HIERARCHICAL INCREMENTAL CONTEXT MODELING ON ROBOTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FETHİYE IRMAK DOĞAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

DECEMBER 2017

Approval of the thesis:

HIERARCHICAL INCREMENTAL CONTEXT MODELING ON ROBOTS

submitted by **FETHİYE IRMAK DOĞAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering Department, METU** _____

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Erol Şahin
Computer Engineering Department, METU _____

Assist. Prof. Dr. Emre Akbaş
Computer Engineering Department, METU _____

Assist. Prof. Dr. Esra Kadıoğlu Ürtiş
Computer Eng. Dept., TOBB Univ. of Economics and Technology _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: FETHİYE IRMAK DOĞAN

Signature :

ABSTRACT

HIERARCHICAL INCREMENTAL CONTEXT MODELING ON ROBOTS

DOĞAN, FETHİYE IRMAK
M.S., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Sinan Kalkan

December 2017, 76 pages

Context is very crucial for robots to be able to adapt themselves to circumstances and to fulfill their tasks accordingly. There have been many studies on modeling context on robots, however, these studies either do not construct an incremental and hierarchical structure (i.e., use a fixed number of contexts and context layers) or determine the necessity of adding a new context by using rule-based approaches. In this thesis, we propose two different methods to model context. In the first method, we extend the Restricted Boltzmann Machines, a generative associative model, by incrementing the number of contexts and context layers when needed. This model constructs the hierarchical and incremental contextual representations by considering the confidence of the objects and contexts after each new scene encountered. Moreover, this deep incremental model obtains better or on-par results when compared to the incremental or non-incremental models in the literature on different tasks. In the second method, in contrast to our first method and the methods in the literature, determining the necessity of adding a new context is formulated as a learning problem. In order

to be able to do that, Latent Dirichlet Allocation (LDA) model is used to generate the data with known number of contexts. The intermediate LDA models with/without the correct number of contexts are then fed to a Recurrent Model, which is trained to predict whether to add a new context or not. Our analysis on artificial and real datasets demonstrate that such a learning-based approach generalizes well, and is a promising approach for solving such incremental problems.

Keywords: Context, Hierarchical/Incremental Context Modeling, Artificial Neural Networks, Developmental Robotics, Deep Learning

ÖZ

ROBOTLARDA HİYERARŞİK ARTTIRIMLI BAĞLAM MODELLENMESİ

DOĞAN, FETHİYE IRMAK
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Sinan Kalkan

Aralık 2017, 76 sayfa

Bağlam, robotlar için kendilerini çevre koşullarına adapte etmede ve görevlerini düzgün bir şekilde yerine getirmede çok önemlidir. Bağlamın robotlarda modellenmesi çalışılmış bir konu olsa da bu çalışmalar ya artırımlı ve hiyerarşik bir yapı oluşturmayıp bağlam sayısının ve bağlam katmanlarının belirli olduğunu varsaymış ya da bağlam sayısını arttırmak için kural tabanlı yaklaşımlar izlemişlerdir. Bu tezde, bağlamı modellemek için iki farklı yöntem önermekteyiz. İlk yöntemde, üretken ilişkisel bir model olan Kısıtlı Boltzmann Makineleri'ni bağlam sayılarını ve bağlam seviyelerini gerekli durumlarda artıracak şekilde genişletmekteyiz. Bu yöntem, nesnelerin ve bağlamların temsil edilebilme miktarlarını göz önünde bulundurarak karşılaşılan her bir yeni sahnede hiyerarşik ve artırımlı bağlamsal ilişkileri modellemektedir. Buna ek olarak, sunulan derin artırımlı model, çeşitli görevlerde literatürdeki artırımlı olan ve olmayan modellerle karşılaştırıldığında eşit düzeyde ya da daha iyi performansa sahip sonuçlar elde etmiştir. İkinci yöntemde, ilk yöntemdekinin ya da literatürdeki diğer çalışmaların aksine, modele yeni bir bağlam eklemenin gerekliliği kural tabanlı

bir yaklaşım kullanılmadan, bağlam sayısını arttırmayı öğrenme problemi olarak formüle edilmiştir. Bu problemi çözmek için, Gizli Dirichlet Ayırma yöntemi ile bağlam sayısı belirli olan bir veri kümesi elde edilmiştir. Doğru ya da yanlış sayıdaki bağlam sayısı ile eğitilmiş Gizli Dirichlet Ayırma yönteminin her bir ara modeli Tekrarlı Sinir Ağları'na girdi olarak sağlanmıştır. Bu girdiyi kullanarak, tekrarlı derin model, yeni bir bağlam eklemenin gerekli olup olmadığını tahmin etmek için eğitilmiştir. Yapay ve gerçek veriler kullanılarak yapılan incelemeler, böylesi bir öğrenme tabanlı yaklaşımın iyi bir genelleme kapasitesi sağladığını ve artırımlı problemleri çözmede umut vadettiğini göstermiştir.

Anahtar Kelimeler: Bağlam, Hiyerarşik/Artırımlı Bağlam Modelleme, Yapay Sinir Ağları, Gelişimsel Robotik, Derin Öğrenme

To my family who advised me making beneficial deeds for the humanity and friends who made me feel the warmth of a family

ACKNOWLEDGMENTS

I am very thankful to everyone, who was with me at the happiest and saddest moments, made my life more valuable and collected precious memories with me.

To my family, for not only encouraging me to follow my own path but also being so humanitarian and supporting the peace and equality of the people. They tried to teach me being on the side of fairness and the most important responsibility in my life is to be a good person. I feel always lucky to have them. They were always very patient and never gave up believing in me even I did. I am proud of having the most empathic and understanding mother of the world and the most sensitive father to the problems of the humanity. I hope everybody can have a family like them.

To Sinan Kalkan, being such a good supervisor and a good-hearted person. I met him while I was in the second year of my Bachelor and this totally changed the way of my academic life. He showed me how to be a good researcher, a good teacher, a good advisor and most importantly a good person. He was always ready to help me when I felt stuck about any subject from scientific ones to my health problems. I am very honored to work with him and I will always be grateful to him for the things he taught me. All his efforts as a supervisor, researcher, and friend will stay priceless and unforgettable for me.

To Göktürk Üçoluk, for providing me lots of laughs and funny talks. He was very patient while trying to help me about the way of my academic career and gave me invaluable advices. He made me feel like a part of Amele Sofrası and showed me how a good professor can be cheerful, love to share and kind. He was always very thoughtful and take me from the department to lab with his lovely conversation.

To Erol Şahin, for lending a helping hand to us in the lab. Whenever I need a help for anything in the lab, I know he will be willing to help me. Moreover, he is a perfect academic and students taking a course from him are very lucky since they will have an opportunity to obtain a solid background about the issue.

To Fatoş Yarman Vural, for helping me while directing my academic life starting from my Bachelor. She supervised my last year Bachelor project with a big patience and was willing to help me whenever I have a question about not only pattern recognition but also important decisions regarding my education.

To Hande Çelikkanat, for being such a positive, gracious and helpful person. She was

the mother of the lab with a huge love for everybody. If anybody needed anything, Hande would be there to help regardless of the subject. She showed me how a person should behave in human relations. Moreover, Hande has a compound background about lots of research topics and she helped me a lot while constructing my models in my thesis with her patience and smile.

To my dear lab friends, for turning the lab into a friendly environment. I would like to thank İlker Bozcan for helpful talks about Boltzmann Machines, enjoyful night walking, protecting me against the dogs in the paper submission deadline period and never leaving me alone. I am very thankful to Ezgi Ekiz for her nice friendship, sharing lots of common interests, making me laugh a lot, listening to my privates and not hesitating to share hers. I am also thankful to Cemal Aker for his nice talks on scientific issues, useful discussions about neural networks and being a very kind peer during my master. I want to thank Osman Dursun for being a funny and talkative friend and for helping me a lot while adapting the lab.

To each member of Ciciş family, who made my university days valuable. We were together at the happiest and saddest moments of the university and we become a family to each other in our university days.

To each member of Ayı family, for growing up with me and sharing lots of precious moments. Thanks to you, I am not a single child anymore and you are my selected family.

To Seçil Güler, for accepting me as who I am, being always with me since we met in High School, so faithful and a friend of bad days.

To Hüseyin Aydın, for answering my questions about the thesis process patiently and always being a kind person. He is the one you always want to be in your life.

To Hüsnü Şener, who helped me a lot, listened to my problems, tried to help me in my decisions and never left me alone after my eye surgery. Being his friend is priceless to me.

To Güven Turan, who was always with me, supported me and believed in me every step of my life for more than seven years. He is the one who deserves best.

I am also thankful to Scientific and Technological Research Council of Turkey (TÜBİTAK) for founding “Context in Robots” project with project number 215E133 and NVIDIA for donating Tesla K40 GPU which is used for some parts of the experiments.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

BM	Boltzmann Machine
RBM	Restricted Boltzmann Machine
iRBM	Incremental Restricted Boltzmann Machine
DBM	Deep Boltzmann Machine
diBM	Deep Incremental Boltzmann Machine
LDA	Latent Dirichlet Allocation
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory

CHAPTER 1

INTRODUCTION

We live in sophisticated environments, which we try to perceive and understand from low-level sensory data captured using limited low-level sensors. Often, such low-level projections of the environment bear incomplete, noisy and ambiguous information, making interpretation and perception of such an environment very challenging. In visual perception, we e.g. try to estimate a 3D model of the environment, from its 2D projections. An important mechanism that we employ in addressing this challenge is the previous knowledge, or experience, which is present in the current stimuli, but appear to be irrelevant to the current task – see e.g. Figure 1.1. Such a priori information that modulate our processing stages is called context.

Context is essential for our cognitive capabilities, functioning as a modulator affecting our perception, reasoning, communication and action [5, 66]. Context helps these processes in e.g. by resolving ambiguities, rectifying mispredictions, filtering irrelevant details, and adapting planning.

It is possible to observe the effects of context in human daily life from the most basic situation to the most complex one. For instance, a cup can be carried differently, modulated according to context, which might be the temperature of the cup, being in a hurry or the existence of obstacles. Another example could be the way of speech with an everyday seen neighbor, which can be affected by being in a hurry, our health or various psychological conditions. Many such examples can easily be drawn from our daily activities.

Understanding, learning and using context are also very important for robots since

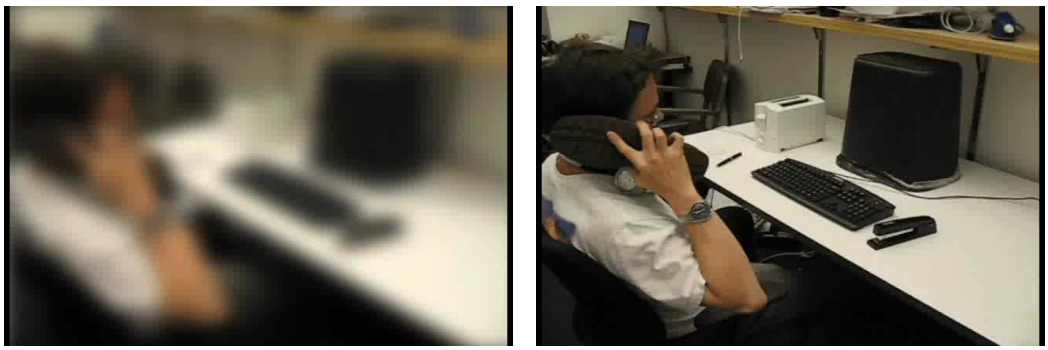


Figure 1.1: Contextual information plays a crucial role to achieve tasks such as object recognition [Figure source: [57]]

we expect from them to have similar cognitive abilities like us. Robots also should adapt themselves while fulfilling their tasks by taking into consideration contexts, sub-contexts, and super-contexts. For instance, a robot should be more careful while carrying a hot drink when there is a child around since the child may hit the robot accidentally. Moreover, a robot should be quieter while cleaning the home when someone is at sleep. As can be easily concluded, robots should understand the context and adjust their behaviors in terms of context to achieve their goals more properly.

1.1 Problem Definition

In this thesis, we address the following problems:

- What can we say about the structure of context?

  Even though context plays an important role in natural and artificial cognition, there is still more to discover regarding especially its structure. Is it flat? Shallow? Multi-dimensional? Such aspects affect how computational models should be developed, and therefore, should be investigated.

- Can we model or learn context incrementally?

  Robots need to learn context incrementally as humans do. When we consider a baby, she firstly learns family context as a social context and mostly indoor contexts especially home context as a spatial context. After some time, when

she meets new people other than her family and goes outdoors, she will learn different social and spatial contexts. This is how the contextual knowledge should be expanded in robots. In other words, robots need to learn contexts incrementally since they do not have a chance to know all possible situations at the beginning. Therefore, the methods focused on modeling context should also follow an incremental approach.

1.2 Contributions

The main contributions of the thesis are as follows:

- **The nature of context:** We conducted a survey examining the structure of context which is beneficial for many studies in many diverse areas. In Section 2.2, our observations about the structure of context is proposed in terms of its features, computational models, and we concluded that context has a hierarchical nature. Therefore, context modeling studies should use hierarchical structures to show contextual relations properly.

  This part of the thesis has been published as a technical report [17].

- **Building a deep incremental Boltzmann Machine for modeling context:**

  We developed two models for rule-based incremental construction of context in a generative deep model. The proposed methods determine the necessity of adding a new context and a context layer for each encountered scene without requiring any prior knowledge about the number of contexts, layers in the hierarchy or the nature of the input data. Moreover, the model does not demand the availability of the whole data at the beginning, and it is trained by using one train instance (i.e., scene) at a time.

- **A learning-based approach to incremental context modeling*:**

  We proposed formulating incremental context modeling as a learning problem.To the best of our knowledge, this is the first model that considers finding

* This study is conducted in equal contribution with İlker Bozcan

the number of contexts as a learning problem by using deep models rather than using rule-based approaches. One of the challenges in this problem is the absence of a dataset with a known and an exact number of contexts. To overcome this challenge, Latent Dirichlet Allocation (LDA) is used since it is a generative model, capable of producing artificial data for a given number of contexts. Therefore, artificial data produced by LDA contains information about the correct number of context for that data.

Recurrent Neural Networks are used to solve the learning problem by using states of the LDA model as an input with their labels indicating the necessity of incrementation. The deep network tries to handle this problem as a "sequence to label" problem and states of LDA symbolize the probability of objects given context and contexts given object. Therefore, the input is variable length depending on either the number of contexts or the number of objects and many-to-one recurrent networks employ for solving the binary decision problem (i.e., increment or not increment).

The contributions presented in this thesis are disseminated in the following studies:

- Fethiye Irmak Doğan, and Sinan Kalkan. A Deep Incremental Boltzmann Machine for Modeling Context in Robots. International Conference on Robotics and Automation (ICRA), 2018. (Submitted)
- Fethiye Irmak Doğan[†], İlker Bozcan[†], and Sinan Kalkan. A Learning Based Approach to Incremental Context Modeling in Robots. International Conference on Robotics and Automation (ICRA), 2018. (Submitted)
- Fethiye Irmak Doğan, and Sinan Kalkan. Hierarchical Context Modeling Using Incremental Deep Boltzmann Machines. Technical Report No: METU-CENG-TR-2017-01, Department of Computer Engineering, Middle East Technical University, 2017.
- Fethiye Irmak Doğan, and Sinan Kalkan. Bağlamın Hiyerarşik Doğası. Türkiye Robot Bilimi Konferansı (ToRK), 2016.

[†] Equal contribution

1.3 Organization

In Section 2, the work related to our study is examined by focusing on topic modeling and context modeling studies. Moreover, incremental and hierarchical approaches that are used for topic or context modeling are discussed in this section. The deficiencies of these works are also summarized.

In Section 3, our proposed models for rule-based context modeling, namely, incremental Restricted Boltzmann Machines (iRBM) and deep incremental Boltzmann Machines (diBM), are presented. The details of the construction steps are stated and the algorithms for building iRBM and diBM are described.

In Section 4, the construction steps of the learning based approach for incremental context modeling are introduced. Firstly, the step to generate the artificial dataset by using LDA is clarified then how to train and test the recurrent network by using this generated dataset is shown as a second step.

In Section 5, the experimental results from different models (i.e., incremental and non-incremental models in the literature, models in Section 3 and Section 4) are presented. These results are analyzed in terms of their accuracies, the number of contexts found by these models and their entropy change over time. Some document and scene datasets (i.e., artificial and real datasets) are used for this purpose.

In Section 6, the thesis is concluded by summarizing the proposed models. Then, the limitations of the models and future work are discussed in order to obtain better models.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we examine the structure of context and discuss related studies to our work. This chapter is divided into sections as follows: Hierarchical Nature of Context, Scene modeling, Topic Modeling, Context Modeling, Boltzmann Machines and summary of the sections.

2.1 What is Context?

Context can be described as *"the set of circumstances or facts that surround a particular event, situation, etc."* [2]. This description points out the importance of external and internal situations in context. Moreover, context is not only affected by these situations but also affects them.

In the study of Çelikkanat et al., context is defined as *"the totality of the information characterizing the situation of a cognitive system; e.g., it can include objects, persons, places, and temporally extended information related to ongoing tasks, but also information not directly related to these tasks."* [11]. This description indicates the importance of context for cognitive systems in terms of being related to not only ongoing actions but also the circumstances which may not have a direct relation to these actions. Additionally, the description of Çelikkanat et al. refers to the spatial, temporal and social properties of context, e.g, by including objects, persons and places, and these properties will be expressed in Section 2.2 in detail.

Both of these descriptions imply the relation between circumstances and contextual

information and these relations need to be modeled in order to obtain a proper contextualized scene model which is our main focus in this thesis.

2.2 Hierarchical Nature of Context

In order to model context, its structure should be examined in more detail. Therefore, we first start with reviewing the properties of context.

Context can be conveyed through different modalities. Social, spatial and temporal components of context are highlighted in the study of Zimmermann [69] where social properties may contain for instance being in a family or friendly environment and spatial and temporal properties correspond space and time aspects of context.

A context can include or be related to another context due to its social, spatial and temporal aspects. If we consider the "preparing breakfast" context, it may contain "family" context as a social component, "kitchen" as a spatial component and "morning" as a temporal component. Therefore, context is hierarchical and may contain each other with smaller "scope". Figure 2.1 demonstrates the hypothetical hierarchy representation of the home context. That hierarchy may be expanded by considering other aspects of home context.

McCarthy is one of the important pioneers who exploit context in artificial intelligence studies [37]. He emphasizes the important properties of context and the third property of context in his definition points that context may change dynamically, one context can be obtained from another context. Therefore, a context may enter the range of another context which can be expressed by the relational structure between contexts. Coping with the complicated dynamic structure of context may be facilitated thanks to these relational structures [30]. In other words, McCarthy points out the certain features of context and states that one context may be in the scope of another context which leads us to the hierarchical characteristic of context.

Barsalou is another influential pioneer who studied context. He emphasizes the effect of context in terms of episodic memory, object perception, and language compre-

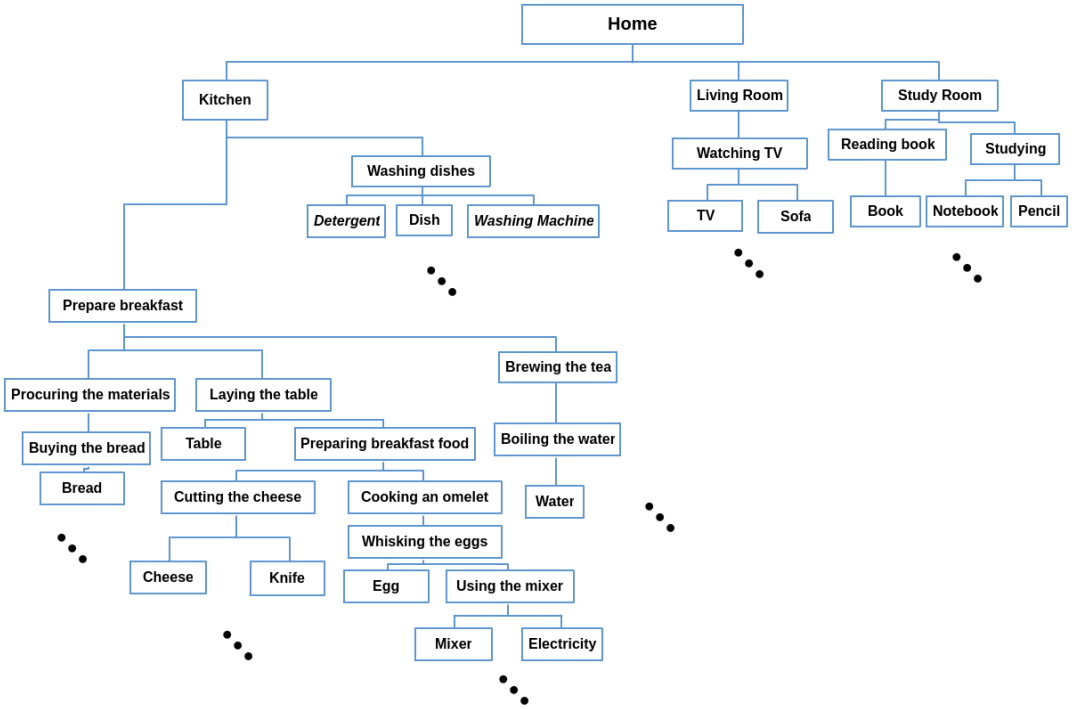


Figure 2.1: Hierarchic representation of home context [Figure source: [17]]

hension so he gives a huge emphasis on context for human cognition and cognitive processes (see, e.g., [66]). He also mentions that a situation may go on from an extensive space with a giant amount of time to a tiny space with a limited amount of time and this shows that stations are influenced by the "grain size". He also states that hierarchical association of situations with many levels of grain size creates contexts which points out the hierarchical disposition of context.

Human body, nature, society and lots of phenomena in the world have a hierarchical characteristic with multilayer organization. For instance, the hierarchical working system of human psychology is shown in the work of Saaty by exemplifying the hierarchical organization of decision making process [43].

Another support for hierarchical contextual relations comes from neuroscience. McKenzie states that, according to recent findings, the hippocampal neural system which connects the related memory parts has a hierarchical structure and this is just a small demonstration of the hierarchical working system of human body [38]. Moreover studies in neuroscience claim that hippocampus is the main portion of a contextual

formation, it makes crucial contributions to contextual coding [15, 42, 51]. Since hippocampus has a gradual hierarchical operating structure, hierarchical nature of context is backed thanks to the findings from neuroscience.

Hierarchy is essential for many other domains as well. According to Lane, social hierarchies which are composed of society, culture, and economy has more ambiguous character than physic-chemical (i.e., hierarchy between fundamental particle, nucleus, atom, and molecule) and biological (i.e., hierarchy between organelle, cell, organ, multicellular creature, population, species and ecosystem) hierarchies [31].

As explained above, hierarchical working systems are faced in many diverse areas from neuroscience to society. Therefore, it comes as a consequence that context should have a hierarchical formation by taking into account its social, spatial and temporal properties connecting to all these fields. In other words, since human life is led by social, spatial and temporal multi-layered relations, context should be hierarchical by containing all these attributes [37, 69].

In addition, there are also some hierarchical computational studies that focus on modeling context. Computer vision is one of the most popular areas that focuses on modeling context [14, 34, 54, 63, 64]. However, these efforts suffer from the absence of crucial features of context stated by McCarthy [37]. Despite all these shortages, the performances become higher thanks to the modulation of contextual knowledge in many difficult challenges in different areas such as object recognition and planning [4, 12, 27]. The details of these models are expressed in Section 2.5.1. In short, since hierarchical context modeling studies are shown improving the performance of computational models, context should model hierarchically to overcome uncertainties in the real-world problems.

In sum, by examining (i) the properties of context, (ii) the findings from neuroscience, biology, psychology, society and many other areas and (iii) computational context modeling performances, contextual relations should be taken into consideration for a proper solution for many challenging problems and context should be modeled hierarchically while trying to solve these problems.

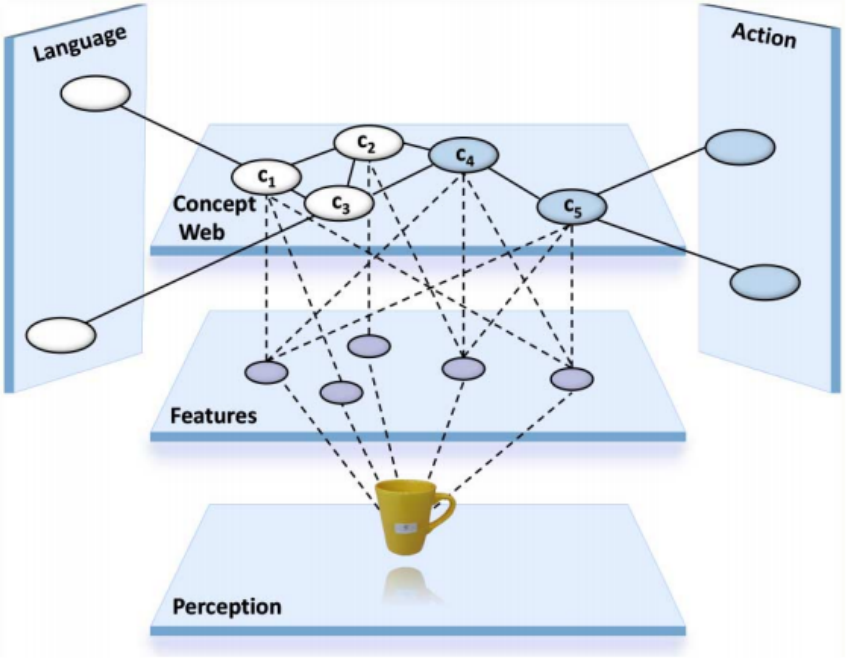


Figure 2.2: A schematic representation of the concept web which shows the concepts and their relations [Figure source: [10]]

2.3 Scene Modeling

Scene modeling corresponds to modeling the scene in terms of what it contains. Scene modeling is important for robots since they need to analyze the objects in the scene and make some interpretations about that scene.

There are a variety of models used for scene modeling in computer vision and robotics.

Markov Random Fields are used for contextually guided semantic labeling and building a concept web on a humanoid robot [4, 10]. In the study of Anand et al. [4], contextual relations to determine the object labels are captured by using these graphical models. Thanks to graphical models, geometric relationships of 3D scenes are obtained to predict where an object should be placed. In another study of Çelikkanat et al. [10], a concept web is built to represent concepts and conceptual relations in terms of co-occurrences of concepts by using Markov Random Fields. The schematic representation of their concept web can be seen in Figure 2.2.

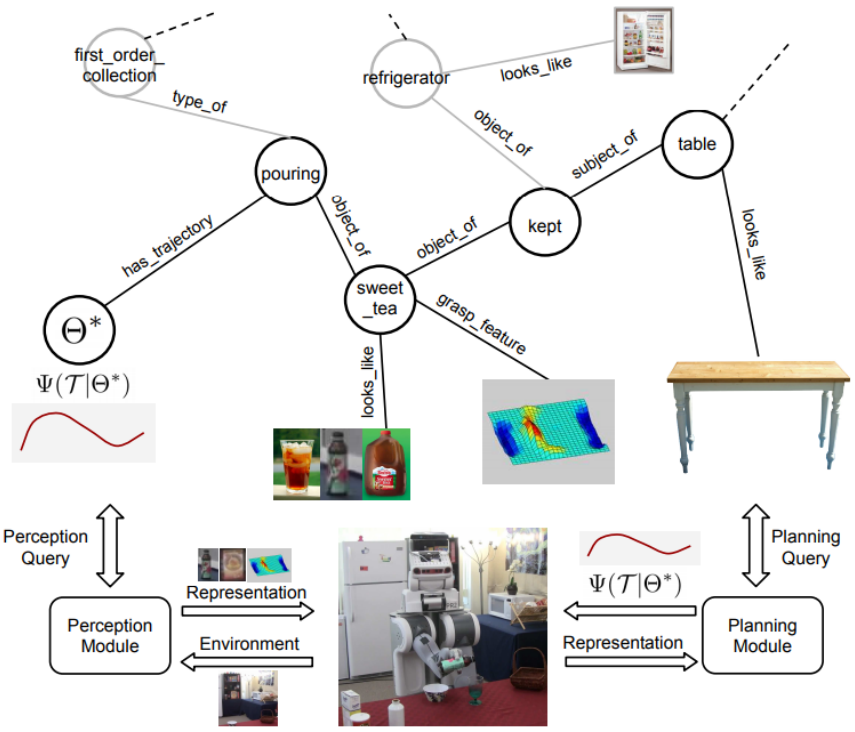


Figure 2.3: A demonstration of how a robot uses RoboBrain to perform tasks [Figure source: [46]]

Bayesian Networks are another model which are used for scene modeling. This networks are used for contextual reasoning in underwater robots [33] and for object detection in moving objects [48].

Latent Dirichlet Allocation which is actually created for document modeling is also used for scene modeling with some extensions such as Geometric LDA [41] or Spatial LDA [62].

Predicate Logic [25, 36] and Scene Graphs [7] are also used for scene modeling problems such as object recognition, distributed context assessment, and forming a domain specific language.

Ontology-based methods are also frequently used for scene modeling in order to model connections between objects and their relations [25, 46, 56]. Hwang et al. use ontology-based topic modeling for object recognition [25] and Saxena et al. employ these models to create RoboBrain which is a large-scale knowledge engine for

Table 2.1: Correspondence between context modeling and topic modeling

Context Modeling	Topic Modeling
a single scene	each document
all encountered scenes	corpus
objects in the scene	words in the document
context	topic

robots [46]. A demonstration which shows how a robot uses RoboBrain to perform tasks is presented in Figure 2.3. Moreover, ontology-based methods are also used for knowledge processing for autonomous robots [56].

2.4 Topic Modeling

Topic modeling is representing extensive input data in a more compact form without losing its informative statistical connections [6].

In natural language processing, many graphical models such as Hidden Markov Models are used to combine latent variables with hidden information in the input data for topic modeling.

These models are pursued by recent successful topic modeling approaches. For instance, Dumais et al. proposed organizing texts into semantic structures by using Latent Semantic Analysis [19] and Griffiths et al. tried to find the topics with Latent Dirichlet (LDA) Allocation [21]. LDA is a probabilistic generative model and details of this model are explained in Section 4.1.

Topic modeling approaches are generally used for modeling the documents and finding their topics. In addition, they are also employed for modeling contextual information [12]. In that kind of approach, each scene is interpreted as a document, objects in the scene can be regarded as words in the document and topics of the documents can be considered as contexts of the scenes. Correspondence between topic and context modeling can be observed in Table 2.1.

Topic modeling approaches (e.g. LDA) require the number of contexts at the beginning however that is not possible in a robotic scenario since the robot encounters each scene one by one and needs to increment the number of contexts after encountering a new scene belongs to a different context. Some extensions of LDA (Hierarchical Dirichlet Processes [55] or its nested version [40]) tried to solve the number of topic requirement but these methods either are not suitable for hierarchical data or require the existence of all the data at the beginning which is also impossible in our case. Some of these models are explained in the following section.

2.4.1 Hierarchical Topic Modeling

Since our observations in Section 2.2 state that context has a hierarchical structure, we need a topic modeling method which is suitable for modeling hierarchical documents.

One of the extensions of LDA for hierarchical topic modeling is proposed as Nested Chinese Restaurant Process (nCRP) [20]. The model assumes each topic is composed of distribution over words as in LDA and considers each node as a topic. Moreover, in order to generate a topic, a path from the root to leaf node needs to be followed. One of the drawbacks of this model is the assumption of predetermined hierarchy depth at the beginning which is impossible for a robot to know beforehand. Another lack of the model is about corresponding each document with a single path which means corresponding each document with a single topic, but a document is a mixture of topics in the real world.

An extension of nCRP is called as Nested Hierarchical Dirichlet Process (nHDP) and also used for modeling hierarchical topics [40]. This model constructs a tree by ordering the topics from more general to more specific and represent these topics from root to leaf nodes accordingly. This model can compete with the problem of associating each document with a single topic in nCRP by performing word-specific path clustering rather than document-specific paths. As can be concluded, a document needs to reach entire tree since it is considered as a mixture of different paths rather than a single path. Path selection difference for a document between nCRP and nHDP

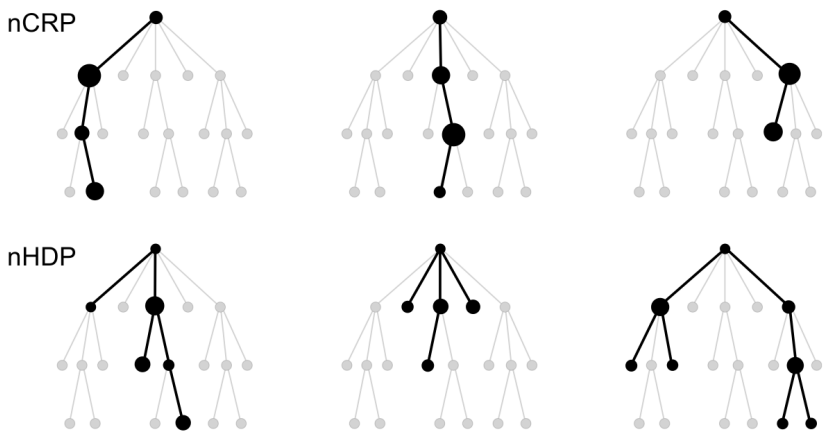


Figure 2.4: An example of path selection of a document for nested Chinese restaurant process (nCRP) and nested Hierarchical Dirichlet process (nHDP) [Figure source: [40]]

can be seen in Figure 2.4. In spite of this generalization capacity, this model still suffers from the predetermined number of nodes and layers in the hierarchy.

A top-down recursive model for hierarchical topic modeling is presented by Smith et al. [50]. This model splits and re-models all the documents recursively until corpus become too narrow to re-model. This model also suffers from the fixed number of topics in each level of the hierarchy.

Another study for hierarchical topic modeling is based on joining Chinese Restaurant process and distance dependent Chinese Restaurant process and it is used for joint segmentation and activity recovery [47]. This model handles the problem of requiring the number of topics but is not suitable for adapting its structure for newly encountered data, i.e., not proper for incremental learning.

Wang et al. also presented a hierarchical topic modeling approach where the hierarchy is generated based on the predetermined number of topics and hierarchy levels and this hierarchy may be varied by user interactions after the fixed construction [60]. To vary the top-down hierarchy, merging, removing or branching operations are possible by the user selections. However, this model is not appropriate for building a hierarchy in terms of a corpus, it requires the number of topics and hierarchy levels at the beginning.

Another bottom-up hierarchical topic modeling is proposed by Zavitsanos et al. [68]. In this study, the hierarchy is represented by a tree structure where the vocabulary and multinomial distributions over subtopics correspond to leaf and intermediate nodes of the tree respectively. This model handles the problem of the pre-determined number of topics and hierarchy layers but suffers from not being appropriate for incremental learning as in the study of Seiter et al. [47].

2.4.2 Incremental Hierarchical Topic Modeling

Since all the documents in the corpus may not be available at the beginning and they should be encountered one by one in real-world problems, we need a topic modeling approach not only hierarchical but also incremental.

Some extensions of LDA (e.g, [9]) tries to solve incremental hierarchical topic modeling problem but there are still a limited amount of incremental hierarchical topic modeling efforts. Some of these studies are explained as follows:

One of the studies that focused on incremental hierarchical topic modeling problem belongs to Hu et al. [24]. This model is composed of two steps. In the first step, a topic hierarchy is modeled recursively and in the second step, an incremental top-down hierarchical topic alignment algorithm is deployed in order to merge topics. This is determined in terms of the similarity matrix between subtopics and they are merged if the similarity is more than the threshold value. This model suffers from requiring exactly three levels in the hierarchy and number of topics beforehand.

Evolving hierarchical Dirichlet processes (EHDP) is another study based on online hierarchical topic clustering [61]. In that model, these hierarchical clusters may be born, evolve, branch and die-out over time. Moreover, clusters are evolutionary in this model that enables online learning and incremental construction of clusters. This model extends Chinese Restaurant Processes and Hierarchical Dirichlet Processes in the formation phase of the hierarchy and Gibbs sampling in the inference phase. This model solves the problems in the necessity for determining the number of clusters and number of branches at the beginning of the construction.

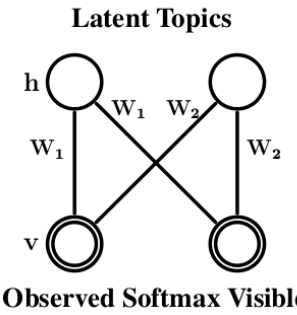


Figure 2.5: Replicated Softmax Model [Figure source:[22]]

2.4.3 Neural Networks for Topic Modeling

Different neural network architectures are used for topic modeling. Replicated Softmax, a Neural Autoregressive Topic Model (DocNADE), Deep Boltzmann Machines (DBMs) are explained in this section since these are the most commonly used ones for topic modeling.

Replicated Softmax Model is a graphical model which contains undirected two layers and rather than considering each document as a distribution over topics, it behaves each document as a binary distribution [22]. This model can be thought as a parameter sharing version of Restricted Boltzmann Machines (RBMs) and RBMs are explained in detail in Section 2.6.1. Replicated Softmax Model can be seen in Figure 2.5 where bottom layer represents the softmax visible neurons and top layer symbolizes the binary topic features.

Neural Autoregressive Topic Model (DocNADE) is a generative model and it extends the Replicated Softmax by adding hierarchical layers [32]. DocNADE has a binary tree structure and each leaf nodes corresponds to the words in the vocabulary. Comparison of Replicated Softmax and DocNADE is shown in Figure 2.6. As in Replicated Softmax Model, a visible unit called v_i represents a word and weights are shared between visible and hidden neurons in DocNADE as well.

Two-layered Deep Boltzmann Machines (DBMs) are used for topic modeling in the work of Srivastava et al. [53]. This work extends Replicated Softmax by putting one

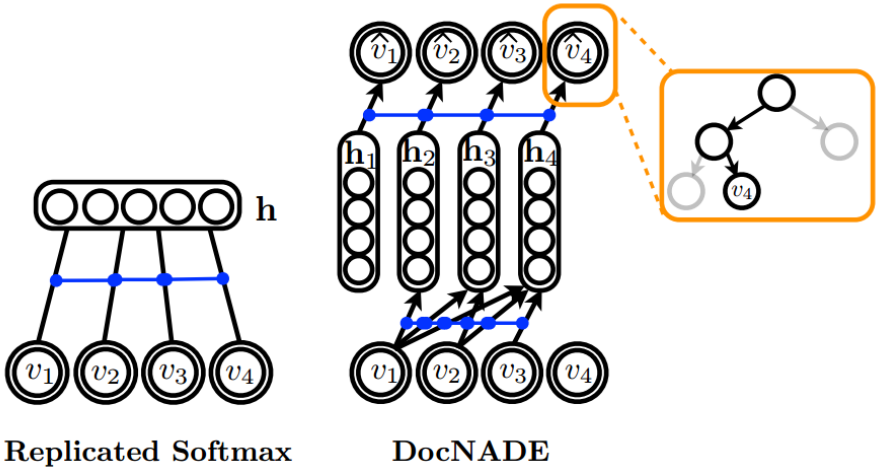


Figure 2.6: Replicated Softmax and DocNADE models [Figure source: [32]]

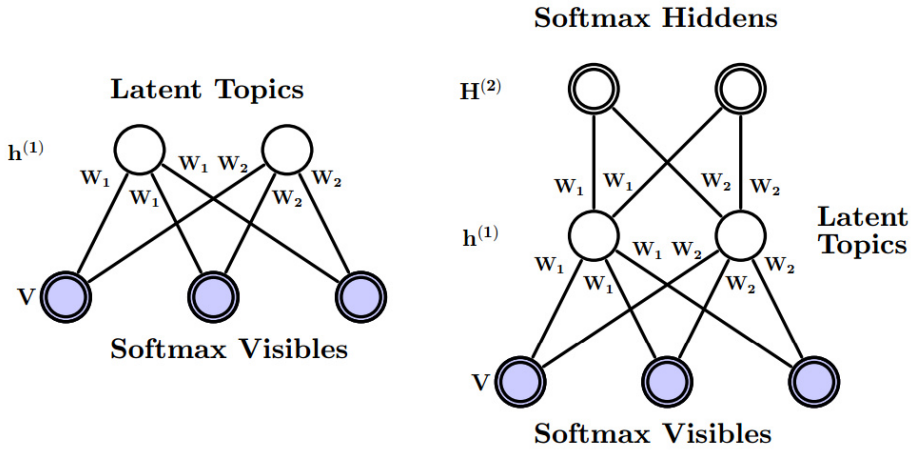


Figure 2.7: Replicated Softmax Model and Deep Boltzmann Machines [Figure source: [53]]

more hidden layer on top of the first one. Comparison of these two models can be seen in Figure 2.7.

All these three models suffer from requiring the number of visible and hidden neurons. In addition, DBM has a drawback because of requiring the depth of the hidden layers.

2.5 Context Modeling

McCarthy is recognized as the first scientist who described context in artificial intelligence view to model it [37]. His description is based on propositional logic and this work was followed by the similar propositional logic studies but these studies are generally based on fixed rules and connections between entities [8, 30]. However, counting all these fixed rules and connections between entities are impossible in real-world problems.

Context modeling is studied in many different fields in computer science such as computer vision, pattern recognition, and robotics.

In computer vision and pattern recognition studies, Torralba et al. focus on context-based object detection and recognition by using the correlation between the scenes and the objects by adapting graphical models such as Markov Random Fields [58, 59]. On the other hand, Marszalek et al. build a model to learn actions and contexts of the scenes by classifying actions using Support Vector Machines [35]. In these models, local interactions between predictions are used for building contextual knowledge in general.

In robotic studies, context is also used for facilitating complex real-world problems. In the study of Anand et al., they adapt graphical models to capture contextual relations in order to determine the object labels [4]. Jiang et al. also propose a graphical method to model context in order to choose a proper placing for an object in a scene [28]. An extension of a Bayesian Networks is utilized for contextual reasoning in underwater robots in the work of Li et al. [33].

2.5.1 Hierarchical Context Modeling

Since Section 2.2 suggest that context should model hierarchically, hierarchical context modeling studies should be examined in detail.

There are many promising attempts on modeling a hierarchical structure for context. For instance, Sun et al. describe the spatio-temporal context for action recognition problem in videos by using 3 levels of context hierarchy which is defined and designed by hand [54]. These layers are called point-level context, intra-trajectory context (trajectory transition descriptor), and inter-trajectory context. In another study, Wang and Ji construct a hierarchical structure to recognize events in surveillance videos [63, 64]. They also build a 3-layer hierarchy and emphasize that previous works generally extracted context from one layer, and there are not many studies that extract different contexts from different layers simultaneously. In his work, feature-level context, semantic-level context and prior-level context are obtained simultaneously and these obtained contexts are combined in order to detect events in surveillance video correctly. In these works, the hierarchy is not incorporated into a semantic hierarchy as shown in Figure 2.1, and it is statically structured. Moreover, context is considered only for specific modalities, making them restricted to a smaller scope.

In another interesting article, Choi et al. focus on object categorization using a hierarchical model [14]. They extract a hierarchical context model from a large database of object categories and model a graphical tree structure in terms of co-occurrence and spatial relations of objects. Dependencies between object categories and scenes emerge thanks to this tree structure. This model demonstrates the relations between objects in a hierarchical way and makes important contributions to object and scene recognition problems. Although this study creates a semantic hierarchy between objects, it only refers contextual relations between objects and it does not produce a general contextual structure. In addition, co-occurrence and spatial properties of objects are used as a prior model, which does not completely overlap with the dynamic property of contexts.

Lastly, Li et al. [34] work on recognizing human attributes such as gender, and cloth-

ing style, and they use a deep hierarchical context structure for this end. This deep context includes human contexts as well as background scene context. Convolutional Neural Networks (CNNs) are adopted for this purpose. In other words, CNN learns not only the scores of human bodies and human-specific attributes but also the score functions of deep hierarchical context. Especially, scoring cooperation while modeling the human and scene plays an important role to create the contexts related to humans and scenes. Combination of human and scene contexts generates a hierarchical context structure. The general context relations could not be shown in this study as well and the focus is placed on human attributes. Moreover, Li et al. show hierarchy as an association of human and scene oriented contexts, and they could not demonstrate any semantic hierarchy as mentioned in Figure 2.1.

2.5.2 Incremental Context Modeling

Context should model not only hierarchical but also incremental since robots interact with different circumstances but cannot know beforehand all the circumstances. Therefore, both hierarchical and incremental learning of contexts are needed in real-world problems.

Some works in computer vision and robotics tries to model context incrementally. In computer vision, Yu et al. [67] propose an incremental approach by using Restricted Boltzmann Machines and this model is also implemented in our work to compare our results. Ortiz & Baille [39] also build an incremental Restricted Boltzmann Machines by using reconstruction error as a cue in robotics. In robotics field, another study belongs to Çelikkanat et al. [12] and they suggest an incremental LDA model which uses maximum weight between a context and an object as a cue. All these models are rule-based and examine the errors or the entropies (perplexity) of the systems in order to decide when to increment the number of contexts. Moreover, they are not suitable for hierarchical construction.

Since this study is inspired by incremental LDA algorithm [12], details of this algorithm need to be examined in order to obtain a compact background. In that study, a

model tries to capture when to increment the number of contexts while encountering each scene one by one. The initial number of contexts is set as 1 and the model calculates the probabilities of words for a given context by using LDA. (The details LDA is explained in Section 4.1.) Then, confidence values for each object are calculated by assigning each object to a context which has the highest probability. They defined a C_{low} in order to represent set of objects whose confidence is less than a threshold and if there is an item in C_{low}, they increment the number of context by one and recomputed the probability distribution of objects given contexts. Incrementing the context count goes until C_{low} become $\varnothing$. Then, the model starts to wait for a new scene. Incremental LDA algorithm can be seen in Algorithm 1.

Algorithm 1: Incremental Latent Dirichlet Allocation algorithm (Source: [12])

```
initialize context count K ← 1.
for all encountered scenes do
    run K-Incremental Gibbs sampler with K
    while C_low ≠ ∅ do
        increment context count K ← K + 1
        run K-Incremental Gibbs sampler with K
    end
    output converged context assignments z⃗_N for the scene
end
```

2.6 General and Restricted Boltzmann Machines

In this section, a brief introduction is provided for General Boltzmann Machines [3] and Restricted Boltzmann Machines [45] which are used for constructing incremental Restricted Boltzmann Machines (iRBM) and deep Incremental Boltzmann Machines (diBM). Schematic comparison of Boltzmann Machines, Restricted Boltzmann Machines and Deep Boltzmann Machines [44] (i.e., which have more hidden layers in order to obtain a better latent representation of the data) is shown in Figure 2.8.

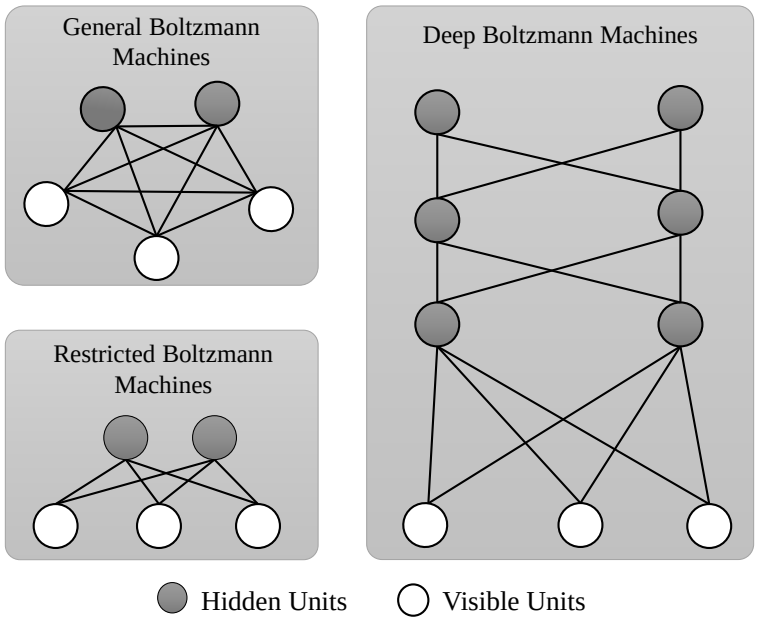


Figure 2.8: A schematic comparison of Boltzmann Machines, Restricted Boltzmann Machines and Deep Boltzmann Machines [Figure source: [18]]

2.6.1 General Boltzmann Machine

A Boltzmann Machine is a generative graphical model [3] which contains visible nodes represented with $\mathbf{v} = \{v_i\}_{i=1}^{V} \subset \{0,1\}^V$, hidden nodes represented with $\mathbf{h} = \{h_i\}_{i=1}^{H} \subset \{0,1\}^H$ and symmetrical edges between nodes represented with $W = \{w_{ij}\}$ with $w_{ij} \in \mathbb{R}$. All hidden and visible nodes are connected to each other without any restriction as it may be seen in Figure 2.9.

Boltzmann Machine is build based on a physical system and it tries to lower the energy function which can be defined as follows:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i<j} v_i w_{ij}^{vv} v_j - \sum_{i<j} h_i w_{ij}^{hh} h_j - \sum_{i<j} h_i w_{ij}^{hv} v_j. \tag{2.1}$$

Moreover, probability of activating a node is also depend on the energy function and can be defined as follows:

$$p(x = 1) = \frac{1}{1 + e^{\Delta E_x / T}}, \tag{2.2}$$

where x stands for a visible or a hidden node, ΔE_x represents for the change in energy and T corresponds to the temperature of the system.

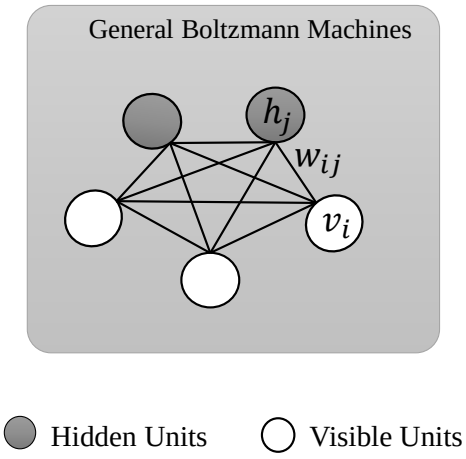


Figure 2.9: A schematic representation of General Boltzmann Machines

In Boltzmann Machines, visible nodes are modeled in terms of latent variables:

$$P(\mathbf{v}) = \sum_{H} P(\mathbf{v}, h_j) \tag{2.3}$$

Suppose, distribution over the training set is denoted by $P^+(V)$ and distribution when the Boltzmann Machines reached the thermal equilibrium is represented by $P^-(V)$. In order to train the Boltzmann Machine, $P^+(V)$ should be approximated to $P^-(V)$ where V is the visible units of Boltzmann Machine. For this purpose, a similarity measure is proposed based on the Kullback Leibler Divergence as follows:

$$G = D_{KL}(P^+(V), P^-(V)) = \sum_{v} P^+(v) \ln \frac{P^+(v)}{P^-(v)}, \tag{2.4}$$

where the summation is calculated all the possibilities of V. Since G is a weight function, gradient descent on G can be used for updating weights. In order to update the weights, the following is used:

$$w_{ij} \leftarrow w_{ij} - \frac{\partial G}{\partial w_{ij}}, \tag{2.5}$$

$$\frac{\partial G}{\partial w_{ij}} = \frac{1}{R} \times [p_{ij}^+ - p_{ij}^-], \tag{2.6}$$

where R is learning rate and p_{ij}^+ and p_{ij}^- are the probability of having all the units on at the thermal equilibrium in two phases (i.e., (i) positive phase where the visible units

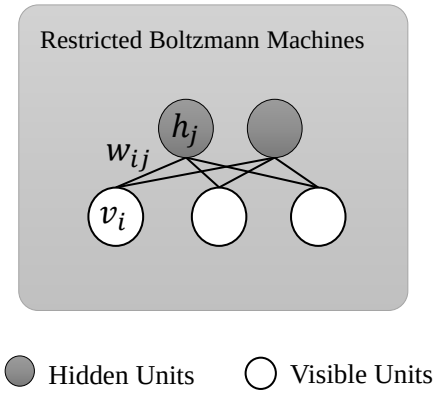


Figure 2.10: A schematic representation of Restricted Boltzmann Machines

are clamped with training data randomly, (ii) negative phase where the network runs freely).

2.6.2 Restricted Boltzmann Machines

The training and inference are slow and limited in Boltzmann Machines. To overcome these limitations, the connections between hidden to hidden and visible to visible units are discarded and only visible to hidden unit connections are kept in Restricted Boltzmann Machine (RBM) [45]. The structure of RBM can be observed in Figure 2.10.

In order to train Restricted Boltzmann Machines, firstly data should be clamped to visible units and then updating the hidden and visible units should continue until the equilibrium. However, since reconstructing and re-estimating visible and hidden units for one step also gives an idea about the way of the gradient, a shortcut is possible and this makes training much faster and easier in RBMs. This shortcut is shown in Figure 2.11. This process can be summarized in three steps:

Positive Phase: (i) Data is clamped to the visible neurons $\mathbf{v}$ (ii) Hidden neurons $\mathbf{h}^0$ are activated (iii) Average joint activations $< v_i h_j >^0$ are calculated.

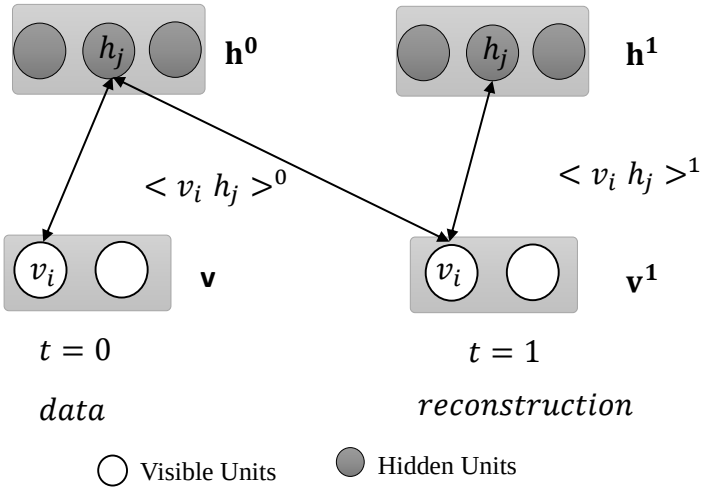


Figure 2.11: A schematic representation of training phases of Restricted Boltzmann Machines

Negative Phase: (i) Visible neurons $\mathbf{v}^1$ are reconstructed from $\mathbf{h}^0$ (ii) Hidden neurons $\mathbf{h}^1$ are re-estimated from $\mathbf{v}^1$ (iii) Average joint activations $< v_i h_j >^1$ are calculated.

Weights update: $w_{ij} \leftarrow w_{ij} + \epsilon \times (< v_i h_j >^0 - < v_i h_j >^1)$.

2.7 Summary

- Incremental topic or context modeling studies in the literature are generally rule-based approaches and look for the entropy [39] or the error [67] of the system in order to decide when to increment.
- There are no models that focus on learnability of incrementing the number of topics or contexts.
- Existing studies are either not hierarchical or use a predetermined number of hidden nodes.
- Some of the hierarchical models assume the availability of all the data [40, 55] at the beginning and that is impossible for a robotic task.

CHAPTER 3

INCREMENTAL RESTRICTED BOLTZMANN MACHINES (iRBM) AND A DEEP INCREMENTAL BOLTZMANN MACHINE (diBM)

In this chapter, we propose two novel models in order model context incrementally and hierarchically. The first model is called an Incremental Restricted Boltzmann Machines (iRBM) which solves the problem of requiring the number of contexts at the beginning and a Deep Incremental Boltzmann Machines (diBM) which focuses on building a hierarchical structure dynamically in order to model context.

This work is submitted to International Conference on Robotics and Automation (ICRA 2018) which is still under evaluation [18].

3.1 Incremental Restricted Boltzmann Machines (iRBM)

One of the first improvement we have progressed is extending Restricted Boltzmann Machine (RBM) in an incremental way. Detailed information about General Boltzmann Machines and Restricted Boltzmann Machines can be found in Section 2.6.1. As shown in Section 2.6.2, RBM has a fixed number of nodes in its hidden layers so a fixed number of contexts since the contextual information is represented as a latent variable in our approach. The necessity of knowing the number of hidden neurons at the beginning is not suitable for building contextual information in an incremental way on robots. Therefore, we started by building an incremental model by extending RBM in order to solve this problem.

Studies in the literature utilize the entropy of the system [39] or reconstruction error

[67] in order to determine when to increment. However, our model is incremental based on confidence values of each visible unit:

$$c_v \leftarrow \max_j w_{vj}. \tag{3.1}$$

Equation 3.1 forces each visible unit to be connected to a hidden unit and it also measures how strongly connected to the maximum weightily hidden unit. If the measured maximum weight from a visible unit to a hidden unit is low than a threshold value, then it can be concluded that the system does not found a hidden neuron to represent that visible where visible neurons correspond to objects/words and hidden neurons corresponds to contexts/topics in our documents/sceneries. In other words, this means the system does not found a strong context to represent for that object yet.

Moreover, a baseline confidence value needs to be calculated in order to measure the threshold value for the system. This baseline confidence value is responsible for the whole system confidence with the current hidden units and it is represented with $c_m^{|\mathbf{h}|}$. Softmax function is also used in order to normalize baseline confidence and give a smoother characteristic to that value. Baseline confidence is calculated as follows:

$$c_m^{|\mathbf{h}|} \leftarrow \frac{1}{Z_0} \exp\left(\min_v c_v\right), \tag{3.2}$$

where Z_0 corresponds to the summation of the confidence values for all visible neurons. It is represented as:

$$Z_0 \leftarrow \sum_v \exp(c_v). \tag{3.3}$$

Our model is encountered each scene ($\mathbf{v}$) one by one and after some time it fails to represent all the objects with the present contexts. In other words, $p(\mathbf{v})$ decreases because of the decrease in the current confidence value which means inefficient representations of visible units. The current confidence value is represented by

$$c_m^{curr} \leftarrow 1/Z_0 \exp\left(\min_v c_v\right). \tag{3.4}$$

When c_m^{curr} lies under the baseline confidence (i.e., $c_m^{|\mathbf{h}|}$), the number of hidden neurons should be incremented by one in order to represent that visible neuron and enhance the representation scope of the system. This condition can be shown as follows:

$$c_m^{curr} < t \times c_m^{|\mathbf{h}|}. \tag{3.5}$$

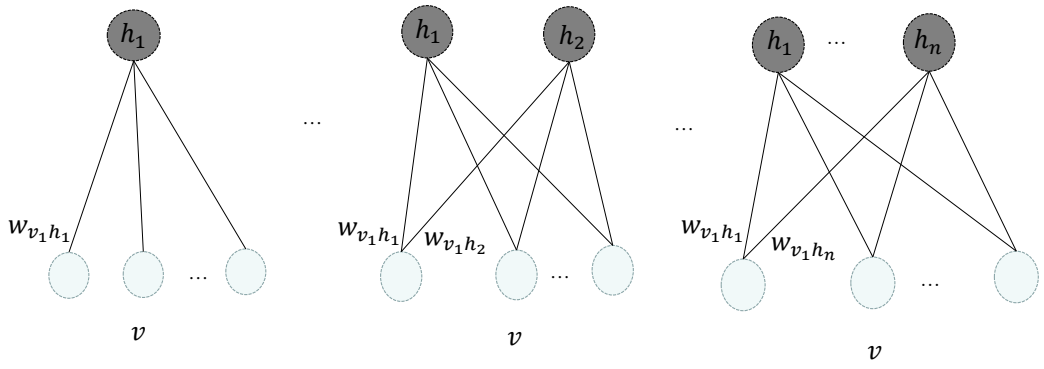


Figure 3.1: An illustration of construction of iRBM

In short, when Equation 3.5 is satisfied, a new context is added as a hidden neuron to our model in order to represent the object which the system is insufficient to represent. In this condition, t shows the scaling factor in order to control the system patience for adding a new hidden neuron.

Since Equation 3.2 and 3.4 contain Z_0s on both sides, Equation 3.5 can be simplified by dropping out the Z_0s on both sides.

An illustration of construction for iRBM can be observed in Figure 3.1.

Moreover, weights from newly added hidden neuron to visible neurons are not randomly initialized. Weights of each visible neuron to the new hidden neuron are inversely proportional to weights to the other hidden neurons. It can be formulated as follows:

$$w_{ik} \leftarrow \left(\sum_{j=1}^{|\mathbf{h}|-1} w^{ij} \right)^{-1} . \tag{3.6}$$

As can be seen in Equation 3.6, weights from each visible unit v_i to h_k are initialized by using an inverse proportion of weights from that visible unit to other hidden units. v_i can be assumed as strongly related and represented by the existing hidden neurons if that sum is large therefore a small weight w_{ik} should be initialized between v_i and h_k. However, v_i can be seen as not adequately represented by existing hidden neurons if this sum is small, so a higher weight should be initialized for w_{ik} in that case.

Summarization of iRBM algorithm can be seen in Algorithm 2.

Algorithm 2: Incremental RBM for a new scene. Initially, there is only one hidden node, i.e., $|\mathbf{h}| = 1$, and t^{iRBM} (patience of the model) is set to $\exp(-0.5)$.

Input:

- $\mathbf{s}$: A new scene (i.e., a $\mathbf{v}$ vector, s.t. $\mathbf{v}_i = 1$ if $\mathbf{s}$ contains object with label i)
- W, $|\mathbf{v}|$, $|\mathbf{h}|$: Current model

Output: W: Updated model

Clamp $\mathbf{v}$, estimate $\mathbf{h}^0$ and calculate $< v_i h_j >^0$ ▷ Positive phase

Reconstruct $\mathbf{v}^1$ from $\mathbf{h}^0$, estimate re-estimate $\mathbf{h}^1$

Calculate $< v_i h_j >^1$ ▷ Negative phase

$w_{ij} \leftarrow w_{ij} + \epsilon \times (< v_i h_j >^0 - < v_i h_j >^1)$ ▷ update weights

$c_v \leftarrow \max_j w_{vj}$ ▷ calculate confidence for visible neurons

if $\exp\left(\min_v c_v\right) / Z_0 < t^{iRBM} \times c_m^{|\mathbf{h}|}$ **then**

  Add a new hidden neuron, let k be its index

  $w_{ik} \leftarrow \left(\sum_{j=1}^{|\mathbf{h}|-1} w^{ij}\right)^{-1}$ ▷ Initialize new weights

  $Z_0 \leftarrow \sum_v \exp(c_v)$

  $c_m^{|\mathbf{h}|} \leftarrow \exp\left(\min_v c_v\right) / Z_0$ ▷ Update baseline confidence for new $\mathbf{h}$

end

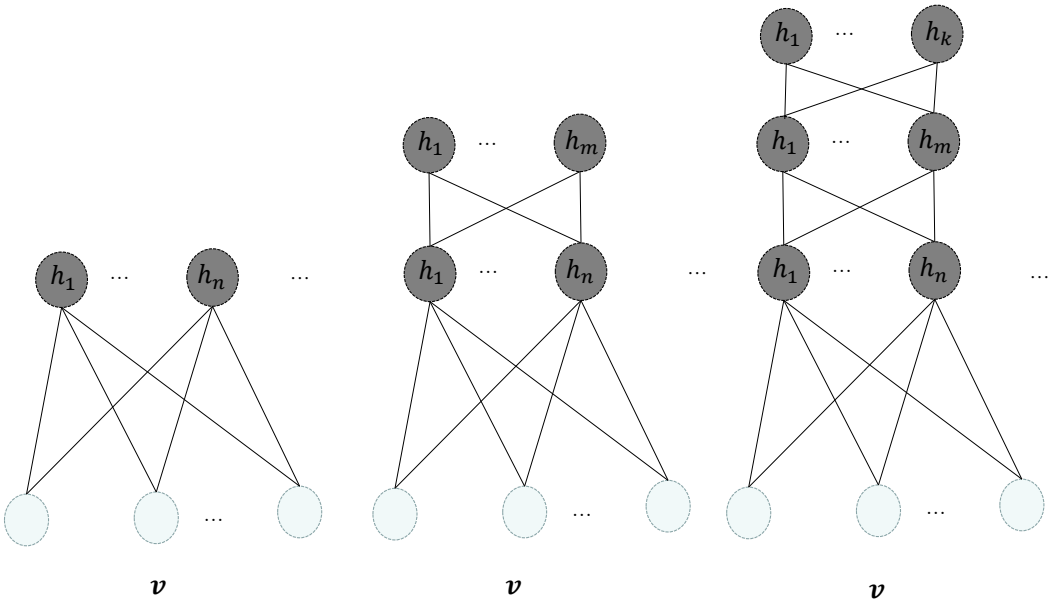


Figure 3.2: An illustration of construction of Stacked iRBM

3.1.1 Stacked Incremental Restricted Boltzmann Machines (Stacked iRBM)

In order to represent super-contexts and sub-contexts, stacking mechanism is added to iRBM model. For this purpose, after all the inputs are assumed to be encountered and iRBM is constructed, all the inputs are run, and their hidden activations are stored to use them as an input for another iRBM. In other words, hidden activations of an iRBM are used as an input for another iRBM. This stacking is illustrated in Figure 3.2. An iRBM layer construction is finished when all the inputs are encountered and another iRBM layer is added on top if final iRBM model contains any close contexts (i.e, hidden neurons since contexts are represented with latent variables).

We calculate a baseline value (d^s_{base}) for last iRBM layer ($iRBM^t$) by using previous one in the stack ($iRBM^{t-1}$) as follows:

$$d^s_{base} \leftarrow \min_{h_i,h_j \in iRBM^{t-1}} dist(h_i, h_j), \tag{3.7}$$

where $dist(h_i, h_j)$ calculates the distance between between h_i and h_j. It can be defined as:

$$dist(h_i, h_j) = \frac{1}{2}[d_{KL}(\mathbf{w}^i, \mathbf{w}^j) + d_{KL}(\mathbf{w}^j, \mathbf{w}^i)], \tag{3.8}$$

where $\mathbf{w}^i =< w_{ki} >$ and $\mathbf{w}^j =< w_{kj} >$ are the representations of the weights

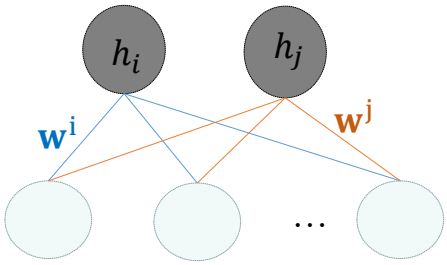


Figure 3.3: Representation of $\mathbf{w}^i$ and $\mathbf{w}^j$. Different colored edges represent the vector of weights connecting h_i and h_j to the previous layer's nodes. [Best viewed in color]

connecting h_i and h_j to the previous layer's nodes. These vectors can be seen in Figure 3.3.

The distance between hidden neurons in the final iRBM layer ($iRBM^t$) is defined as:

$$d^s_{curr} \leftarrow \min_{h_i,h_j \in iRBM^t} dist(h_i, h_j). \tag{3.9}$$

When training $iRBM^t$ is finished, d^s_{curr} and d^s_{base} are compared as follows:

$$d^s_{curr} < d^s_{base} + e^s, \tag{3.10}$$

where e^s represents the extensibility of the model while measuring the closeness of hidden neurons. If there are similar hidden neurons in $iRBM^t$, another iRBM ($iRBM^{t+1}$) is trained by using hidden activations of $iRBM^t$ as an input for $iRBM^{t+1}$ to represent these two similar hidden neurons. Moreover, d^s_{base} should be updated as follows:

$$d^s_{base} \leftarrow \min_{h_i,h_j \in iRBM^t} dist(h_i, h_j). \tag{3.11}$$

The whole algorithm for stacked iRBM model is presented in Algorithm 3.

Algorithm 3: The algorithm for adding an iRBM ($iRBM^t$) to the stack. e^s (extendibility of the model) is empirically set to 1. n is number of scenes in the corpus.

Input: $\mathbf{i}^{t-1}$: hidden activations of $iRBM^{t-1}$ composed of $< i_1^{t-1}, i_2^{t-1}, .., i_n^{t-1} >$.

Output: $\mathbf{i}^t$: hidden activations of $iRBM^t$ composed of $< i_1^t, i_2^t, .., i_n^t >$.

Train $iRBM^t$ using each $i_m^{t-1} \in \mathbf{i}^{t-1} for\ m \leqslant n$ and behaving i_m^{t-1} as an encountered scene in Alg. 2

$d_{curr}^s \leftarrow \min_{h_i, h_j \in iRBM^t} dist(h_i, h_j)$

if $d_{curr}^s < d_{base}^s + e^s$ **then**

  $d_{base}^s \leftarrow \min_{h_i, h_j \in iRBM^t} dist(h_i, h_j)$

  Add a new iRBM ($iRBM^{t+1}$) to the stack and train it by using hidden activations of $iRBM^t$

3.2 Deep Incremental Boltzmann Machines (diBM)

In order to represent a hierarchical nature of context, iRBM model is extended. In stacked iRBM model all scenes are assumed to be encountered in order to build other hierarchical layers but in this model for each encountered scene, model dynamically determines to add a new hidden layer and/or a hidden neuron. An overview of this model can be seen in Figure 3.4.

A baseline confidence r_f is used in order to decide when to add a hidden layer on top of the final hidden layer f. r_f is calculated when layer f has absolutely 2 hidden neurons as can be seen in the first state of diBM in Figure 3.5. Baseline confidence is computed as follows:

$$r_f \leftarrow d(h_i, h_j), \qquad \text{for } h_i, h_j \in \mathbf{h}^f, \tag{3.12}$$

where $d(h_i, h_j)$ is very similar to $dist(h_i, h_j)$ function in stacked iRBM and used for defining Kullback Leibler Divergence of h_i and h_j in terms of their weights. It can be

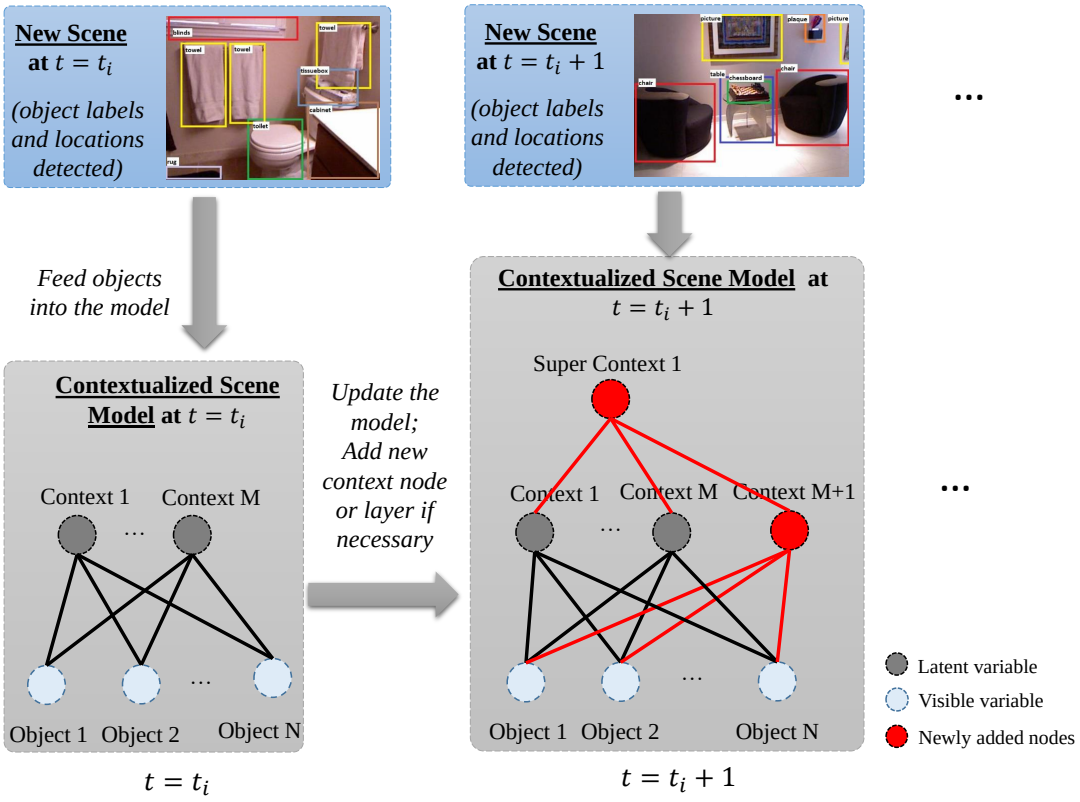


Figure 3.4: An overview of diBM model. diBM obtains one scene at a time, and updates the model by adding a new context node and/or a context layer in order to represent close context in a upper layer in the hierarchy. [Figure source: [18]]

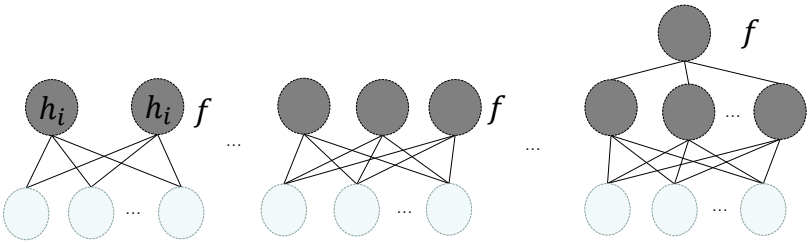


Figure 3.5: Different phases of diBM which has one hidden layer with two neurons, one hidden layer with three neurons and two hidden layers with one neuron in the final layer respectively

defined as:

$$d(h_i, h_j) = \frac{1}{2}[d_{KL}(sm(\mathbf{w}^i), sm(\mathbf{w}^j) + d_{KL}(sm(\mathbf{w}^j), sm(\mathbf{w}^i))], \tag{3.13}$$

where $\mathbf{w}^i =< w_{ki} >$ and $\mathbf{w}^j =< w_{kj} >$ corresponds the weights which connects h_i and h_j to the previous layer nodes. Representation of these weights can be seen in Figure 3.3.

As can be noticed, the only difference between $d(h_i, h_j)$ and $dist(h_i, h_j)$ is using $sm(.)$ function in $d(h_i, h_j)$ which stands for vector-defined softmax function in order to normalize weight vectors. Softmax function can be formulated as:

$$sm(\mathbf{w})_i = \frac{\exp(w_i)}{\sum_j \exp(w_j)}. \tag{3.14}$$

When the number of hidden neurons for the final hidden layer f is incremented by using the rules of Algorithm 2 and more than two (i.e., when $|\mathbf{h}^f| > 2$) as shown in the second state of diBM in Figure 3.5, the current confidence r_f^{curr} for the final layer is represented as:

$$r_f^{curr} \leftarrow \min_{h_i, h_j \in \mathbf{h}^f} d(h_i, h_j). \tag{3.15}$$

When r_f^{curr} is small, hidden neurons (i.e., contexts) are close to each other and represent the similar objects or contexts. When this value is smaller than the threshold which is determined dynamically, a new hidden layer with one hidden neuron should be added. That layer would be an upper layer of f^{th} layer which is called as $(f+1)^{th}$ layer of the model. This condition is demonstrated in the third state of diBM in Figure

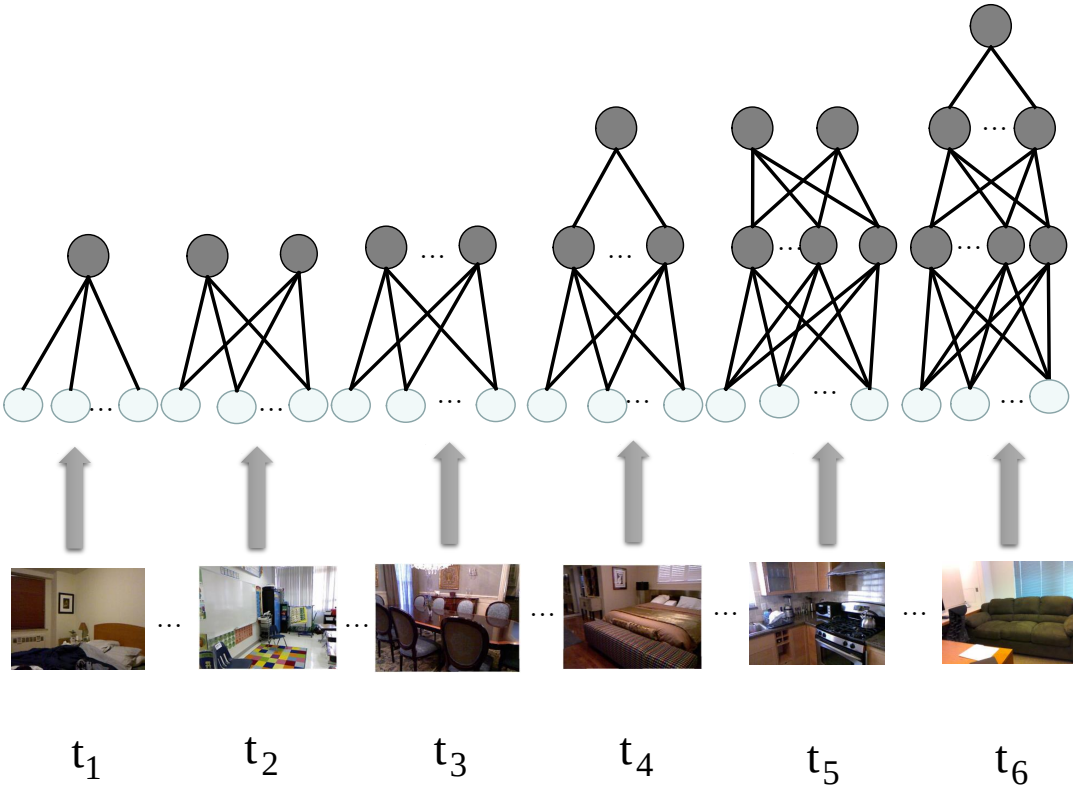


Figure 3.6: An illustration of construction of diBM after encountering different scenes

3.5 and can be shown as follows:

$$r_f^{curr} < t^{diBM} r_f. \tag{3.16}$$

In Equation 3.16, t^{diBM} stands for a scale to control the tolerance of the system for incrementing the number of hidden layers by one with a single hidden unit. Weights from hidden neurons in the f^{th} hidden layer to the single hidden neuron in the $(f+1)^{th}$ hidden layer is randomly initialized.

diBM construction phases after being fed by different scenes are illustrated in Figure 3.6.

Summary of diBM construction steps can be examined in Algorithm 4.

Algorithm 4: The algorithm for deep incremental BM (diBM). $\mathbb{R}$ initially contains one hidden layer with one hidden neuron. t^{diBM} (patience of the model) is empirically set to 0.1.

Input:

- $\mathbf{s}$: A new scene (i.e., a $\mathbf{v}$ vector, s.t. $\mathbf{v}_i = 1$ if $\mathbf{s}$ contains object with label i)
- $\mathbb{R} = \{\mathbf{R}^0, ..., \mathbf{R}^l\}$: The current (latent) hierarchy, with $\mathbf{R}^i = \{\mathbf{h}^i, \mathbf{W}^i\}$ being the hidden neurons and the weights of layer i.

Output: $\mathbb{R}$: The updated hierarchy.

Update each $\mathbf{R}^i \in \mathbb{R}$ using Alg. 2, adding new hidden neurons if necessary
Let $\mathbf{R}^f$ be the last layer, and $\mathbf{h}^f$ be its hidden neurons
If $|\mathbf{h}^f| < 2$, set the last layer's baseline confidence, r_f, to 0.
if *Hidden neurons in* $\boldsymbol{R}^f$ *is incremented, and* $|\mathbf{h}^f| = 2$ **then**
    $r_f \leftarrow d(h_i^f, h_j^f)$, for $h_i, h_j \in \mathbf{h}^f$
else if $\left[\min_{h_i, h_j \in \mathbf{h}^f} d(h_i, h_j)\right] < (t^{diBM} \times r_f)$ **then**
    $\mathbf{R}^{f+1} \leftarrow$ a new incremental RBM layer with one node
    $\mathbb{R} \leftarrow \mathbb{R} \oplus \mathbf{R}^{f+1}$ ▷ Add new layer to diBM
    $r_f \leftarrow 0$

CHAPTER 4

A LEARNING BASED APPROACH TO INCREMENTAL CONTEXT MODELING

Incremental learning of contextual information is very important for both humans and robots since they construct the contextual knowledge over time. However, models in the literature for incremental context modeling is generally based on some pre-determined rules. They decide to increment based on entropy (e.g. [39]) or the error value of the system (e.g. [67]). In our model, we focus on when to increment the number of contexts as a learning problem and use Latent Dirichlet Allocation (LDA) for generating data and built a Long-Short Term Memories (LSTM) in order to learn when to increment the number of contexts.

In our approach, we assume that each object may be observed in different contexts and contexts are modeled as latent variables of the model as in the model of Çelikkanat et al. [12]. The overview of our model can be seen in Figure 4.1 where the system updates the contextualized scene model by using lda and gives these model to LSTM as an input and tries to estimate necessity of incrementing the number of contexts by using LSTM.

This work is cooperated with İlker Bozcan with equal contribution and submitted to International Conference on Robotics and Automation (ICRA 2018) which is still under evaluation [16].

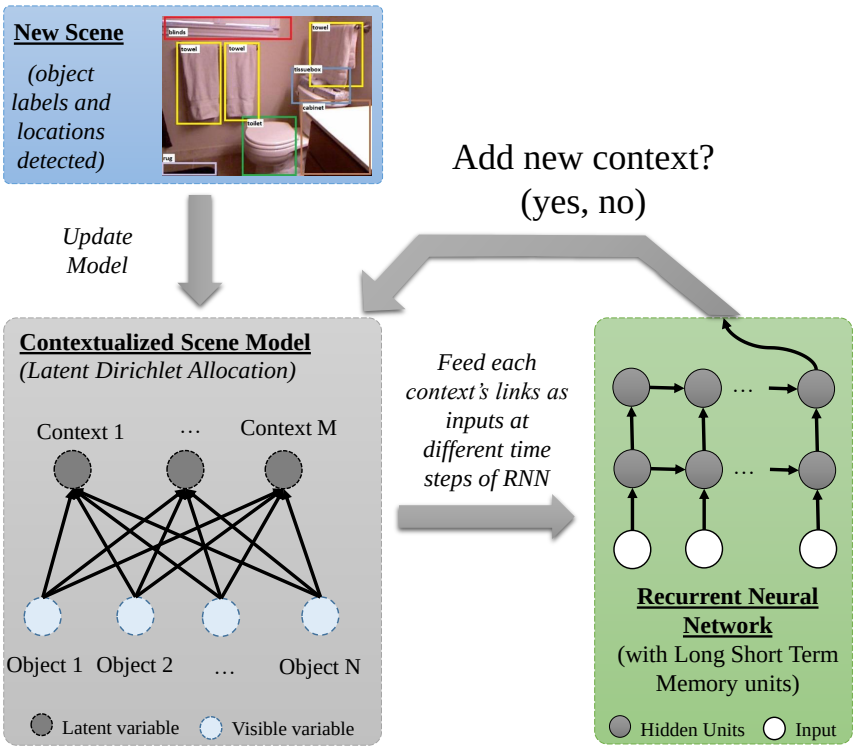


Figure 4.1: An overview of how incremental context modeling is addressed as a learning problem. When the model encounters the scenes, labeled objects are detected and the Latent Dirichlet Allocation Model is updated. Then, states of the LDA model is provided as an input to the Recurrent Model in order to estimate the necessity of incrementing the number of contexts. [Figure source: [16]]

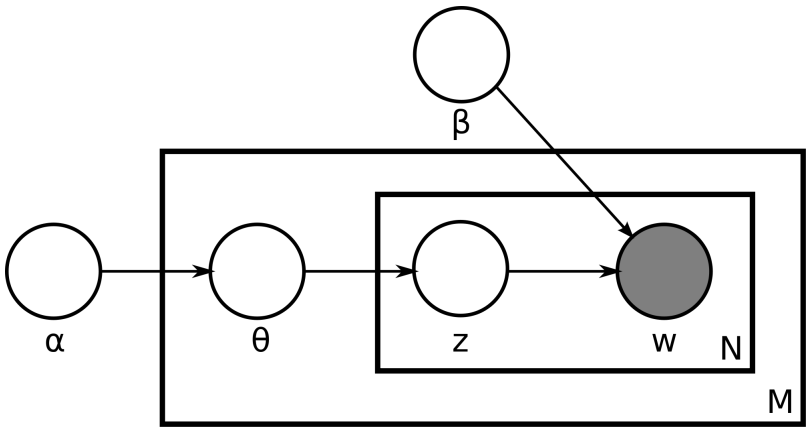


Figure 4.2: Graphical representation of Latent Dirichlet Allocation [Figure source: [6]]

4.1 Contextualized Scene Modeling with Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation is a generative model and it is generally applied documents for topic modeling [21]. LDA will be introduced with a topic modeling perspective in this section and the correspondence between topic modeling and context modeling is explained in Section 2.4 and Table 2.1.

As it is presented in the work of Blei et al [6], LDA model makes the following assumptions: (i) a document $d \in \mathbb{D}$ is composed of a set of words $w_1, ..., w_N$ where $\mathbb{D}$ represents the corpus (i.e., whole documents in the dataset). (ii) Words are sampled from a fixed size vocabulary (i.e., $w_i \in \mathbb{W}$ for vocabulary of size $|\mathbb{W}|$). (iii) A document can be treated as a mixture of a fixed number of topics $z_1, ..., z_k$. It can be described as $z_t \in \mathbb{Z}$ and $|\mathbb{Z}| = k$ where k is the total number of topics. Therefore, a document can be expressed with a probability of each topic and this can be denoted as $p(z_t|d_i)$. (iv) A topic, on the other hand, can be considered as a mixture of words (i.e., $w_1, ..., w_N$) in the vocabulary and it can be reflected as $p(w_j|z_t)$.

LDA tries to infer these probabilities by using all the documents in the corpus $\mathbb{D}$. Moreover, probability of generating a corpus can be defined as follows:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha)(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn}, \beta))d\theta_d \tag{4.1}$$

The variables in Equation 4.1 can be observed in graphical representation of LDA in Figure 4.2. In this equation, M and N refers to number of documents and words, α and β corresponds to Dirichlet Parameters of document-topic and topic-word distributions, θ_d shows per document topic proportions, z_{dn} corresponds to per word topic assignment, w_{dn} shows the observed word, D symbolizes the corpus (collection of M documents) and d represents each document where $d \in D$ (sequence of N words).

4.2 Dataset Collection

The existing datasets do not give an exact information about the number of contexts. Even if they are labeled and categorized, it is possible to infer more contexts since

high-level contexts may be discarded in these scene classification datasets. For instance, in that kind of dataset, there may be a home and an office contexts, but home-office context may not be taken into consideration. Even if some samples belong to a home-office context in that dataset, those samples may be labeled as either home or office.

In order to solve dataset problem, Latent Dirichlet Allocation (LDA) [21] is used. LDA is a generative model and it enables to generate artificial documents from a different number of contexts. These documents can be used for context modeling by considering each document d_i as a scene s_i, a word w_i as an object o_i, a topic t_i as a context c_i. Since we are trying to find the distribution between the objects and the contexts and necessity of adding a new context, using an artificially generated data should not cause any problem. Generating an artificial dataset using Dirichlet Distribution only requires the Dirichlet Distribution between objects and contexts but it seems very rational requirement since Dirichlet Distribution can approximate different distributions, many high-level categories and natural phenomena.

As can be observed in Figure 4.3, artificially generated dataset and the real dataset (SUN-RGBD dataset [52]) yielded similar distributions in terms of context-object frequencies. For this purpose, α which shows the topic mixture distributions per document and β which represents the word distributions per topic selected as 0.9 and 0.01 respectively. In other words, thanks to Dirichlet parameter selections, we obtained a distribution on the artificial dataset which yields similar to the real dataset.

In order to generate artificial dataset, followings steps should be pursued:

- Dirichlet Parameters should be chosen:

  α : Parameter of scene-context distribution which describes the environment

  β : Parameter of context-object distribution which corresponds the likeness of objects for topics

- Each context c_i should be generated for each index i as the probability dis-

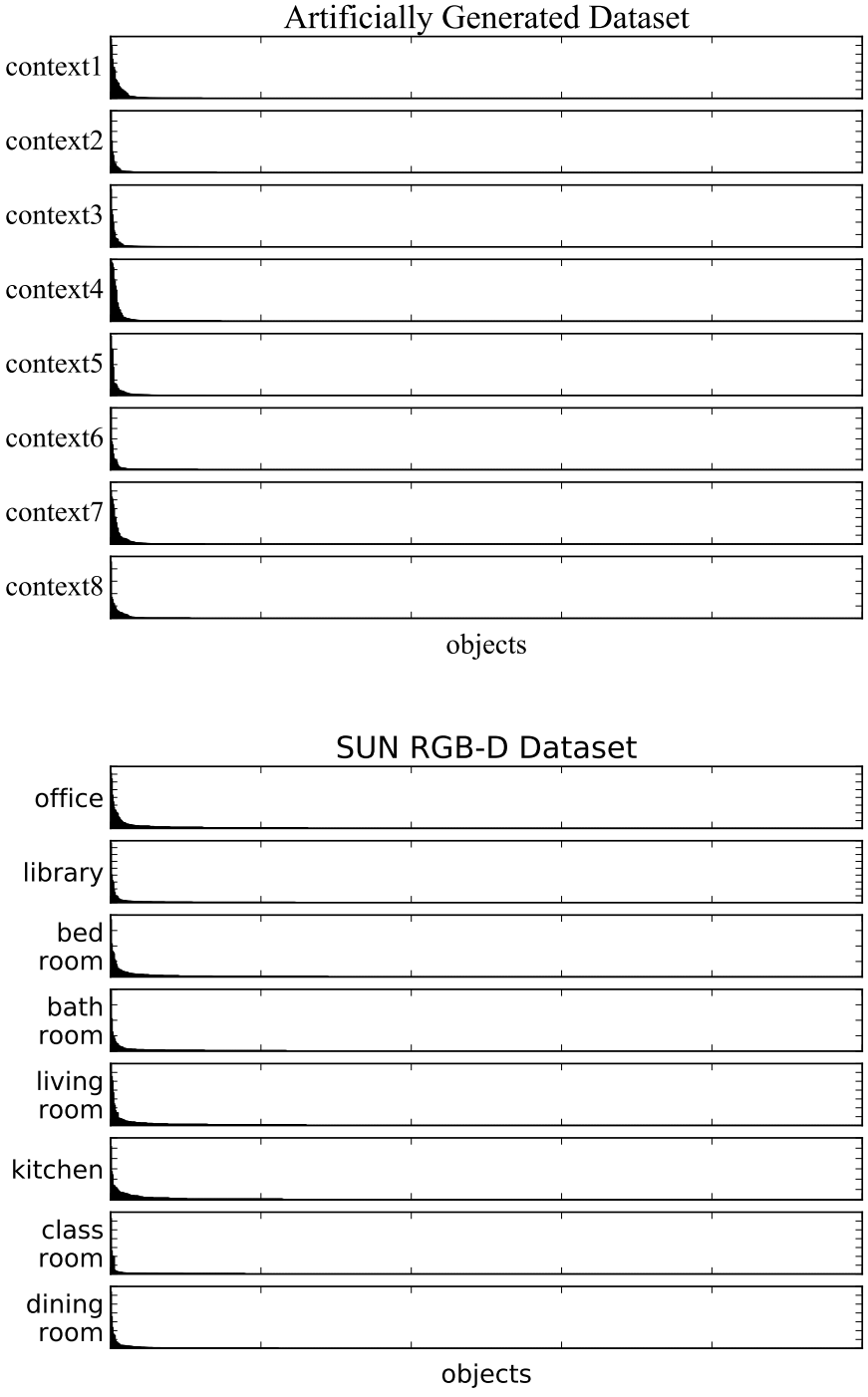


Figure 4.3: Context-object frequencies of artificially generated dataset and real dataset (SUN-RGBD [52]) [Figure source: [16]]

tribution over objects and c_i is sampled from Dirichlet distribution of β (i.e., $c_i \sim Dir(\beta)$).

- Number of objects N in a scene should be decided.
- θ should be sampled where $\theta \sim$ Dir(α) as the probability distribution of contexts for a given scene.
- To generate each object in the scene:

  a context c_i should be sampled where $c_i \sim Multinomial(\theta)$.

  an object o_i should be sampled by using sampled context c_i where $o_i \sim Multinomial(c_i)$.

Assume that corpus which contains all the scenes are generated by using k number of contexts and called D^k where $D^k = \{S_1^k, ..., S_{L_k}^k\}$. Different LDA models should be trained by using D^k for k_0 number of contexts where $k_0 \leqslant k$. This training will produce the inputs $\mathbf{x}$ and labels y for each input of recurrent models.

Tuples $(\mathbf{x}, y)$ used for training the deep recurrent network is generated as follows:

(a) $\mathbf{x}$: Input of deep recurrent network which shows the different LDA models. It is variable length since the number of context is changed from 1 to k for depending on the number of context in the generation process. $\mathbf{x}_i$ is composed of probabilities of each context for a given object (i.e., $\mathbf{p}_{c_i}$) which can defined as follows:

$$\mathbf{x}_i = \mathbf{p}_{c_i} = \{p(c_i|o_j)\}_{j=1}^{N}. \tag{4.2}$$

Actually, LDA model does not give $p(c_i|o_j)$ but it gives the probabilities of each context for a given scene (i.e., $\{p(c_i|s_j)\}_{j=1}^{M}$) and each object for a given context (i.e., $\{p(o_i|c_j)\}_{j=1}^{k}$). In order to obtain $p(c_i|o_j)$, we used Bayes formula:

$$P(c_i \mid o_j) = \frac{P(o_j \mid c_i)\, P(c_i)}{\sum_{i=1}^{k} P(o_j \mid c_i)\, P(c_i)}. \tag{4.3}$$

Moreover, $P(c_i)$ term can be obtained through marginalization as follows:

$$P(c_i) = \sum_{t=1}^{M} P(c_i|s_t)P(s_t). \tag{4.4}$$

In Equation 4.4, $P(c_i|s_t)$ term comes from LDA model and $P(s_t)$ is assumed to be $\frac{1}{M}$ since each scene is equally probable and follows a uniform distribution where M refers the number of scenes, N refers the number of objects and k refers the number of contexts.

(b) y: Represents the binary label for the input $\mathbf{x}$. It represents incrementing the number of contexts (i.e., $y = 1$) when $k_0 < k$ and stopping to increment number of contexts ($y = 0$) when $k_0 = k$.

By following the mentioned steps, we generate 14400 instances up to 10 contexts and each of them contains 1000 scenes and 100 objects for each scene. The total number of objects are selected as 1000.

After the generation part, we trained different LDA models with $y = 1$ and $y = 0$ labels by trying to enhance the number of instances with each label. Therefore, in total, we had 27,000 $(\mathbf{x}, y)$ pairs for training and 3,400 pairs for testing. Collecting more data may give more accurate results but we faced the time limitation since training different LDA models are time-consuming because of training each instance with $y = 1$ and $y = 0$ labels.

4.3 The Deep Recurrent Networks

Recurrent Neural Network (RNN) is an artificial neural network where hidden neuron connections are based on a directed cycle. In our model, RNNs are employed in order to handle variable length inputs which are caused by training LDA models with different numbers of contexts. The architecture comparison between Feed Forward Neural Networks and Recurrent Neural Networks can be seen in Figure 4.4.

The problem is considered as a learning problem for determining when to increment the number of contexts. $\mathbf{x}$ which shows the states of the LDA is given as an input to

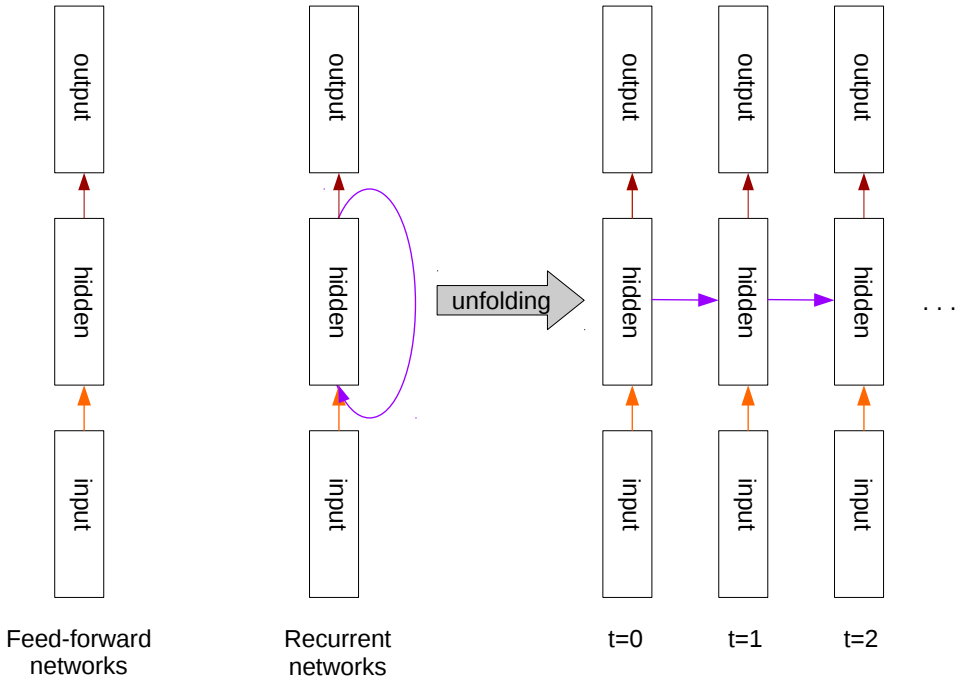


Figure 4.4: Comparison of Feed Forward Neural Networks and Recurrent Neural Networks

deep recurrent network and y which shows the necessity of adding a new context is predicted by the model. The deep recurrent architecture of our model can be seen in Figure 4.5.

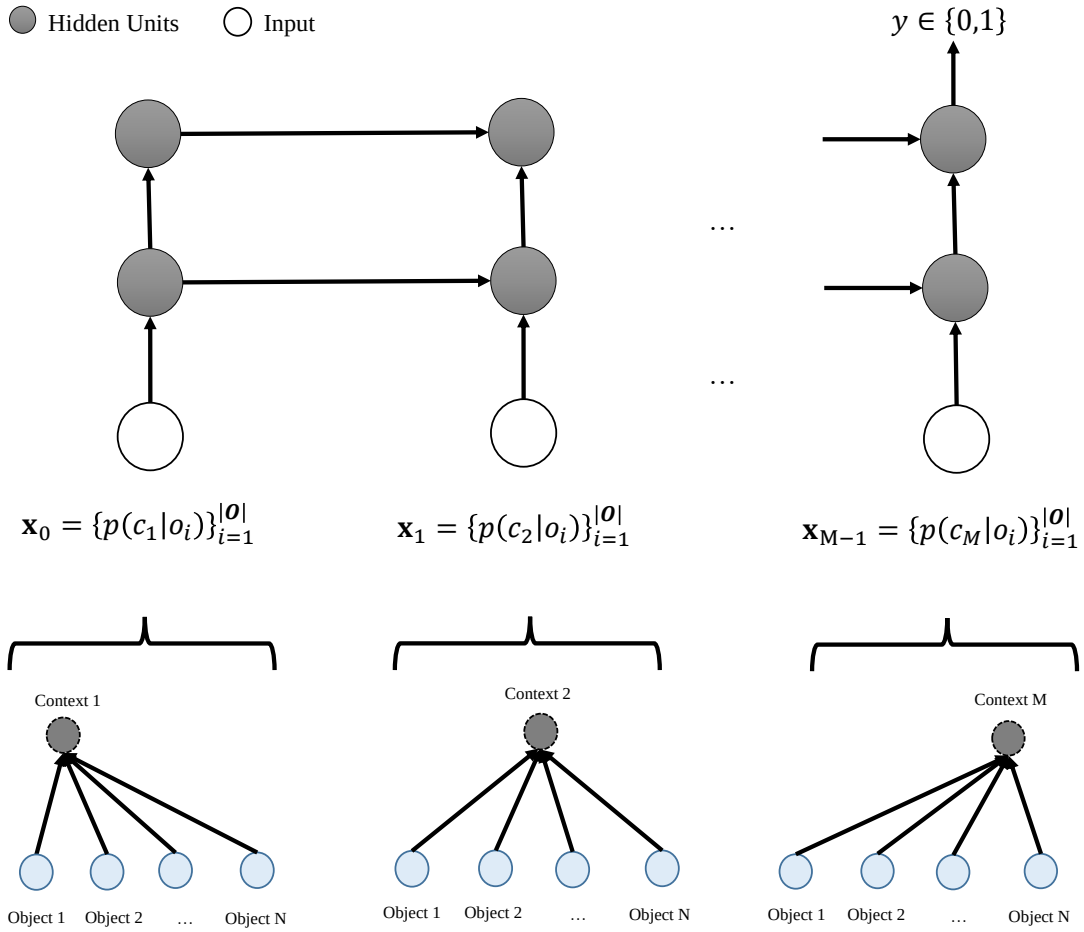


Figure 4.5: Unfolded view of RNN architecture used for predicting when to increment number of contexts [Figure source: [16]]

Different types of Recurrent Neural Networks with a different number of hidden units and number of layers have been tested in our model. For instance, Long Short-term Memories [23] are RNN architectures that are able to remember the necessary values over arbitrary intervals. Many to one LSTMs seem also suitable to fulfill long-term memory requirements of our problem. Moreover, GRU [13]) is also implemented in order to evaluate our system. Most successful ones are shown in Section 5.2.

Since neural networks try to minimize the loss function, we need to determine to the loss function that will be used in our system. For this purpose, we use a binary cross-entropy loss $\mathcal{J}$ which can be defined as follows:

$$\mathcal{J}(W) = -\frac{1}{n}\sum_{i}\left[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right]. \tag{4.5}$$

In Equation 4.5, W corresponds the parameters of the model, $\hat{y}_i$ shows the prediction of the model for the i^{th} sample in the dataset, n represents the number of samples or the batch in the dataset.

L2 regularization loss is also added on weights in order to avoid over-fitting.

4.4 Training the Recurrent Model

Adam optimizer is employed in order to train recurrent network which is very popular while training the deep models [29]. Default values of the parameters are used ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) and the batch size is chosen as $m = 100$. Moreover, the training of the network is ended when the test set accuracy become to decrease (i.e., early-stopping).

CHAPTER 5

EXPERIMENTS AND RESULTS

In this chapter, different tasks are solved by using two different methods. In Section 5.1, the results from iRBM and diBM models are shown by comparing them against other successful models in the literature. These results are also presented in [18]. In Section 5.2, experimental outcomes from the combination of LDA and RNN models are presented. These results are also presented in [16].

5.1 Incremental Restricted Boltzmann Machines (iRBM) and A Deep Incremental Boltzmann Machine (diBM)

In the following experiments, results from iRBM, stacked iRBM, diBM, vanilla RBM initialized with the same number of hidden neurons found by iRBM, stacked RBM initialized with the same number of hidden layers and hidden neurons found by stacked iRBM, DBM initialized with the same number of hidden layers and hidden neurons found by diBM, incremental RBM model proposed by Yu et al. [67] and incremental LDA proposed by Celikkanat et al. [12] are presented. In order to compare these models, each of them is trained with the same number of epochs.

Since RBM and DBM are not incremental methods, these models are evaluated by training on one instance at a time (i.e., online mode) and whole instances at once (i.e., batch mode). Moreover, we also used diBM weights to initialize the vanilla DBM in order to examine how well starting point diBM weights can provide for vanilla DBM. This initialization is represented as DBM $\leftarrow$ diBM in the following experiments.

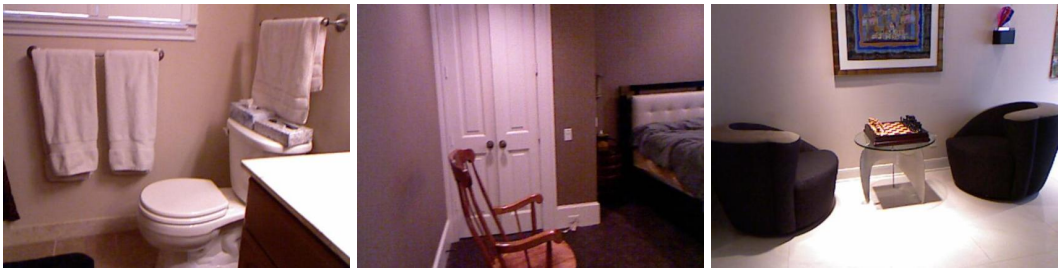


Figure 5.1: A few samples from the SUN-RGBD scene classification and segmentation dataset [52]

5.1.1 Dataset

Two datasets are used in order to evaluate our models.

The first dataset is called AP news document dataset which is obtained from Associated Press articles [1]. It is introduced by David Blei and originally generated for evaluating Latent Dirichlet Allocation (LDA). This dataset contains 2246 documents and 10473 different words in its vocabulary. 246 documents are chosen randomly for testing and the rest of them is used for training. We took the first 2000 documents for training and last 246 documents as test data. In this dataset, each document is considered as a scene where words and topics correspond objects and contexts respectively in order to use these documents in context modeling.

The other dataset is SUN RGB-D scene classification and segmentation dataset [52]. This dataset contains 10,335 labeled scenes, 11600 different objects and it is composed of NYU depth v2 dataset [49], the Berkeley B3DO dataset [26], and the SUN3D dataset [65]. 7,000 scenes are used for training and the rest of 3,335 scenes are employed for testing the models. This dataset is selected since it suits the nature of robotic problems by containing different scenes from different contexts with various objects. Since the robot needs to learn these contexts by having no prior knowledge about the contexts and number of them, this dataset seems proper to extract contextual information incrementally. Moreover, this dataset includes object labels with their annotations, positions, and depths. Labeled objects are used for facilitating the problem by excluding object recognition task out of the scope of our study since it

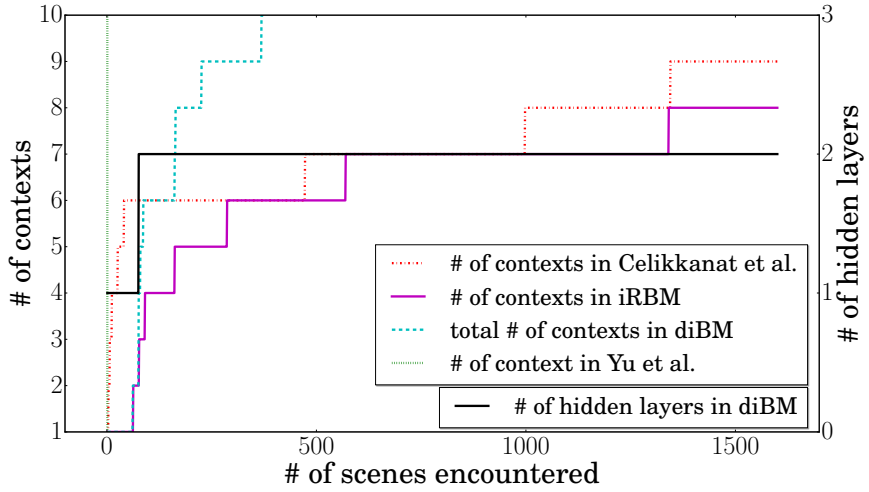


Figure 5.2: Number of hidden layers and topics on a subset of SUN RGB-D Dataset obtained from 8 contexts and 200 scenes from each context with online learning. The number of hidden layers is shown only for diBM model which results with 16 contexts in total thanks to representing super-contexts and sub-contexts in the hierarchical layers. [Best viewed in color]

can be achieved with a great success by using state of the art deep models. A few samples from SUN RGB-D dataset can be seen in Figure 5.1.

In order to use both datasets, bag-of-words and bag-of-objects approaches are followed. These words/objects are fed to the model from visible units and hidden representations of these datasets are obtained in an incremental and hierarchical way.

5.1.2 Number of Contexts

The first task is examining the number of contexts and hidden layers found by different incremental models as well as our models. Since SUN RGB-D does not contain an equal number of scenes for each context, a sub-dataset is obtained from SUN RGB-D which contains 8 contexts (scene categories) and 200 scenes for each context (i.e., 1600 scenes in total) and 3352 different objects. These contexts are labeled as an office, library, bedroom, bathroom, living room, kitchen, classroom and dining room.

Figure 5.2 shows the number of hidden neurons for incremental models and hidden

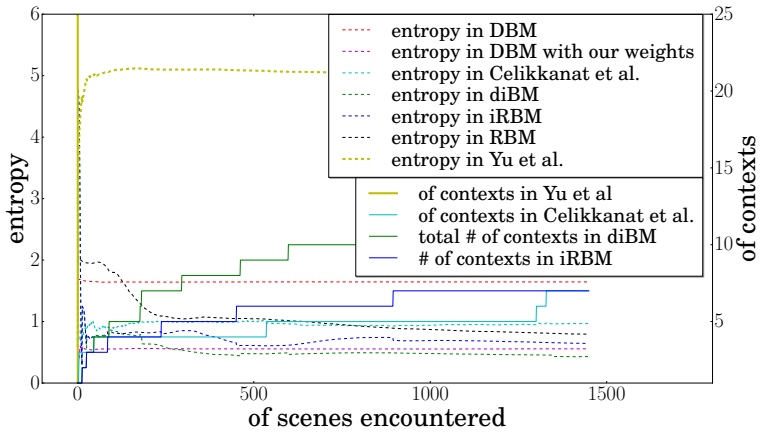


Figure 5.3: Entropy change over time obtained from different models on NYU Depth Dataset. DBM and RBM are excluded from the number of contexts since they have fixed number of hidden units. [Best viewed in color]

layers for diBM with online learning. As can be seen in the figure, iRBM manages to find the correct number of contexts. The number of hidden neurons found by diBM is 16 since the figure shows the total number of hidden neurons in all the hidden layers. Therefore, this means diBM also finds super-contexts and sub-contexts thanks to the hierarchy.

5.1.3 Entropy of the Models

Since Boltzmann Machines try to minimize the entropy of the models, we examined the entropy changes of different models while the systems evolve. The entropy of a contextual model is defined by Çelikkanat et al. [12] as follows:

$$\hat{H} = \rho H(o|c) + (1-\rho)H(c|s), \tag{5.1}$$

where o represents the objects, c corresponds the contexts and s shows the scenes. Moreover, $H(\cdot|\cdot)$ measures the conditional entropy as follows:

$$H(c|s) = -\sum_{i}\sum_{j} p(c_i|s_j) \times \log_2 p(c_i|s_j). \tag{5.2}$$

In Equation 5.1, $H(o|c)$ measures the entropy for observing a specific object in a given context and $H(c|s)$ determines the confidence of a context for a given scene.

Balancing these two terms gives the most specific contextualized scene models as stated in Çelikkanat et al. [12]. Therefore, ρ is a constant used for balancing the significance of these two terms and used as 0.5 in our experiments.

Entropy changes of different models can be seen in Figure 5.3. As can be seen, diBM finds a model which yields the lowest entropy. Entropies of DBM and diBM are calculated by taking the mean of entropies come from all layers in each sample.

5.1.4 Qualitative Inspection of Context Coherence (Hidden Nodes)

In order to observe the model strength for representing contexts, highest weighted objects for each hidden neuron are inspected. Since objects are directly related to first hidden layer neurons, one-layer methods are inspected in this section which correspond the iRBM, incremental RBM [67], incremental LDA [12] and online vanilla RBM. diBM, DBM, stacked RBM and stacked iRBM are discarded since they end up with similar results with their single-layer counterparts, i.e., RBM and iRBM.

Table 5.1 shows the highest weighted 10 objects (i.e., visible units having the highest 10 weights to a hidden neuron) of different models on the subset of SUN RGB-D data for the best three contexts which are selected by visual inspection. This sub-dataset contains 8 contexts and 200 scenes from each context (i.e., 1600 scenes in total). The irrelevant objects which do not suit the contexts are written with a red color. In terms of our observations, iRBM ends up with the best results by finding the most relevant objects together in separate contexts. iRBM has found office/library context in the first hidden neuron, kitchen context in the second hidden neuron and bathroom context in the third hidden neuron which all exist in the dataset. Results from Celikkanat et al. [12] keep irrelevant objects together in the third hidden neuron. Moreover, incremental RBM [67] and online vanilla RBM seem to end up worst results in terms of our visual inspection.

Table 5.1: Most probable 10 objects of different models on a subset of SUN RGB-D dataset for the best 3 hidden units. “d]” is indeed a label in the dataset. Red colored objects correspond the irrelevant ones to the context.

	Hidden1	Hidden2	Hidden3
iRBM (9 contexts found)	keyboard	oven	sink
	mouse	stove	toilet
	computer monitor	carpet	faucet
	cord	countertop	pipe
	chair	toaster	soap
	cpu	microwave	tap
	monitor	tilefloor	cabinets
	pillar	refrigerator	urinal
	desktop	painting	towel
	scanner	plates	Toilet Paper Dispenser
Çelikkanat et al. [12] (9 contexts found)	chair	wall	wall
	table	keyboard	floor
	floor	monitor	sink
	wall	computer	toilet
	desk	desk	cabinet
	door	paper	counter
	window	mouse	pipe
	board	floor	door
	bookshelf	door	towel
	chairs	window	microwave
Yu et al. [67] (3527 contexts found)	urinal	keyboard	floor
	toilet	book	chair
	towel	monitor	cup
	pipe	pillow	minifridge
	book	flowers	lid
	sink	mirror	refrigeration
	window	window	cabinet
	bookshelf	adapter	insulatedbag
	garbage	wall	stallsreflection
	bottiles	floor	frame
RBM [online] (8 contexts given)	mirror	counter	table
	sink	teapot	chairs
	floor	toaster	bookcase
	window	coffeemaker	sofa
	plumbing	wall	chairline
	mop	carrier	Electrical Device
	towel	stove	triangle
	wall	light	classplate
	counter	d]	dress
	toilet	cupboard	glass

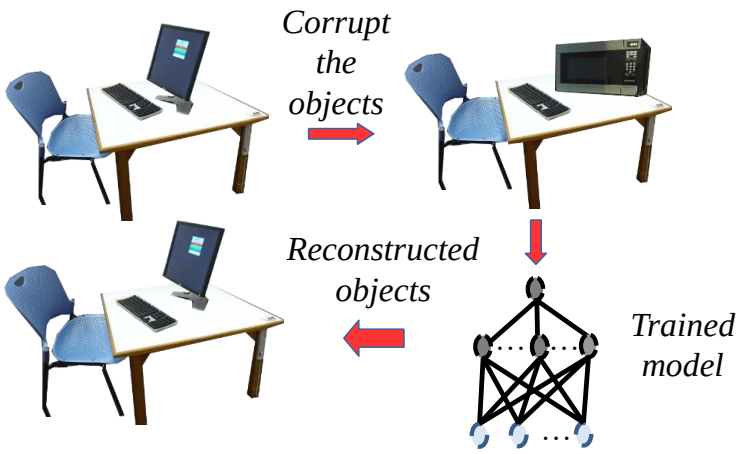


Figure 5.4: An illustration of scene reconstruction [Figure source: [18]]

5.1.5 Partially Damaged Scene Reconstruction

In order to evaluate our models in terms of data distributions, samples from test set (i.e., $\mathbf{v} \in \mathbf{V}$) are partially-corrupted. This partially corrupted input is represented with $\tilde{\mathbf{v}}$ and tried to be reconstructed (shown as $\mathbf{v}'$). For this task, partially corrupted samples $\tilde{\mathbf{v}}$ are fed into the model as visible neurons and thanks to going forward and backward from visible neurons to hidden neurons, $\tilde{\mathbf{v}}$ is reconstructed in visible neurons. An illustration of this scene reconstruction task can be observed in Figure 5.4.

In order to corrupt the samples from the test set, α dimensions are randomly chosen in $\mathbf{v}$ and selected dimensions are flipped with probability 0.5.

Some metrics are proposed in order to examine the described task. These metrics which measure the performance of the models are defined as follows:

$$\text{CD} = 1 - \frac{\sum_{\mathbf{v}\in\mathbf{V}} \sum_i a(v_i - v_i')}{\alpha|\mathbf{v}| \times |\mathbf{V}|}, \tag{5.3}$$

$$\text{CDa} = 1 - \frac{\sum_{\mathbf{v}\in\mathbf{V}} \sum_i a(v_i - v_i')}{\sum_{\mathbf{v}\in\mathbf{V}} \sum_i a(v_i - \tilde{v}_i)}. \tag{5.4}$$

In Equation 5.4, CD shows the corrupted dimensions and CDa represents the corrupted data. Moreover, $a(\cdot)$ is the absolute value function calculates the difference between the real data and the reconstructed data.

These metrics consider all the corrupted and uncorrupted parts of the input in the absolute value function, therefore CD and CDa may be negative if the number of destroyed bits are more than successfully reconstructed ones. Since some models destroy the uncorrupted objects while trying to reconstruct the corrupted ones, two other metrics are proposed as an optional measure to obtain always positive performances:

$$\text{CD}^k = 1 - \frac{\sum_{\mathbf{v}\in\mathbf{V}} \sum_i a(u_i - u'_i)}{\alpha|\mathbf{v}| \times |\mathbf{V}|}, \tag{5.5}$$

$$\text{CDa}^k = 1 - \frac{\sum_{\mathbf{v}\in\mathbf{V}} \sum_i a(u_i - u'_i)}{\sum_{\mathbf{v}\in\mathbf{V}} \sum_i a(v_i - \tilde{v}_i)}, \tag{5.6}$$

where absolute value function is calculated in terms of corrupted bits of the input rather than considering all the corrupted and uncorrupted ones. Therefore, u and u' represent the corrupted part of v and v'.

Tables 5.2 and 5.3 show the results from scene and document datasets by using different models.

Inside the batch methods which are based on training the model by using all data at once, stacked RBM and DBM initialized with diBM weights gives the best results for the scene and document datasets respectively. It is natural to obtain the best results from DBM initialized with diBM in the document dataset since it has a deeper representation power and pre-processed weights. The surprising part is obtaining a better result in Stacked RBM than DBM or DBM initialized with diBM weights in scene dataset. This can be caused by the convergence time of DBM in the scene dataset. Since all the models are trained with the same number of epochs, this shows us stacked RBM converges faster than other models on SUN RGB-D dataset. Faster convergence of Stacked RBM is also supported by document dataset since Stacked RBM gives the second-best accuracies and performs better than DBM.

By examining the online methods which are based on training the model with a single train instance at a time, diBM and DBM initialized with diBM weights seem to give best results for scene and document dataset respectively. Moreover, diBM also performs the best not only inside the incremental methods but also inside the batch models on SUN RGB-D dataset. Moreover, obtaining the best result from DBM initialized with diBM weights in document dataset suggests that diBM can be used for

Table 5.2: Reconstruction of performances for the testing part of SUN RGB-D dataset [52]. The corruption rate (α) is 40%. KCP and UCP represent the known and unknown corrupted parts which show determining error value in terms of corrupted parts or whole data respectively.

		KCP		UCP	
		CD^k	CDa^k	CD	CDa
Batch	RBM	0.719	0.438	0.290	-0.419
	Stacked RBM	0.998	0.997	0.996	0.992
	DBM	0.854	0.707	0.631	0.262
	DBM ← diBM	0.969	0.938	0.925	0.849
Online	RBM	0.752	0.504	0.373	-0.253
	Stacked RBM	0.998	0.996	0.996	0.993
	iRBM	0.962	0.925	0.906	0.812
	Stacked iRBM	0.997	0.995	0.994	0.987
	diBM	**0.999**	**0.998**	**0.997**	**0.994**
	DBM	0.997	0.994	0.993	0.989
	DBM ← diBM	0.997	0.993	0.991	0.983
	Yu et al. [67]	0.521	0.042	-0.099	-1.200

Table 5.3: Reconstruction of performances on the testing part of Associated Press dataset [1]. The corruption rate (α) is 30%. KCP and UCP represent the known and unknown corrupted parts which show determining error value in terms of corrupted parts or whole data respectively.

		KCP		UCP	
		CD	CDa	CD^k	CDa^k
Batch	RBM	0.657	0.313	-0.139	-1.278
	Stacked RBM	0.994	0.988	0.917	0.834
	DBM	0.890	0.781	0.632	0.262
	DBM ← diBM	0.996	0.992	0.956	0.911
Online	RBM	0.597	0.194	-0.180	-1.359
	Stacked RBM	0.976	0.952	0.796	0.593
	iRBM	0.888	0.777	0.412	-0.176
	Stacked iRBM	0.994	0.989	0.920	0.839
	diBM	0.995	0.990	0.920	0.840
	DBM	0.993	0.987	0.940	0.879
	DBM ← diBM	**0.997**	**0.995**	**0.961**	**0.921**
	Yu et al. [67]	0.835	0.669	0.323	-0.355

initializing the DBM as a pre-processing step. This initialization may facilitate the training of DBM by reducing the convergence time with a better accuracy and getting rid of the necessity of determining the number of hidden neurons and hidden layers at the beginning. Knowing the number of hidden nodes and layers beforehand is very difficult and may yield worse results if it is not done properly. Finally, iRBM performs noticeably better than RBM no matter training on document or scene datasets in a bath or incremental way.

It can be inferred from Table 5.2 and 5.3 that our incremental models can converge a model that has a fixed structure. Comparing the results of iRBM with RBM, stacked iRBM with stacked RBM and diBM with DBM shows that evolving incremental and hierarchical models may be constructed. Additionally, these models can achieve as good results as their rigid counterparts even they can achieve better performances in some datasets. Moreover, evolving models solve the problem of determining the number of hidden neurons and layers (i.e., model selection) before training. Finally, results suggest that our models can be used for initializing their rigid counterparts.

5.2 A Learning Based Approach to Incremental Context Modeling

In this section, the results from the recurrent model where the different Latent Dirichlet Allocation models are used as an input are examined. Firstly, accuracies of training and testing of different deep models based on correct increment decisions on artificial dataset are shown. Then, generalization capability of the recurrent model is tested by investigating the probabilities of incrementing the number of context on artificial dataset when the ground truth is more than the number of contexts while the model is trained to. After examining the probabilities of incrementing the number of contexts, the entropies of incremental models are analyzed on the artificial dataset. Finally, outcomes of a real dataset are indicated by using scenes belong to different contexts in terms of investigating the probabilities of incrementing the number context and entropies as in the case of the artificial dataset.

5.2.1 Deep Network Training and Testing Performance

Training and testing performances of the proposed method can be observed in Table 5.4.

As can be seen in Table 5.4, different models in terms of memory units and layers are tested on the artificial dataset. Moreover, different inputs are evaluated in these models. As defined in Section 4.2, these inputs are $\mathbf{p}_{c_i}$ which shows the probability of contexts given objects, $\mathbf{p}_{o_i}$ which represents the probability of objects given contexts and $\mathbf{p}_{c_i} \oplus \mathbf{p}_{o_i}$ which corresponds the concatenation of these probabilities. The calculation procedure of these probabilities and what these probabilities state are expressed in Section 4.2 in detail.

In Table 5.4, accuracies are computed in terms of correct incrementation decisions by using artificial data. Moreover, 50 hidden unit is used for each hidden layer in these experiments and this number is determined empirically.

Among all the models, the model which contains LSTM memory units, 3 hidden layers and uses $\mathbf{p}_{c_i}$ as an input achieved the best accuracies. Since using $\mathbf{p}_{c_i}$ as an input ends up with the best accuracies for all the models, the probability distribution of contexts for a given object provides the fundamental information about the necessity of incrementing the number of contexts.

There is a very tiny difference between training and testing performances in Table 5.4, especially for the inputs $\mathbf{p}_{c_i}$ and $\mathbf{p}_{c_i} \oplus \mathbf{p}_{o_i}$. This shows the networks do not tend to over-fit the training data for these inputs. The larger difference between training and testing while using $\mathbf{p}_{o_i}$ as an input states that there is a more complicated and difficult to learn visible unit representations and input spaces in $\mathbf{p}_{o_i}$.

5.2.2 Applying Trained RNN to Incremental Context Modeling

In this term, results from the recurrent network trained with LSTM memory units and 3 hidden layers are examined on artificially generated dataset as explained in

Table 5.4: Accuracies of training and testing performances of different models with different memory units and hidden layers. Performances are calculated based on correct increment determinations on artificial data.

	Input: $\mathbf{p}_{c_i}$		**Input:** $\mathbf{p}_{o_i}$		**Input:** $\mathbf{p}_{o_i} \oplus \mathbf{p}_{c_i}$	
	Training Acc.	**Test Acc.**	**Training Acc.**	**Test Acc.**	**Training Acc.**	**Test Acc.**
Vanilla RNN (1 layers)	98.0%	97.1%	72.6%	66.2%	99.3%	95.5%
Vanilla RNN (2 layers)	99.2%	97.7%	97.0%	69.5%	97.5%	94.8%
Vanilla RNN (3 layers)	99.7%	97.9%	99.5%	71.4%	94.8%	93.9%
GRU (1 layers)	99.4%	94.7%	97.2%	71.0%	99.5%	93.7%
GRU (2 layers)	99.3%	97.3%	99.9%	71.7%	99.4%	96.0%
GRU (3 layers)	99.4%	97.3%	99.8%	71.7%	94.3%	94.2%
LSTM (1 layers)	99.1%	92.9%	73.4%	67.5%	99.7%	89.6%
LSTM (2 layers)	99.4%	97.9%	99.7%	70.6%	99.7%	96.7%
LSTM (3 layers)	**99.9%**	**98.0%**	99.4%	70.9%	99.4%	94.2%

Section 4.2) and real dataset known as SUN RGB-D [52] whose details are explained in Section 5.1.1.

5.2.2.1 Experiments on the Artificially Generated Dataset

As explained in Section 4.2, datasets may contain contextual information but it is difficult to find a dataset which includes the exact number of contexts since there may be more contexts because of discarding the existence of some sub-contexts. Therefore, in this section probabilities of incrementing the number of context and entropy of the model are discussed on artificially generated data.

5.2.2.1.1 Probabilities of Incrementing Number of Contexts

The probabilities of incrementing the number of contexts are shown in Figure 5.5 where the datasets are generated with k contexts. k is chosen as $5, 7, 10, 15, 20$ randomly. After the data generation, the different LDA models trained with k_0 context where $k_0 \leqslant k$ are used to feed to the recurrent network. As it can be observed, the LSTM predicts to increment the number of context (i.e., k_0) with very high probability when $k_0 < k$. This probability is 0.98 when $k = 5, 7$ and 0.84 when $k = 15, 20$ on average. In addition, when $k_0 = k$, the probability of incrementing k_0 decreases expectedly. Therefore, the network increments the number of contexts when it is less than the ground truth and stops incrementing the number of contexts when it reaches the ground truth. This shows our network handles the problem of determining when to increment the number of contexts properly.

Moreover, the probabilities in part d and e in Figure 5.5 show the result on the input data generated with 15 and 20 contexts. For these contexts, it also follows a similar pattern where the probability of incrementing the number of contexts is decreasing while approaching the ground truth. Since our LSTM model is trained for artificial data generated with up to 10 contexts, it can be concluded that our deep model has good predictions for more contexts than it has been trained for with a good generalization capability. The only deficient is that model tries to stop incrementing the

number of context for a bit less number than the ground truth.

The network's well generalization capability can be caused by the resemblance between distributions of data generated with $k < 10$ and $k > 10$ contexts. Therefore, the network is not depending on the number of contexts when the probabilities between contexts and objects are provided through the weight-sharing mechanism of the recurrent network over different time steps.

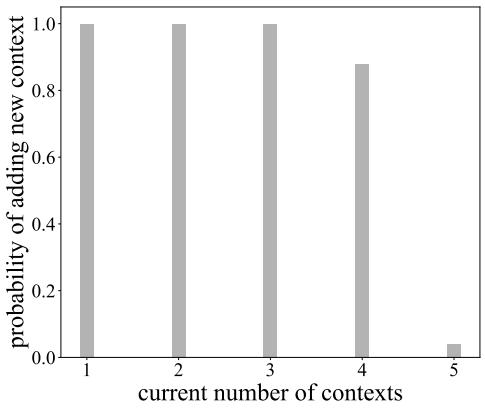
5.2.2.1.2 Entropy of the Model

We examined how the system evolves while encountering new scenes thus incrementing the number of contexts. In order to analyze this change, a sub-dataset is generated with 5 contexts. The results may be observed in Figure 5.6 where entropy change of our model compared with the entropy change in the model of Çelikkanat et al. [12]. This comparison shows that our model achieves to end with a correct number of context with almost the same low-level entropy in the study of Çelikkanat et al. [12]. However, they yield a wrong number of contexts which is more than the ground truth.

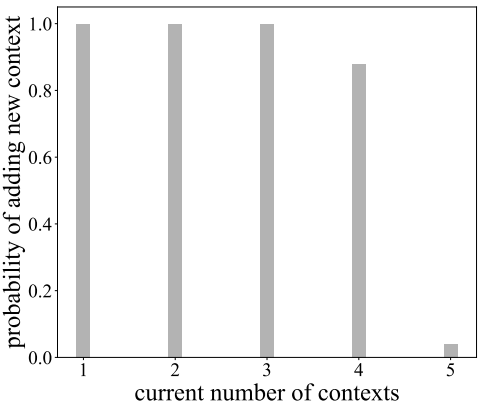
In order to evaluate the entropy change, the same measure in Section 5.1.3 is used which is adapted from Çelikkanat et al. [12]. Equation 5.1 is used for the entropy where the details of these equation can be seen in Section 5.1.3. The only difference is that we used 0.9 for the term ρ which balances the confidences of encountering certain objects in a given context and experiencing certain contexts in a given scene.

5.2.2.2 Experiments on a Real Dataset

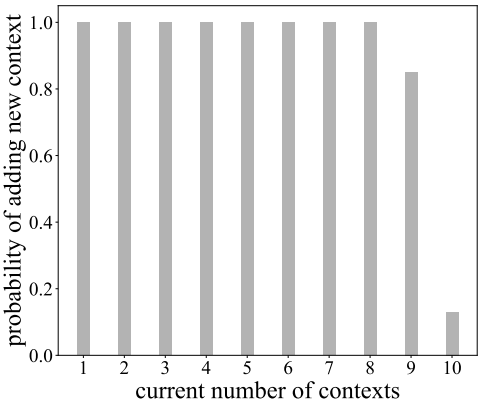
The subset of SUN RGB-D dataset [52] is used to examine the results on a real dataset. The details of this dataset are explained in Section 5.1.1. In order to evaluate our model in a real dataset, a subset of SUN RGB-D is extracted by taking 878 scenes with 1000 different objects. This sub-dataset contains 8 main contexts and 25 sub-contexts. In this explanation, a context may correspond to an office context and a sub-context may represent a home-office context. The number of sub-contexts



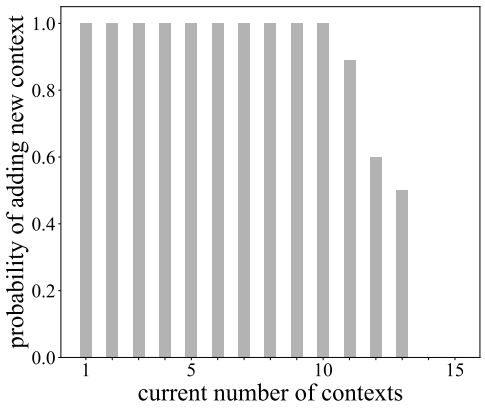
(a) Ground truth=5



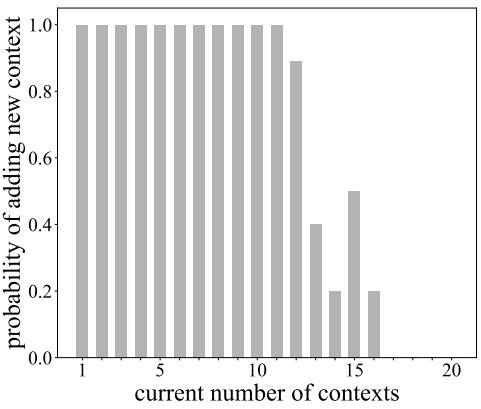
(b) Ground truth=7



(c) Ground truth=10



(d) Ground truth=15



(e) Ground truth=20

Figure 5.5: Probabilities of incrementing the number of contexts on the artificial data generated by different LDA models. Ground truth is 5, 7, 10, 15, 20 from a to e respectively. The recurrent model is trained on the inputs come from LDA models trained up to 10 contexts.

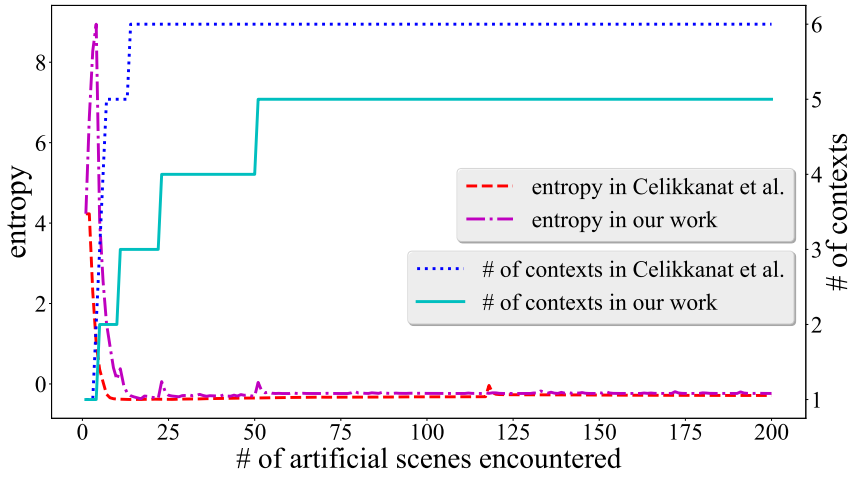


Figure 5.6: The change in entropy with respect to the number of contexts and encountered scenes by using **artificial** dataset. The entropy change is evaluated with the subset of dataset which contains 5 context chosen randomly. The number of context yielded by [12] is more than the ground truth. [Best viewed in color]

gives the baseline for learning the number of context problem in this sub-dataset. This dataset which includes different contexts, sub-contexts and objects is proper for a robotic scenario where a robot experiences one scene at a time and learns when to increment the number of contests incrementally. Labeled objects are used in order to exclude the object recognition task out of the scope of this study as explained in Section 5.1.1. Some samples from SUN RGB-D dataset can be observed in Figure 5.1.

In the following sections, probabilities of incrementing the number of contexts and entropy change of the model on the subset of SUN RGB-D is presented.

5.2.2.2.1 Probabilities of Incrementing Number of Contexts

The probabilities of incrementing the number of contexts for each k_0 is shown in Figure 5.7. As stated, there are 25 sub-contexts in the dataset and the incrementation probabilities are very high when k_0 is less than the baseline. When k_0 become closer to 25, the probability of adding a new context decreases drastically as expected and

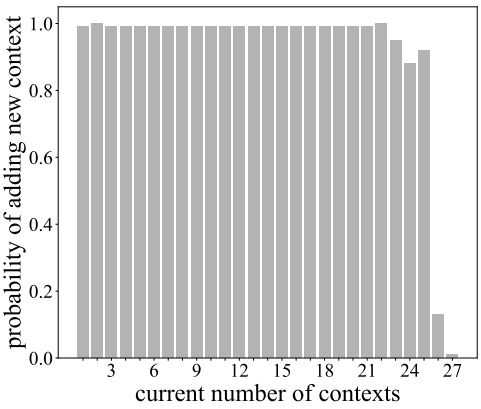


Figure 5.7: Probability of adding a new context for different LDA models on the real data (i.e., subset of SUN RGB-D data [52])

the model tries to stop adding a new context.

5.2.2.2.2 Entropy of the Model

The entropy changes through adding new contexts by observing a new scene at a time are examined for different models on a subset of SUN RGB-D data which contains 8 contexts and 25 sub-contexts as a baseline. These results can be examined in Figure 5.8. Combination of LDA and RNN model seems to yield a number of contexts close to the number of sub-contexts and other models obtain a number of contexts close to the main context categories. These results suggest that combination of LDA and RNN model tends to represent the sub-contexts better compared to the other methods. Moreover, it is very difficult to obtain exactly 25 contexts because of the noises while labeling the sub-contexts and discarding some of them in the real dataset. However, the combination of LDA and RNN model seems to converge close enough number of contexts in order to represent all the scenes in terms of their sub-context categories.

Combination of LDA and RNN model also converges a very low entropy while encountering new scenes and incrementing the number of contexts. This entropy value is very close to the entropy in [12] and lower than iRBM and diBM models as can be seen Figure 5.8. These entropy values are calculated by using Equation 5.1 explained

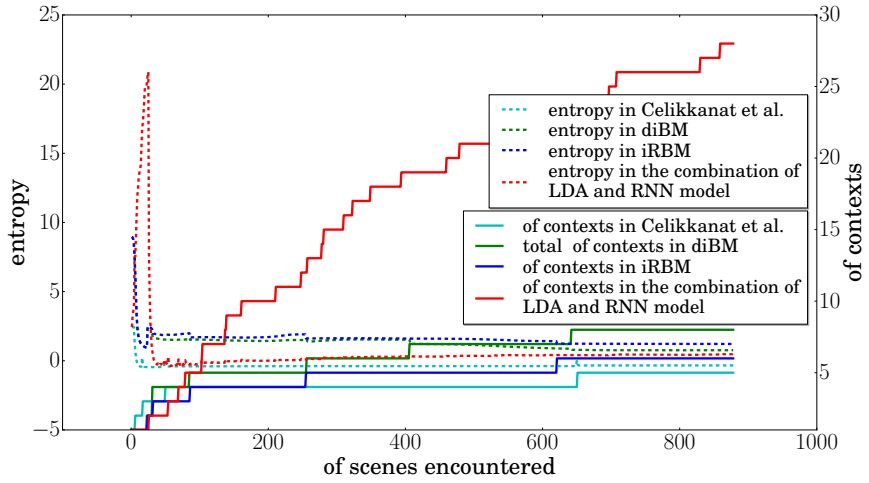


Figure 5.8: Entropy of the models while facing different scenes and incrementing number of contexts on **real** (i.e., SUN RGB-D [52]) data which includes 8 context and 25 sub-context as a baseline. Combination of LDA and RNN model finds the number of context closer to the sub-contexts categories compared to the other models which diverge drastically to a closer number to main contexts categories [Best viewed in color]

in Section 5.1.3. Moreover, balancing constant for two terms (i.e., ρ) is selected as 0.9 (i.e., the same value in Section 5.2.2.1.2).

5.2.2.3 Inferences from Artificial and Real Datasets

Through the observations come from Section 5.2.2.2 and Section 5.2.2.1, even though our recurrent model was trained with artificial data generated by LDA models, it has a very high generalization capability for more contexts than it has been trained for (more than 10 contexts) and for real data (i.e., SUN RGB-D data [52]). This generalization capability comes from similar object to context alignment between artificial data generated up to 10 contexts, more than 10 contexts and real data. Thanks to our model's ability to capture the distributions between objects and contexts, our deep model learns properly determining when to increment the number of contexts for the problems which follow a similar distribution.

CHAPTER 6

CONCLUSION AND DISCUSSION

In this thesis, we proposed two different methods to model contextual information on robots by taking into account the properties of the context.

Since the properties of context suggest an incremental and hierarchical structure, in the first model, Restricted Boltzmann Machines are extended to represent contextual knowledge incrementally and hierarchically. For this purpose, two different algorithms are formed. These algorithms can be summarized as follows:

(a) First algorithm is for constructing an incremental approach. The idea behind generating an incremental model is based on the assumption that each object should be represented at least one context. This means when an object is not represented by the current contexts (e.g., its confidence value is low than a threshold and this threshold value is also determined by the model dynamically), a new context is added in order to represent that object by the newly added context. This condition also suggests that one context should be activated by at least one object. Moreover, when the model is a deep, this assumption also implies that a context should be represented by at least one other context at a higher level in the hierarchy.

(b) Second algorithm is about when to add an upper layer to the hierarchy while representing scenes. This is based on representing close contexts in an upper layer of the hierarchy. In other words, when the distance between any two contexts in the final layer is closer than another threshold value (also determined by the model dynamically), a context layer is added at the top of the hierar-

chy in order to represent these similar contexts. Moreover, incrementing the number of contexts in each context layer proceeds by following the previously mentioned incrementation algorithm.

When we examine the results of this model, we see that our deep incremental model has a good capability to represent contextual information and the object-context data distributions. Our model could find the correct number of contexts for a subset of RGB-D data where all the other incremental methods failed in this task. Moreover, since we claim that hidden neurons correspond to the contexts, we investigated the most probable visible neurons to the hidden neurons in order to verify our assumption. We see that our model gives the importance to the different objects for different hidden neurons and these objects seem to form a context by containing logical connections. Moreover, the entropy of our model yields a low value (i.e., lowest in diBM) which shows it can reach a stable condition at the end. Finally, we viewed the reconstruction performances of our model and we see that our model has a good ability (i.e., the best in SUN RGB-D dataset and second at the Associated Press articles) to understand the data distributions between contexts and objects. These results also state that our model can be used as a pre-processing step for DBM's by solving the problem of determining the number of hidden neurons and hidden layers in DBMs. Since we used all the same algorithm and the same parameters for Associated Press articles and SUN RGB-D scenes and obtained very high reconstruction performances, iRBM and diBM have good generalization abilities.

In the second model, learnability of a number of contexts is studied by following an incremental approach. To the best of our knowledge, this model is the first model that focuses on learning the number of context without using any rule-based method. This method is also composed of two steps:

(a) First step is based on generating the dataset for the learning problem. Since the dataset with the correct number of context is a challenge, we employed LDA model in order to generate data with different numbers of contexts. After the generation part, each data is trained with different numbers of contexts and

when this number is equal to the number used in the generation, it is labeled as 0 (i.e., not increment), otherwise labeled as 1 (i.e., increment).

(b) Different LDA models which are trained with different numbers of contexts is given to the recurrent network in order to learn when to increment the number of contexts. This is a "sequence to label" problem with binary labels.

The second model is evaluated by using artificial data and real data. In both datasets, performances on finding the correct number of contexts are evaluated and our model end with high results (e.g., 98% accuracy in the artificial dataset and very close to the grand-truth sub-contexts in the real dataset) in both datasets. Moreover, our model is trained for up to 10 contexts but also yields good results for the datasets which contain more than 10 contexts. This also shows that our model has a generalization capacity. Finally, our model is tasted for entropy change over time and it yields a very low entropy which shows reaching a stable condition at the end.

6.1 Limitations and Future Work

Our first model has a limitation on allowing only to grow the contextual model, but it should be improved by allowing to shrink the model. It enables adding new contexts and context layers for now, but a context should be deleted if it represents none of the objects and the contexts should be merged if they represent similar objects (i.e., they correspond to similar contexts).

Moreover, the scene dataset (SUN RGB-D dataset) which both of our models are evaluated only contains spatial contexts but our models should be tested on a dataset which has temporal or social contexts. Obtaining a dataset with these connections will be a challenge for this extension.

Finally, our second method does not consider the hierarchical relations between contexts and it should be also enhanced in order to build the hierarchical relations by considering it as another learning problem.

REFERENCES

[1] Associated press news document dataset. `http://www.cs.columbia.edu/~blei/lda-c/ap.tgz`. Accessed: 2017-12-2.

[2] Definition of context. `http://www.dictionary.com/`. Accessed: 2017-11-30.

[3] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[4] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, 32(1):19–34, 2013.

[5] L. W. Barsalou. Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1281–1289, 2009.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] S. Blumenthal and H. Bruyninckx. Towards a domain specific language for a scene graph based robotic world model. *arXiv preprint arXiv:1408.0200*, 2014.

[8] S. BuvaE and I. A. Mason. Propositional logic of context. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. sn, 1993.

[9] K. Canini, L. Shi, and T. Griffiths. Online inference of topics with latent dirichlet allocation. In *Artificial Intelligence and Statistics*, pages 65–72, 2009.

[10] H. Çelikkanat, G. Orhan, and S. Kalkan. A probabilistic concept web on a humanoid robot. *IEEE Transactions on Autonomous Mental Development*, 7(2):92–106, 2015.

[11] H. Celikkanat, G. Orhan, N. Pugeault, F. Guerin, E. Sahin, and S. Kalkan. Learning and using context on a humanoid robot using latent dirichlet allocation. In *IEEE International Conferences on Development and Learning and Epigenetic Robotics*, 2014.

[12] H. Celikkanat, G. Orhan, N. Pugeault, F. Guerin, E. Şahin, and S. Kalkan. Learning context on a humanoid robot using incremental latent dirichlet allocation. *IEEE Transactions on Cognitive and Developmental Systems*, 8(1):42–59, 2016.

[13] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[14] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 129–136. IEEE, 2010.

[15] L. Davachi. Item, context and relational episodic encoding in humans. *Current Opinion in Neurobiology*, 16(6):693–700, 2006.

[16] F. I. Doğan, I. Bozcan, and S. Kalkan. A learning based approach to incremental context modeling in robots. *arXiv preprint arXiv:1710.04981*, 2017.

[17] F. I. Dogan and S. Kalkan. Hierarchical context modeling using incremental deep boltzmann machines. *Technical Report*.

[18] F. I. Doğan and S. Kalkan. A deep incremental boltzmann machine for modeling context in robots. *arXiv preprint arXiv:1710.04975*, 2017.

[19] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 281–285. Acm, 1988.

[20] D. Griffiths and M. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in Neural Information Processing Systems*, 16:17, 2004.

[21] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228–5235, 2004.

[22] G. E. Hinton and R. R. Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems*, pages 1607–1614, 2009.

[23] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[24] L. Hu, C. Shao, J. Li, and H. Ji. Incremental learning from news events. *Knowledge-Based Systems*, 89:618–626, 2015.

[25] W. Hwang, J. Park, H. Suh, H. Kim, and I. H. Suh. Ontology-based framework of robot context modeling and reasoning for object recognition. In *International Conference on Fuzzy Systems and Knowledge Discovery*, 2006.

[26] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*. 2013.

[27] Y. Jiang, H. Koppula, and A. Saxena. Modeling 3d environments through hidden human context. *Technical Report*, 2015.

[28] Y. Jiang, M. Lim, C. Zheng, and A. Saxena. Learning to place new objects in a scene. *The International Journal of Robotics Research*, 31(9), 2012.

[29] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] S. Klarman and V. Gutiérrez-Basulto. Description logics of context. *Journal of Logic and Computation*, 2013.

[31] D. Lane. Hierarchy, complexity, society. In *Hierarchy in Natural and Social Sciences*, pages 81–119. Springer, 2006.

[32] H. Larochelle and S. Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716, 2012.

[33] X. Li, J.-F. Martínez, G. Rubio, and D. Gómez. Context reasoning in underwater robots using mebn. *arXiv preprint arXiv:1706.07204*, 2017.

[34] Y. Li, C. Huang, C. C. Loy, and X. Tang. Human attribute recognition by deep hierarchical contexts. In *European Conference on Computer Vision*, pages 684–700. Springer, 2016.

[35] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936. IEEE, 2009.

[36] F. Mastrogiovanni, A. Scalmato, A. Sgorbissa, and R. Zaccaria. Robots and intelligent environments: Knowledge representation and distributed context assessment. *Automatika – Journal for Control, Measurement, Electronics, Computing and Communications*, 52(3):256–268, 2011.

[37] J. McCarthy. Notes on formalizing context. *International Joint Conference on Artificial Intelligence*, pages 555–560, 1993.

[38] S. McKenzie, A. J. Frank, N. R. Kinsky, B. Porter, P. D. Rivière, and H. Eichenbaum. Hippocampal representation of related and opposing memories develop within distinct, hierarchically organized neural schemas. *Neuron*, 83(1):202–215, 2014.

[39] M. G. Ortiz and J.-C. Baillie. Incremental training of restricted boltzmann machines using information driven saccades. In *IEEE International Conferences on Development and Learning and Epigenetic Robotics*, 2014.

[40] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, 2015.

[41] J. Philbin, J. Sivic, and A. Zisserman. Geometric LDA: A generative model for particular object discovery. In *The British Machine Vision Conference*, 2008.

[42] J. W. Rudy. Context representations, context functions, and the parahippocampal–hippocampal system. *Learning & Memory*, 16(10):573–585, 2009.

[43] T. L. Saaty. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26, 1990.

[44] R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.

[45] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[46] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula. Robobrain: Large-scale knowledge engine for robots. *arXiv preprint arXiv:1412.0691*, 2014.

[47] J. Seiter, W.-C. Chiu, M. Fritz, O. Amft, and G. Tröster. Joint segmentation and activity discovery using semantic and temporal priors. In *IEEE Conference on Pervasive Computing and Communications*, pages 71–78. IEEE, 2015.

[48] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[49] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *European Conference on Computer Vision*, pages 746–760, 2012.

[50] A. Smith, T. Hawes, and M. Myers. Hiérarchie: Interactive visualization for hierarchical topic models. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 71–78, 2014.

[51] D. M. Smith and S. J. Mizumori. Hippocampal place cells, context, and episodic memory. *Hippocampus*, 16(9):716–729, 2006.

[52] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015.

[53] N. Srivastava, R. R. Salakhutdinov, and G. E. Hinton. Modeling documents with deep boltzmann machines. *arXiv*, 2013.

[54] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2004–2011. IEEE, 2009.

[55] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[56] M. Tenorth and M. Beetz. Knowrob—knowledge processing for autonomous personal robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[57] A. Torralba. Scene and object recognition in context. `http://www.iapr.org/members/newsletter/Newsletter10-04/index_files/Page705.htm`. Accessed: 2017-12-5.

[58] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[59] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003.

[60] C. Wang, X. Liu, Y. Song, and J. Han. Towards interactive construction of topical hierarchy: A recursive tensor decomposition approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1225–1234. ACM, 2015.

[61] P. Wang, P. Zhang, C. Zhou, Z. Li, and H. Yang. Hierarchical evolving dirichlet processes for modeling nonlinear evolutionary traces in temporal data. *Data Mining and Knowledge Discovery*, 31(1):32–64, 2017.

[62] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2008.

[63] X. Wang and Q. Ji. A hierarchical context model for event recognition in surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2561–2568, 2014.

[64] X. Wang and Q. Ji. Video event recognition with deep hierarchical context model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4418–4427, 2015.

[65] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[66] W. Yeh and L. W. Barsalou. The situated nature of concepts. *The American journal of psychology*, pages 349–384, 2006.

[67] J. Yu, J. Gwak, S. Lee, and M. Jeon. An incremental learning approach for restricted boltzmann machines. In *International Conference on Control, Automation and Information Sciences*. IEEE, 2015.

[68] E. Zavitsanos, G. Paliouras, and G. A. Vouros. Non-parametric estimation of topic hierarchies from texts with hierarchical dirichlet processes. *Journal of Machine Learning Research*, 12:2749–2775, 2011.

[69] A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 558–571. Springer, 2007.