2}); E_r \cap E_t = \text{false};\} \implies \text{Between}$$

$$\{n > 2; P_{Et} \approx \text{Centroid}(P_{Er1}, P_{Er2}, \dots, P_{Ern}); E_{r(1\dots n)} \cap E_t = \text{false};\} \implies \text{Among}$$

Spatial Constraints-I for ‘Near’ and ‘Far’ Fig. 6.28 shows the configurations for the constraints. In both the configurations, the house is taken as the reference entity and the water drum is taken as the target entity. The target entity’s unit direction vector d_t is defined by the observer viewing direction and he views in Z-axis direction. The spatial constraints for ‘Near’ and ‘Far’ are defined by

$$\{[If(E_t \square E_r) = \text{true}, D_f \approx \max(L_r, W_r, H_r) \times (1 - 3); Elseif((E_t \square E_r) = \text{false}, D_f \approx \max(L_r, W_r, H_r) + (Y_{Er} | Y_{Et}) \times (1 - 3)); E_r \cap E_t = \text{false}; d_t = [0, 0, 1]]\} \implies \text{Near}$$

$$\{D_f \geq \max(L_r, W_r, H_r) \times (4); E_r \cap E_t = \text{false}; d_t = [0, 0, 1]\} \implies \text{Far}$$

Figure 6.27: Spatial configurations of spatial constraints-I for 'Between' and 'Among'

Figure 6.28: Spatial configurations of spatial constraints-I for 'Near', and 'Far'.

6.6 Test on Predicting the Spatial Concept

In this section, KnowNet is tested to see how effectively it predicts the spatial concept given the spatial configuration between the entities. It is interesting to note that almost all the spatial concepts except ‘between’ and ‘among’ each has a configuration, which is constructed only between two entities such as the reference entity and the target entity.

Figure 6.29: Graphical user interface for testing the prediction of spatial concept. It is carried out to see how effectively it predicts the spatial concept given the spatial configuration between the entities.

In this test, the user creates the configuration network and has to select which is the reference entity and the target entity. If not selected, the entity having the greater volume is taken as the reference entity. After selecting the entities, the values for the spatial constraint parameters are calculated from the configuration. The generated parameters values are compared against the previously learned pa-

Figure 6.30: The Configurations of the entities and their predicted spatial concepts: part-I.

rameters values of the spatial concepts to predict the spatial concept between the entities. Fig. 6.29 shows the user interface for predicting the spatial concept. Its configuration construction pane is similar to the pane for the spatial concept learning. In the prediction pane, the list box displays the entities in the configuration. The user can select the reference entity and the target entity to predict the spatial concept. The first selection of the entity is taken as the reference entity. Fig. 6.30,

Figure 6.31: The Configurations of the entities and their predicted spatial concepts: part-II.

Fig. 6.31, and Fig. 6.32 show the different configurations of the entities and their corresponding predicted spatial concepts.

As seen from the above examples, KnowNet predicted the expected spatial concepts for the given configuration. The main reason for this is that the spatial constraints are modeled in a deterministic way rather than a probabilistic way, and the parameters are defined based on bounding box, empty space exist in the entity, intersection between the entities and the stability test.

Figure 6.32: The Configurations of the entities and their predicted spatial concepts: part-III.

6.7 Generation of Sentence(s) from Set of Entities

In the last sections, the modeling and learning of entities and their static interactions are discussed. These form the elementary meanings of the physical and conceptual world. This section discusses how to generate the sentences given the input set of entities. KnowNet has already learned the sixty entities and eighteen

spatial constraints, and their associated words.

The following procedure is used to generate the sentences from the set of entities.

- Get the physical model corresponding to each entity from the elementary meanings database.
- Pass the physical models to the rigid body statics algorithm. The following procedure is used to construct the configuration network. For more details, refer to Chapter 4
 - Arrange the entities in descending order based on the bounding box volume.
 - Select first two entities and the entity having bigger volume and more strength is taken as the reference entity
 - Perform the geometrical analysis on the reference entity and place it on the ground
 - Take the next entity and then create the possible configurations with respect to the reference entity. It is important to note that while generating the configurations, the target entity is positioned with respect to the reference entity based on the target entity's usage configuration.
 - Then get the next entity, if one exists and create the configurations of it with respect to the configurations already generated in the last step.
 - Continue the previous step till no entity exists
- Get conceptual models correspond to each entity's name.
- Using the configuration network and conceptual model as the input, generate the sentences using the procedure mentioned in Chapter 5

6.7.1 Sentence Generation Experiments

In this, the entities 'table', 'clock', and 'book' are taken for the sentence generation experiment. Based on the volume constraint, the table is identified as the reference

entity. Using the above procedure, the different possible configuration networks are constructed. Among ten configuration networks, six networks are chosen for the sentence generation and scene synthesis.

Figure 6.33: The sentences are generated from the configuration network. The thicker line shows that the entities intersect each other. The dotted lines are used to denote that the entities do not intersect.

Sentence Generation from Configuration-I Fig. 6.33 shows the configuration network for ‘table’, ‘clock’ and ‘book’. The nodes in the network represent the entities and edges represent the spatial relationship between the entities. The rigid body statics model determines the spatial relationship between the entities. The thicker lines denote that the reference entity and the target entity intersects each other. The dotted lines denote that they do not intersect. The number of sentences generated is equivalent to the number edges in the configuration network. Since there are five edges, the same number of sentences are generated. The following sentences are generated: 1) *The table is on the ground* 2) *The book is on the table* 3) *The clock is on the table* 4) *The book is near the clock* 5) *The clock is near the*

book

Sentence Generation from Configuration-II Fig. 6.34 shows the configuration network, in which there is a table and the book and the clock are on the table, and the two are certain distance apart. It is similar to the above configuration except that the observer views in the x-axis direction. The table and the book have two spatial relationships such as ‘on’ and ‘front’. Similarly, the table and the clock have ‘on’ and ‘back’. The book and the clock have positioned by the ‘near’ relationship. Since the configuration network has seven edges, seven different sentences are generated and they are as follows: 1) *The table is on the ground* 2) *The book is on the table* 3) *The clock is on the table* 4) *The book is near the clock* 5) *The clock is near the book* 6) *The book is front (of) the table* 7) *The clock is back (of) the table*.

Figure 6.34: The Sentences generation for the configuration-II

Sentence Generation from Configuration-III Fig. 6.35 shows the configuration network, in which the table is on the ground and the book is on the table and the clock is on the book. The configuration network has nodes that are connected one by one. This is due to that all the entities are touching each other. Since the network has four edges, only four sentences are generated. They are as follows: 1) *The table is on the ground* 2) *The book is on the table* 3) *The clock is on the book* 4) *The clock is above the table*.

Figure 6.35: The Sentences generation for the configuration-III

Sentence Generation from Configuration-IV Fig. 6.36 shows the configuration network, in which the table, the book and the clock are on the ground. All the entities touch the ground and none of them intersect each other. There are six edges that connect the entities. The following six sentences are generated: 1) *The table is on the ground* 2) *The book is on the ground* 3) *The clock is on the ground* 4) *The clock is under the table* 5) *The book is under the table* 6) *The clock is near*

the book

Figure 6.36: The Sentences generation for the configuration-IV

Figure 6.37: The Sentences generation for the configuration-V

Figure 6.38: The Sentences generation for the configuration-VI

Sentence Generation from Configuration-V Fig. 6.37 shows the configuration network, in which the table and the book are on the ground and the clock is on the book. Since the book and the clock are bounded in the empty space of the table, they have the spatial relationship ‘under’ with the table. The following six sentences are generated: 1) *The table is on the ground* 2) *The book is on the ground* 3) *The clock is on the book* 4) *The clock is above the ground* 5) *The book is under the table* 6) *The clock is under the table*.

Sentence Generation from Configuration-VI Fig. 6.38 shows the configuration network, in which the table and the clock are on the ground and the book is on the table. The table and clock have the ‘under’ spatial relationship and the book and the clock have the ‘below’ spatial relationship. There are six edges in the network and the following sentences are generated: 1) *The table is on the ground* 2) *The clock is on the ground* 3) *The book is on the table* 4) *The book is above the ground* 5) *The clock is under the table* 6) *The clock is below the book*.

6.8 Understanding of Sentences

Understanding deals with interpreting and verification of communicated information, which may be in the form of text or speech. The grammatical analysis is used to partially interpret the meaning of the sentences. It gives the information about agent, action, patient, spatial and timing relationships. But it will not tell whether given spatial relationship is true of the the agent and patient and will not tell the cause-effect relationship between the entities and properties of the entities.

The physical models are used to represent the entities and their interactions. The rigid body statics algorithm takes the physical models as the input and then it constructs the configuration network. The sequence of the entities and constraints are important for interpreting the input sentence. The sequence is obtained from the grammatical analysis of the sentence. The rigid body statics model is used to verify the meaning of the sentence(s). In other words, to verify whether the interpreted meaning follows the truthfulness of the physical world. The understanding of the compound sentences, the sentences related to complex prepositional phrases and the sentences that violate the truthfulness of the physical world are discussed in this section.

6.8.1 Grammatical Analysis of the Sentences

The main purpose of the grammatical analysis is to get ‘the reference entity-constraint-the target entity’ sequences from the sentences. They are identified from the sentences using the physical and conceptual models information of the words.

Grammatical Analysis of Simple Sentence As mentioned earlier, all the spatial constraints except ‘among’ and ‘between’ have been modeled by the properties of the reference entity and the target entity. Any simple sentences that describe the spatial concepts have the sentence pattern ‘*Noun phrase(NP)-Verb phrase(VP)-Prepositional phrase(PP)-Noun Phrase(NP)*’. Sometimes the constraints such as

front, back, left and right have the lexical property of ‘Noun’. Due to this, the sentence pattern becomes ‘*Noun phrase(NP)-Verb phrase(VP)-Noun phrase(NP)-Noun Phrase(NP)*’. Any words in the sentence that do not have the properties or the constraints will be discarded except for those belonging to the verb phrase.

The following procedure is used to perform the grammatical analysis of the simple sentence.

- Tokenize the sentences into the words
- Get the physical and conceptual models for each word.
- Form the phrases from the words using phrase formation rules. Then the phrases are classified as noun phrase acting as the entity (NP_E), noun phrase acting as the constraint (NP_C), prepositional phrase acting as the constraint (PP_C) and verb phrase (VP).
- The sequence ‘ $NP_{E1} - VP - PP_{C1}/NP_{C1} - NP_{E2}$ ’ is formed from the grammatical analysis. From that sequence the entities and constraint sequence is determined. The first noun phrase NP_{E1} is the target entity. The (PP_{C1}/NP_{C1}) comes immediately after the VP is taken as the constraint. The noun phrase (NP_{E2}) that comes after the constraint is taken as the reference entity. Finally, the sequence becomes *the target entity (NP_{E1}) - constraint (PP_{C1}/NP_{C1}) - the reference entity (NP_{E2})*
- Exchange positions of the target entity and the reference entity, then the sequence becomes *the reference entity (NP_{E2}) - constraint (NP_{C1}/PP_{C1}) - the target entity (NP_{E1})*
- Pass the sequence to the rigid body statics model

Let’s consider the example sentence ‘the ball is on the table’. The phrase sequence is given by ‘the ball (NP_{E1}) - is (VP) - on (PP_{C1}) - the table (NP_{E2})’.

From that sequence the entities and constraint information are determined. i.e. the reference entity (NP_{E2}) - constraint(PP_{C1}) - the target entity (NP_{E1}).

Grammatical Analysis of Sentence having Complex Prepositional Phrases

The simple sentence may have complex prepositional phrases that may be attached to the target entity or the reference entity. Let's consider the sentence 'the ball near the cup is on the table behind the car'. The phrase sequence is given by ' NP_{E1} (the ball) - PP_{C1} (near) - NP_{E2} (the cup) - VP (is) - PP_{C2} (on) - NP_{E3} (the table) - PP_{C3} (behind) - NP_{E4} (the car)'

The big phrase sequence is then decomposed to entities and constraint sequences of 'the target entity - constraint - the reference entity'. First, the main sequence is identified by taking the first entity in the sentence NP_{E1} (the ball)), the constraint that follows the verb phrase is PP_{C2} (on) and the entity that follows this constraint is NP_{E3} (the table)). The 'table' is the reference entity, 'on' is the constraint and the 'ball' is the target entity.

After getting the main sequence, the modifiers are identified for the entities in the main sequence. Here, there are two possible sequences exist in the sentence:

NP_{E1} (the ball) - PP_{C1} (near) - NP_{E2} (the cup)

NP_{E3} (the table) - PP_{C3} (behind) - NP_{E4} (the car). Then the positions of the target entity and the reference entity are interchanged. Finally, all three sequences are passed to the rigid body statics model to construct the configuration network.

6.8.2 Understanding of Compound Sentences

It is assumed that the observer is located in the ground and views in the Z-axis direction. The following algorithm is used to understand the compound sentences.

- First select the sequences that have constraint values having intersection between the reference entity and the target entity (i.e. $E_r \cap E_t = \text{true}$) is true.
- Construct the configuration network for the above using rigid body statics algorithm.

- Get the remaining sequence(s) and search the network whether it has the entity similar to that in the sequence. If it exists, use that entity spatial information to construct the configuration. If not, choose the arbitrary location on the ground and construct the configuration network.

Figure 6.39: Understanding of the sentences ‘*The table is on the ground. The desktop is left of the table. The monitor is right of the table. The keyboard is in front of the monitor*’ using rigid body statics model.

Understanding Experiment-I The compound sentences such as ‘*The table is on the ground. The desktop is left of the table. The monitor is right of the table. The keyboard is in front of the monitor*’ are considered for understanding. Fig. 6.39 shows the understanding of the above sentences. First the grammatical analysis is performed on the sentences to get the reference entity - constraint - the target entity sequences. Then the sequences are classified as the intersecting ones and the non-intersecting ones. First, the intersecting sequences are passed to the rigid body statics model to construct the configuration network. Then, the non-intersecting sequences is used to construct the configuration network. Here, it is important

to note that the constraint ‘left’ and ‘right’ has two types of modeling. They are intersecting and non-intersecting between the reference entity and the target entity. First, it is assumed that the constraints are of the intersecting type. The desktop and the monitor are placed on the top surface (i.e. right and left) of the table. If the top surface is unable to hold the entities in its left and right area, the non-intersecting constraint is used to position the entities.

Understanding Experiment-II The compound sentences such as ‘*The bed is on the ground. The chair is beside the bed. The table lamp is on the chair. The table fan is behind the bed. The chair is on the ground*’ are considered for understanding. Fig. 6.40 shows the configuration network and the scene of the network. The sentence ‘the chair is beside the bed’ can be interpreted in two ways. The first configuration would be the chair is placed on the bed’s either right or left area. The second configuration would be where the chair is placed on the ground and the both should not touch each other. The sentence ‘the chair is on the ground’ will help the rigid body statics model to select the second configuration.

Understanding Experiment-III The compound sentences such as ‘*The rack is on the ground. The bowl is in the rack. The teapot is in the rack. The paper is below the teapot. The paper weight is on the paper*’ are considered for understanding. Fig. 6.41 shows the configuration network and the scene of the network. The bowl and the teapot can placed in the empty spaces that exist between the parts of the rack. With respect to the teapot, the paper should be placed at some vertical distance away in the negative Y-direction.

Understanding Experiment-IV The compound sentences such as ‘*The burger is in the bowl. The banana is in the plate. The plate is on the table. The bowl is left of the plate. The bucket is below the bowl*’ are considered for understanding. Fig. 6.42 shows the configuration network and the scene of the network.

Figure 6.40: Understanding of the sentences ‘The bed is on the ground. The chair is beside the bed. The table lamp is on the chair. The table fan is behind the bed. The chair is on the ground’ using rigid body statics model

Figure 6.41: Understanding of the sentences ‘The rack is on the ground. The bowl is in the rack. The teapot is in the rack. The paper is below the teapot. The paper weight is on the paper’ using rigid body statics model

6.8.3 Understanding of the Sentences Having Complex Prepositional Phrases

The sentence having the complex prepositional phrases is considered for the understanding. The grammatical analysis mentioned above is used to identify the

Figure 6.42: Understanding of the sentences ‘*The burger is in the bowl. The banana is in the plate. The plate is on the table. The bowl is left of the plate. The bucket is below the bowl*’ using rigid body statics model

entities and constraints sequence from the sentence. Then the sequences are classified into intersecting and non-intersecting ones. The sequences are finally passed to the rigid body statics model for constructing the configuration network.

Understanding Experiment-I The complex sentence ‘*The clock on the table is over the bowl near the bucket*’ are considered for understanding. Fig. 6.43 shows the entities and constraint sequences, configuration network and the scene of the network. The complex sentence is decomposed into one main sequence and two modifier sequences. The main sequence is ‘the clock - over - the bowl’. The modifier sequences are ‘the clock - on - the table’ and ‘the bowl - near - the bucket’. First, the intersecting sequence ‘the clock - on - the table’ is used to construct the network. The table is placed on the ground and the clock is placed on the table. Then using the modifier sequence the bowl is placed on the ground so that with respect to the bowl, the clock should maintain the some positive vertical distance with the bowl. Using the other modifier sequence, the bucket is placed in such a way that the bowl is near the bucket.

Figure 6.43: Understanding of the sentences 'The clock on the table is over the bowl near the bucket' using rigid body statics model

Understanding Experiment-II The complex sentence 'The ball near the pear /monitor is on the round table' are considered for understanding. Fig. 6.44 shows the entities and constraint sequences, configuration network and the scene of the network. The main sequence is 'the ball - on - the round table' and the modifier sequence is 'the ball - near - pear/monitor. First the ball is placed on the round table. Since the pear is small in size it will be placed on the table with a distance of one to three times of the maximum bounding box parameters of the ball. If the monitor is taken, the round table top surface is unable to hold that. So, the monitor is placed on the ground and the distance between the ball and the monitor is two times the vertical distance between the ground and the ball, and the maximum bounding box parameters of the monitor. Otherwise, if the monitor is placed on the ground with the distance of the monitor bounding box, the final relation between them will be 'under'.

Figure 6.44: Understanding of the sentences ‘The ball near the pear/monitor is on the round table’ using rigid body statics model. There are two configurations for the sentence. The first one is with the pear and the other one is with the monitor.

6.8.4 Understanding of the Sentences that Violate the Truthfulness of the Physical World

The understanding of the sentences that do not follow the truthfulness of the physical world is considered. If such sentences are given as input, KnowNet will identify the problem and output the corrected sentence. The violation comes mainly due to strength failure, stability failure, unavailability of empty spaces and the empty space volume limitation.

Strength Failure The strength failure comes when the heavy entity is placed on the less strong entity. Such failure is determined by rigid body statics model using the entities properties. The sentence ‘The rose plant is on the ant’ is considered for understanding. Fig. 6.45 shows two interpretations of the sentence. The first configuration is constructed without considering the physics model. The sentence can be interpreted as follows: the ant is placed on the ground. On the top of the

ant, the rose plant is placed on it. If the sentence is interpreted with rigid body statics model, the weight of the target entity and strength of the reference entity is compared to test the failure. In this case, the ant strength is lesser than the weight of the rose plant so the failure of the reference entity takes place. Then the entities are exchanged and modified configuration can be constructed. The configuration will be that ‘the ant is under the rose plant’.

Figure 6.45: Understanding of the sentence ‘*The rose plant is on the ant*’

Stability Failure If the entity’s center of gravity does not lie within its contact area, stability failure occurs. The sentence ‘the bucket is on the ball’ is considered for understanding. Fig. 6.46 shows the configurations of the sentence. The bucket is placed on the top of the ball. The combined center of gravity of the bucket and the ball should lie in the contact area between the ball and the ground. In this case, the ball does not have any flat patch on its top surface and the center of gravity does not lie in the contact area. So there is stability failure. The other

corrected configurations such as near, left, right and in front of are possible. The configuration with ‘in front of’ is taken as the corrected configuration.

Figure 6.46: Construction of modified configuration for the sentence ‘the bucket is on the ball’. It fails in the stability test.

Unavailability of the Empty Space in the Entity Some spatial constraints are modeled based on the empty spaces that exist in the reference entity. If the empty space does not exist, the spatial relationship won’t be true between the entities. The sentence ‘The bowl is inside the chair’ is taken for the interpretation. Fig. 6.47 shows the configurations of the sentence. When the chair is analyzed for the different types of the empty spaces, it has empty space on the top and the empty space between the ground and the entity’s immediate part. The enclosed empty space is necessary for the constraint ‘inside’. But the chair does not have the enclosed space. So, the spatial constraint ‘inside’ is not true for the chair. The rigid body statics model generates the configurations of the entities with the spatial constraint ‘under’ and ‘on’.

Figure 6.47: Construction of modified configuration for the sentence 'the bowl is inside the chair'. The chair does not have enclosed empty space so the constraint 'inside' is not applicable.

The Empty Space Volume Limitation The volume of the empty space plays an important role to hold the target entity. The sentence 'the desktop is in the bucket' is taken for understanding. Fig. 6.48 shows the configurations of the sentence. The bucket has enclosed empty space in the Y axis direction. But its volume is lesser than the bounding box of the desktop. So, the constraint 'in' is not applicable between the bucket and the desktop. The modified configuration is the placement of bucket on the desktop.

6.9 Discussion

The words related to the entity and the spatial concepts are learned through modeling the entities and their static interactions. The spatial constraints are modeled and learned through the entities interactions. There are nineteen basic spatial position words that are learned through the programmed interface. For learning these spatial concepts, the experiments used only twenty five different types of entities.

Figure 6.48: Construction of modified configuration for the sentence 'the desktop is in the bucket'. The bucket empty space volume is lesser than the desktop volume so the constraint 'in' is not applicable between them.

In total sixty entities have been learned by KnowNet. Testing is carried out on the learned spatial concepts as to how well the modeling predicts the spatial concept given the spatial configuration as input. Sixteen configurations are given as input and KnowNet predicts the expected spatial concepts accurately for all the configurations. The proposed spatial constraints are modeled in a deterministic way rather than a probabilistic way. Most of the existing works model the spatial language in terms of probabilistic modeling and none of them considered the modeling the entities and their static interactions for spatial language learning and understanding. The rigid body statics model takes the entities and constraint sequences as the input and then constructs the configuration network. The sentence generation experiments and different types of understanding experiments have been performed.

Chapter 7

Conclusions, Contributions and Future Work

7.1 Conclusions

A mind for the machine has been designed and implemented to autonomously learn, analyze and synthesize natural language text related to spatial relations. The organized memory is a core component, in which the modeling of physical and conceptual world elements are represented. This thesis has clearly shown the importance of the organized physical meanings for the machine in learning and understanding the meaning of sentences related to spatial relations. A new direction is taken to model the spatial language in a deterministic way using entities properties and their interactions. The spatial words such as on, in, over, under, above, below, inside, outside, behind, in front of, back, front, left, right, beside, between, among, near and far are modeled and learned. The spatial concepts are learned through modeling the entities and their interactions. In contrast, current works use collocation information of words to learn the spatial concepts.

Just given the entities properties and constraints, the machine has to determine the possible interactions between the entities and possible actions or behaviors that can be generated by the entity. This is an important pre-requisite for the machine to learn the natural language. Constraint physics or physical world physics controls interactions between the entities. It gives the new direction to interpret the meaning of the sentences without using the language semantic knowledge. I

believe that this work is one of the possible ways and starting points to realize true machine intelligence in machines.

7.2 Contributions

A mental architecture is proposed and implemented for a machine to autonomously learn and understand the natural language text related to spatial relations. The implemented system is called *KnowNet* and the following are the main contributions of the thesis.

- Proposed and implemented mental architecture for the machine to understand natural language related to spatial relations. The implemented system is called ‘KnowNet’.
- Organized memory is designed, in which modeling of physical and conceptual world elements are represented. It consists of elementary meanings and composite meanings. Elementary meanings are acquired or learned by machine through programmed interface by real time interaction with the human agent.
- Designed and developed the rigid body statics algorithm for the machine to understand the meaning of sentence(s) that are related to entities and spatial relations. KnowNet does not use semantic knowledge of the words for the understanding.

7.3 Future Work

This thesis has implemented modeling of elementary meanings for the physical and conceptual world elements, construction of organized representations of physical meaning such as configuration networks and construction of organized conceptual meanings such as concept networks from the physical meanings and conceptual models. The following are the main future works:

Learning of Elementary Meanings through Sensors: Currently, the physical models and conceptual models are acquired through a programmed interface. Actually, they should be acquired or learned through a machine's sensory systems such as camera sensor for geometrical properties, microphone for speech signal acquisition, sensors for measuring the mechanical properties, etc.

Constraint Physics Algorithm for Constructing Organized Physical Meanings: So far, a constraint physics model such as the rigid body statics algorithm has been developed to construct the configuration network. Just considering the geometrical and mechanical properties, the machine has constructed the configuration network using rigid body statics physics. Similar to that, by considering the configuration network, degrees of freedom (DOF) between the entities and other properties, the machine can construct the behavior network (i.e. chain of cause-effect relationship between the entities) and from which machine has to learn the concepts related to action, behavior and events. In a similar vein, the physics algorithms for event and episode network have to be developed. Any machine that is going to learn human languages should know how to construct the organized physical meanings.

Algorithm for Constructing the Conceptual Constraint Networks: The machine can synthesize meaningful sentences only if they (i.e. sentences) are generated from physical meanings. The conceptual constraint networks help the machine

compose the articles or books by itself from the physical meanings.

Physical Models for All Entities: This work considers the construction of physical meaning only for rigid bodies. But the physical world consists of different kinds of entities such as liquids, gases, solid objects, etc. Modeling the interaction between all these entities is very important for the machine to learn and understand all the aspects of natural language.

Understanding of Natural Languages Other than English: Since the machine use rigid body statics algorithm to understand the meaning of the sentences and elementary meanings are in the form of properties and constraints, the language semantic rules do not play any role here for the text understanding. As a result, any natural language can be learned and understood by the machine.

Bibliography

- [1] M. Xie, J. S. Kandhasamy, and H. F. Chia, “Meaning-centric framework for natural text/scene understanding by robots,” *International Journal of Humanoid Robotics*, vol. 1, June 2004.
- [2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.
- [3] D. W. Carrol, *Psychology of language*. Brooks/Cole Publishing Company, 2001.
- [4] “<http://www.cultsock.ndirect.co.uk/muhome/cshtml/semiomean/meaning1.html>,” June 2004.
- [5] “<http://en.wikipedia.org/wiki/meaning>,” June 2004.
- [6] “http://en.wikipedia.org/wiki/charles_sanders_peirce,” June 2004.
- [7] W. Carl, *Frege’s Theory of Sense and Reference*. Cambridge University Press, 1994.
- [8] A. M. Collins and E. F. Loftus, “A spreading activation theory of semantic processing,” *Psychological Review*, vol. 82, pp. 407–428, 1975.
- [9] A. M. Collins and M. R. Quillian, “Retrieval time from semantic memory,” *Journal of Verbal Learning and Verbal Behavior*, no. 8, pp. 240–247, 1969.
- [10] “http://www.tcd.ie/psychology/ruth-byrne/mental_models/,” July 2004.
- [11] P. N. Johnson-Laird, *Mental models*. Harvard University Press, 1983.

- [12] P. N. Johnson-Laird, "Mental models and deduction.," *Trends in Cognitive Science*, no. 5, pp. 434–442, 2001.
- [13] E. Tulving, "Episodic and semantic memory," in *Organization in memory* (E. Tulving and W. Donaldson, eds.), New York: Academic Press, 1972.
- [14] W. Kintsch, *Learning, memory, and conceptual processes*. New York: Wiley, 1970.
- [15] W. Kintsch and T. vanDijk, "Toward a model of text comprehension and production," *Psychological Review*, vol. 85, no. 5, pp. 363–394, 1978.
- [16] J. A. Fodor, *The language of thought*. New York: Thomas Y. Crowell, 1975.
- [17] J. A. Fodor, *Psychosemantics*. Cambridge MA: MIT/Bradford, 1987.
- [18] J. A. Fodor, "Methodological solipsism considered as a research strategy in cognitive psychology.," *Behavioral and Brain Sciences*, no. 3, pp. 63–109, 1980.
- [19] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical appraisal," *Cognition*, vol. 28, pp. 3–71, 1988.
- [20] A. Newell, "Physical symbol systems.," *Cognitive Science*, no. 4, pp. 135–183, 1980.
- [21] J. R. S. Searle, "Minds, brains and programs," *Behavioral and Brain Sciences*, no. 3, pp. 417–457, 1980.
- [22] S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, pp. 335–346, 1990.
- [23] S. Harnad, "Symbol grounding and the origin of language," in *Computationalism: New Directions* (Scheutz and Matthias, eds.), MIT Press, 2002.
- [24] L. Steels and P. Vogt, "Grounding adaptive language games in robotic agents," in *Proceedings of the Fourth European Conference on Artificial Life, ECAL'97*, 1997.

- [25] L. Steels and A. McIntyre, "Spatially distributed naming games," *Advances in Complex Systems*, vol. 1, no. 4, pp. 301–323, 1999.
- [26] L. Steels and F. Kaplan, "Situated grounded word semantics," in *IJCAI'99 Proceedings of the 16th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 862–867, 1999.
- [27] L. Steels and F. Kaplan, "Aibo's first words: The social learning of language and meaning," *Evolution of Communication*, vol. 1, no. 4, 2001.
- [28] L. Steels, "Language games for autonomous robots," *IEEE Intelligent Systems*, pp. 16–22, 2001.
- [29] D. Roy, "Grounding words in perception and action: Insights from computational models," *Trends in Cognitive Science*, In Press.
- [30] D. Roy and A. Pentland, "Learning words from sights and sounds: A computational model," *Cognitive Science*, vol. 26, no. 1, pp. 113–146, 2002.
- [31] D. Roy, K. Y. Hsiao, and N. Mavridis, "Mental imagery for a conversational robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 3, pp. 1374–1383, 2004.
- [32] D. Roy, "Grounded spoken language acquisition: Experiments in word learning," *IEEE Transactions on Multimedia*, 2003.
- [33] P. Vogt, "The physical symbol grounding problem," *Cognitive Systems Research*, vol. 3, pp. 429–457, 2002.
- [34] P. Vogt, "Anchoring of semiotic symbols," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 109–120, 2003.
- [35] P. Cohen, "Projections as concepts," in *Proceedings of the 2nd European Conference on Cognitive Science*, pp. 56–60, 1997.

- [36] P. Cohen, T. Oates, C. Beal, and N. Adams, “Contentful mental states for robot baby,” in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 2002.
- [37] R. Katarzyniak, “The language grounding problem and its relation to the internal structure of cognitive agents,” *Journal of Universal Computer Science*, vol. 11, no. 2, pp. 357–374, 2005.
- [38] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, “Introduction to wordnet: An on-line lexical database,” *International Journal of Lexicography*, vol. 3, no. 4, pp. 235–244, 1990.
- [39] S. Richardson, W. Dolan, and L. Vanderwende, “Mindnet: acquiring and structuring semantic information from text,” in *Proceedings of COLING ’98*, 1998.
- [40] H. Liu and P. Singh, “Conceptnet: a practical commonsense reasoning toolkit,” *BT Technology Journal*, vol. 22, no. 4, pp. 211–226, 2004.
- [41] D. B. Lenat, “Cyc: A large-scale investment in knowledge infrastructure,” *Communications of the ACM*, vol. 38, no. 11, 1995.
- [42] K. Kipper, H. T. Dang, and M. Palmer, “Class-based construction of a verb lexicon,” in *Seventeenth National Conference on Artificial Intelligence*, (Austin, TX), 2000.
- [43] Marcus, Mitchell, K. Grace, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, “The penn treebank: Annotating predicate argument structure,” in *Proceedings of Human Language Technology Workshop*, pp. 110–115, 1994.
- [44] B. Collin, F. Charles, and L. John, “The berkeley framenet project,” in *Proceedings of the COLING-ACL*, (Montreal, Canada), 1998.

-
- [45] “<http://www.cs.unt.edu/rada/downloads.html>,” July 2004.
 - [46] T. Chklovski and R. Mihalcea, “Open mind word expert: Creating large data collections with web users help,” in *DLib Magazine*, June 2002.
 - [47] P. Blackburn and J. Bos, *Representation and Inference for Natural Language: A First course in Computational Semantics*. CSLI Press, Stanford, 2005.
 - [48] S. Robert and S. Suzanne, “Exploiting a verb lexicon in automatic semantic role labelling,” in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, (Canada), 2005.
 - [49] C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.
 - [50] J. F. Allen, *Natural Language Understanding*. Benjamin Cummings, Second Edition 1994.
 - [51] J. Dowding, J. Gawron, D. Appelt, L. Cherny, R. Moore, and D. Moran, “Gemini: A natural language system for spoken language understanding,” in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, (Columbus, OH), 1993.
 - [52] J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski, “A robust system for natural spoken dialogue,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 62–70, June 1996.
 - [53] E. T. Mueller, “Understanding script-based stories using commonsense reasoning,” *Cognitive Systems Research*, vol. 5, no. 4, pp. 307–340, 2004.
 - [54] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, “Spatial language for humanrobot dialogs,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, May 2004.

- [55] W. Adams, D. Perzanowski, and A. C. Schultz, "Learning, storage and use of spatial information in a robotics domain," in *Proceedings of the Workshop on Machine Learning Spatial Language*, pp. 22–27, 2000.
- [56] K. R. Coventry, "Spatial prepositions, spatial templates and semantic versus pragmatic visual representations," in *Representing Direction in Language and Space* (E. van der Zee and J. Slack, eds.), Oxford University Press, 2003.
- [57] Y. Shinyama, T. Tokunaga, and H. Tanaka, "Kairai: Software robots understanding natural language," in *Third International Workshop on Human-Computer Conversation*, pp. 1–7, July 2000.
- [58] N. Badler, W. Becket, B. D. Eugenio, C. Geib, L. Levison, M. Moore, B. Webber, M. White, and X. Zhao, "Intentions and expectations in animating instructions: the animnl project," in *Intentions in Animation and Action*, (Institute for Research in Cognitive Science, University of Pennsylvania), March 1993.
- [59] R. Moore, J. Dowding, H. Bratt, J. M. Gawron, Y. Gorf, and A. Cheyer, "Commandtalk: A spoken-language interface for battlefield simulations," in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 1–7, 1997.
- [60] P. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "Quickset: Multimodal interaction for distributed applications," in *ACM International Multimedia Conference*, (New York), pp. 31–40, 1997.
- [61] H. Ueno, "A knowledge-based information modeling for autonomous humanoid service robot," *IEICE Transactions on Information and Systems*, vol. E85-D, no. 4, pp. 657–665, 2002.

- [62] S. Lopes and A. Teixeira, “Human-robot interaction through spoken language dialogue,” in *International Conference on Intelligent Robots and Systems*, (Takamatsu, Japan), 2000.
- [63] T. L. Lueth, T. Herzog, G. Stopp, and G. Kamstrup, “Kanttra: Human-machine interaction for intelligent robots using natural language,” in *IEEE International Workshop on Robot and Human Communication*, (Nagoya, Japan), pp. 106–111, 1994.
- [64] C. Fischer and G. Schmidt, “Multi-modal human-robot interface for interaction with a remotely operating mobile service robot,” *Advanced Robotics*, vol. 12, no. 4, pp. 397–409, 1998.
- [65] B. Coyne and R. Sproat, “Wordseye: An automatic text-to-scene conversion system,” in *SIGGRAPH*, (Los Angeles, CA), 2001.
- [66] S. Dupuy, A. Egges, V. Legendre, and P. Nugues, “Generating a 3d simulation of a car accident from a written description in natural language: the carsim system,” in *ACL Workshop proceedings: Temporal and Spatial Information Processing*, July 2001.
- [67] A. Herskovits, *Language and Spatial Cognition*. Cambridge university press, 1986.
- [68] L. Talmy, “How language structures space,” in *Spatial orientation: Theory, research and application* (H. Pick and L. Acredolo, eds.), pp. 225–282, New York: Plenum, 1983.
- [69] G. Lakoff, *Women, fire and dangerous things*. Chicago University Press, 1987.
- [70] T. Regier, *The human semantic potential. Spatial language and constrained connectionism*. MIT Press, 1996.

- [71] G. Logan and D. Sadler, “A computational analysis of the apprehension of spatial relations,” in *Language and space* (L. N. P. Bloom, M. A. Peterson and M. Garrett, eds.), pp. 493–529, MIT press, 1996.
- [72] K. Coventry, M. Pratsala, and L. Richards, “The interplay between geometry and function in the comprehension of over, under, above, and below,” *Journal of Memory and Language*, no. 44, pp. 376–398, 2001.
- [73] G. Martinez, A. Cangelosi, and K. Coventry, “A hybrid neural network and virtual reality system for spatial language processing,” in *Proceedings of IEEE*, pp. 171–180, 1996.
- [74] W. D. Callister, *Materials Science and Engineering: An Introduction*. Wiley Publishing Company, 2002.
- [75] J. Craig, *Introduction to Robotics – Mechanics and Control*. Addison-Wesley Publishing Company, 1986.
- [76] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [77] “Adams: <http://www.mscsoftware.com.au/products/software/msc/adams/>,” July 2004.
- [78] “Matlab simmechanics: <http://www.mathworks.com>,” July 2004.
- [79] R. Norton, *Design of Machinery - An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill, Inc., 1992.
- [80] D. Meagher, *The octree encoding method for efficient solid modeling*. UMI, 1983.
- [81] S. Gottschalk, M. Lin, and D. Manocha, “Obbtree: A hierarchical structure for rapid interference detection,” in *Proceedings of SIGGRAPH*, pp. 171–180, 1996.

-
- [82] “Obb intersection: <http://www.gamasutra.com/>,” August 2004.
- [83] T. Moller, “A fast triangle-triangle intersection test,” *Journal of Graphics Tools*, vol. 2, no. 2, pp. 25–30, 1997.
- [84] T. Moller and B. Trumbore, “Fast minimum storage ray-triangle intersection,” *Journal of Graphics Tools*, vol. 2, no. 1, pp. 21–28, 1997.
- [85] R. Hobbeler and S. Fan, *Engineering Mechanics - Statics*. Prentice Hall, 1997.
- [86] F. Dunn, *3D Math Primer for Graphics and Game Development*. Wordware Publishing Inc, 2002.
- [87] D. M. Bourg, *Physics for Game Developers*. OREILLY, 1996.
- [88] H. Leipholz, *Stability theory: an introduction to the stability of dynamic systems and rigid bodies*. B.G. Teubner, 1987.
- [89] “Stability: <http://www.phy.ntnu.edu.tw/java/block/block.html>,” August 2004.
- [90] C. Barber, D. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.
- [91] “Opengl: <http://www.opengl.org>,” September 2004.