Ambient Assisted Living with Deep Learning

Erinc Merdivan

èse soutenue7139Bordereau de thèse soutenuechapter*.210

NNT : 2019CSUP0006

CentraleSupelec

Ecole Doctorale "Informatique, Automatique, Electronique – Electrotechnique, Mathématiques" (IAEM)

Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)

Thèse de Doctorat

Spécialité : Informatique

soutenue le 17/12/2019 a Metz par

Erinc Merdivan

Ambient Assisted Living with Deep Learning

Composition du jury

Thesis Director	Matthieu Geist
Reporters	Fabrice Lefevre
	Philippe Preux
President of Jury	Herve Frezze-buet
Examiners	Lydia Boudjeloud-assala
	Romuald Elie

Abstract

Ambient Assisted Living (AAL) technologies enables people to maintain their lifestyle in their home. With increase in ageing population all around the world, AAL technologies have gained lot of interest among older population. In recent years, healthcare technologies have improved around the world, thus enabling people to live longer. However, aging brings health problems and requires professional care for the people. The Advanced AAL solutions helps users in self-monitoring their health, provides comfort and improves quality of life in their private home. In order to develop Smart Homes that users would adopt, there is need for certain improvements that should be achieved in certain areas such as activity recognition, privacy, and communication. This work presents deep learning methodologies in the field of activity recognition, privacy preserving and communication.

A Smart Home (SH) system consists of many different modules that interact with each other. It is also crucial that all these interactions should be extendable and easily scalable. In this work, a SH architectural design was proposed that encapsulates different modules i.e privacy-preserving data management module, activity recognition, and voice assistant. The proposed event-based design enables different modules to be plugged into the main system with ease and requires each module to listen to the events that are published and publish their output to the main system. In a Smart Home environment, the system can be idle most of the time and only needs to react if a particular event happens. Therefore, the proposed event-based system is suitable and shows how our improvement can be incorporated into the overall design.

Activity recognition is one of the primary components of Smart Home systems. Previous works have performed activity recognition using statistical and standardized machine learning models such as Naive Bayes, Hidden Markov Models (HMM) and Conditional Random Fields (CRF). In this work, we were the first to propose a deep learning-based activity recognition model using Long Short Term Memory networks (LSTM). LSTM model achieves state-of-art results for Smart Home activity recognition. Later, Convolutional neural networks (1D-CNN) based models are proposed which matches LSTM based model performance in most cases while reducing training time. In real home settings, there is more than one person living together. Therefore, the activity

recognition module should also able to handle multi-occupancy in SH environment. Another important aspect to be investigated is how to handle class imbalanced problems that happen naturally in SH setting. In this regard, different LSTM-based models are developed along with different known methods to overcome class imbalance problems in SH dataset.

Privacy is getting more attention in today's world since various personal data is gathered, stored and transferred more commonly. We developed an encoder-decoder based model which encodes user data and decodes according to privacy settings. This model allows data to be shared in an encoded version instead of raw data. After decoding only certain parts of data is visible to the third party. In the proposed encoder-decoder, neural network weights are encrypted and stored in a file. These weights can be distributed among many files which makes the model more robust. In this order encoding to be solved an attacker needs access to a large amount of encoded vectors-decoded output pairs in order to train a decoder to decode encoded data. This process would relieve privacy concerns while transmitting user data and also secures the data when it is stored since even if communication is compromised only encoded data will be visible to the user.

In addition, user interaction is vital in a Smart Home system. Current user interaction is limited in terms of dialogue management. Data can be in many different formats in the smart home and health care domain and may include format on the text such as tables (health records, medicine details, etc.). Processing this data which includes textual information with visual structures requires different processing techniques rather than only textual processing. Deep learning vision models are applied to test this hypothesis to see if Natural Language Processing (NLP) tasks can be achieved while only using visual processing. It is showed that using only visual processing very close performance has been achieved in many tasks and outperformed in a certain task for sentiment analysis. Visual processing of NLP task can be also beneficial in certain dialogue settings where language patterns are statistic however only tokens changes or in certain information retrieval cases where information is stored with a structure hard to capture through NLP. Deep learning-based models are also applied to this task by showing some promises that they can be applied as a complementary to NLP modules to enrich them with visual cues. Reinforcement learning is very suitable to be applied to dialogue management. However, certain advancement should be achieved, such as a diverse dialogue dataset, accurate reward function, and sample efficient reinforcement learning algorithms. In

this regard, First, diverse dialogue data is collected from open domain dialogue with human annotations to rank dialogue-reply pairs as well as human alternatives for same dialogue history to train models which can produce and rate diverse human replies. Secondly, different deep learning-based models are trained to access dialogue history reply pair quality. Lastly, sample efficient reinforcement learning methods are developed. The developed abstract framework can be instantiated with different configurations in order to be applied to dialogue management systems. It is shown that the new deep RL framework is more sample efficient than state-of-art RL algorithms in continuous control tasks.

Overall, this thesis presents different improvements on various components of SH systems and different challenges are addressed for activity recognition, privacy and dialogue management.

Abstract(in French)

Les technologies d'assistance à l'autonomie à domicile (AAD) permettent aux individus de maintenir leur mode de vie dans leurs domiciles. Avec l'augmentation de la population vieillissante dans le monde entier, les technologies AAD ont gagné beaucoup de popularité auprès des personnes plus âgées. Ces dernières années, les technologies médicales se sont améliorées partout dans le monde, permettant ainsi aux gens de vivre plus longtemps. Cependant, le vieillissement entraîne des problèmes de santé et nécessite une prise en charge professionnelle. Les technologies AAD avancées aident les utilisateurs à auto-surveiller leur santé, leur apportent un confort et améliorent leur qualité de vie dans leur domicile personnel.

Afin de développer des Smart Homes que les utilisateurs adopteraient, il est nécessaire d'apporter certaines améliorations dans certains domaines tels que la reconnaissance des activités, la confidentialité et la communication. Ce travail présente des méthodologies d'apprentissage profond dans le domaine de la reconnaissance d'activité, de la préservation de la confidentialité et de la communication.

Un système de maison intelligente (MI) se compose de plusieurs modules dierents qui interagissent entre eux. Il est aussi crucial que toutes ces interactions soient expansibles et facilement modulables. Dans ce travail, une conception architecturale d'une MI

a été proposée qui intègre les différents modules, c'est-à-dire le module de gestion des données protégeant la confidentialité, la reconnaissance d'activité et l'assistant vocal. La conception basée sur les événements proposée permet aux modules diérents d'être facilement intégrés au système principal et exige à chaque module de suivre les événements qui sont publiés et de publier leurs résultats dans le système principal. Dans un environnement de maison intelligente, le système peut être inactif la plupart du temps et ne réagir qu'à la survenue d'un événement particulier. Par conséquent, le système basé sur les événements proposé est adapté et montre comment notre amélioration peut être intégrée dans la conception globale.

La reconnaissance d'activité est l'un des composants fondamentaux des systèmes de maison intelligente. Les travaux antérieurs ont effectué la reconnaissance d'activité en utilisant des modèles statistiques et standardisés d'apprentissage de machine tels que Naive Bayes, Hidden Markov Models (HMM) et Conditional Random Fields (CRF).

Dans ce travail, nous étions les premiers à proposer un modèle de reconnaissance d'activité basé sur l'apprentissage profond en utilisant les réseaux Long Short Term Memory (LSTM) (en Français longue mémoire à court terme). Le modèle LSTM obtient des résultats de haute qualité pour la reconnaissance d'activités de maisons intelligentes.

Plus tard, des modèles basés sur les réseaux neuronaux convolutionnels (1D-CNN) sont proposés qui ont des performances équivalentes à celles du modèle LSTM dans la plupart des situations, tout en réduisant le temps d'apprentissage. Dans un milieu domestique réel, il y a plusieurs personnes qui vivent ensemble. De ce fait, le module de reconnaissance d'activité devrait aussi être capable de gérer l'occupation collective dans un environnement MI. Un autre aspect important à étudier est la manière de gestion des problèmes de déséquilibre entre les classes qui surviennent naturellement dans le milieu MI. À cet égard, diérents modèles basés sur LSTM sont développés en parallèle avec plusieurs méthodes reconnues de surmonter les problèmes de déséquilibre de classe dans l'ensemble de données de MI.

La confidentialité attire plus d'attention ces jours-ci, depuis que diverses données personnelles sont recueillies, stockées et transférées plus fréquemment. Nous avons développé un modèle basé sur un encodeur-décodeur qui encode les données des utilisateurs et les décode en fonction des paramètres de confidentialité. Ce modèle permet de partager des données sous forme d'une version encodée au lieu de partager des données brutes.

Après le décodage, seules certaines parties des données sont visibles par le tiers. Dans le codeur-décodeur proposé, les pondérations du réseau neuronal sont cryptées et stockées dans un fichier. Ces pondérations peuvent être réparties sur plusieurs fichiers, ce qui rend le modèle plus robuste. Pour décrypter cet encodage, un attaquant doit avoir accès à une grande quantité de paires de vecteurs encodés-résultats décodés afin de pouvoir entraîner un décodeur à décoder les données encodées. Ce processus permettrait de dissiper les soucis de confidentialité lors de la transmission des données de l'utilisateur et de sécuriser les données lorsqu'elles sont stockées, car même si la communication est compromise, seules les données codées seront visibles pour l'utilisateur.

En plus, l'interaction de l'utilisateur est cruciale dans un système de maison intelligente. L'interaction actuelle de l'utilisateur est limitée en termes de gestion du dialogue. Les données peuvent être dans plusieurs formats diérents dans le domaine de la maison intelligente et le domaine médical et peuvent inclure des formats sur le texte tels que des tableaux (dossier médical, détails de médicaments, etc.). Le traitement de ces données, qui comprennent des informations textuelles avec des structures visuelles, nécessite une variété de techniques de traitement plutôt qu'un traitement textuel isolé. Il est démontré que l'utilisation du traitement visuel seul a permis d'obtenir des performances très proches dans de nombreuses tâches et des performances supérieures dans une certaine tâche d'analyse de sentiments.

Le traitement visuel des tâches TALN peut également être bénéfique dans certains paramètres de dialogue où les structures linguistiques sont statistiques, mais que dans les cas des changements symboliques, ou de recherche d'informations où l'information est stockée avec une structure difficile à saisir par le TALN. Les modèles basés sur l'apprentissage profond sont également appliqués à cette tâche et montrent certaines indications sur la possibilité de les appliquer en complément des modules de TALN pour les enrichir d'indices visuels.

L'apprentissage par renforcement est très adapté à la gestion du dialogue. Cependant, certains progrès devraient être accomplis, comme un ensemble diversifié de données de dialogue, une fonction de récompense précise et les algorithmes d'apprentissage par renforcement ecient à l'échantillon. À cet égard, tout d'abord, l'ensemble diversifié des données de dialogue est collecté à partir du dialogue de domaine ouvert avec des annotations humaines pour classer les couples dialogue-réponse ainsi que les alternatives

humaines pour la même histoire de dialogue afin d'entrainer des modèles qui peuvent produire et classer diverses réponses humaines. Deuxièmement, des modèles diérents basés sur l'apprentissage profond sont entraînés pour accéder à la qualité des couples histoire du dialogue-réponse. Enfin, les méthodes de d'apprentissage par renforcement ecient à l'échantillon sont développées. Le cadre abstrait développé peut être instancié avec des congurations différentes pour être appliqué aux systèmes de gestion du dialogue. Il est démontré que le nouveau cadre deep RL est plus ecient à l'échantillon que les algorithmes RL de haute qualité dans les tâches de contrôle continu.

Globalement, cette thèse présente les diérentes améliorations sur divers composants des systèmes MI et les diérents défis sont abordés pour la reconnaissance d'activité, la confidentialité et la gestion du dialogue.

Acknowledgements

My Ph.D. was carried out with funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie GA ACROSSING No 676157.

I would like to thank my Ph.D. supervisor Matthieu Geist, for his guidance, encouragement and expert advice throughout my Ph.D. work. Matthieu always had time for me to discuss any idea, experiment or even administration issues. I am very grateful for the skills I developed for the research and also for the rest of my career. His knowledge on reinforcement learning and wisdom is tremendous and I benefited a lot during my studies.

I am very grateful for the Austrian Institute of Technology GmbH (AIT) to employ me and give me freedom on my research. AIT provided me everything, such as hardware, software, and training to complete my Ph.D. My supervisors in AIT, Sten Hanke, and Johannes Kropf brought me to Acrossing project and gave me perfect guidance on Ambient Assisted Living and also for career development. I also enjoyed greatly and learned a lot from my colleague Deepika Singh at AIT working together in Acrossing. Liming Chen, our Acrossing Project coordinator, managed the project seamlessly and gave many opportunities for me to connect with great researchers from all around Europe.

I could not finish my Ph.D. without constant support from my wife Julie and from my parents who supported me in all aspects of my project. They always had time for me in any matter.

Lastly, all my colleagues in Acrossing and AIT, I benefited from all of their knowledge and expertise and I gained very close friends for the rest of my life.

Contents

List of Figures

List of Tables

Abbreviations

AAL	**A**mbient **A**sisted **L**iving
AE	**A**uto **E**ncoder
AIT	**A**ustrian **I**nstitute of **T**echnology
AMPI	**A**pproximate **M**odified **P**olicy **I**teration
AMT	**A**mazon **M**echanical **T**urk
ASR	**A**utomatic **S**peech **R**ecognizer
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
CNN	**C**onvolutional **N**eural **N**etwork
CPI	**C**onservative **P**olicy **I**teration
CRF	**C**onditional **R**andom **F**ield
DM	**D**ialog **M**anager
DQN	**D**eep **Q**-**N**etwork
DRS	**D**ialog **R**esponse **S**election
DST	**D**ialog **S**tate **T**racker
EBGAN	**E**nergy **B**ased **G**enerative **A**dverserial **N**etwork
ECA	**E**mbodied **C**onversational **A**gent
GAN	**G**enerative **A**dverserial **N**etwork
GRU	**G**ated **R**ecurrent **U**nit
HAN	**H**ierarchical **A**ttention **Networks**
HAR	**H**uman **A**ctivity **R**ecognition
HCI	**H**uman **C**omputer **I**nteraction
HMM	**H**idden **M**arkov **M**odel

HUMOD	**H**uman **A**nnotated **D**ialogues **D**ataset
IOT	**I**nternet **O**f **T**hings
LM	**L**anguage **M**odel
LSTM	**L**ong **S**hort **T**erm **M**emory
LU	**L**anguage **U**nderstanding
MDP	**M**arkov **D**ecision **P**rocess
MoPPO	**M**odified **P**roximal **P**olicy
MoSoPI	**M**odified **S**oft **P**olicy **I**teration
MPI	**M**odified **P**olicy **I**teration
NLG	**N**atural **L**anguage **G**eneration
NLP	**N**atural **L**anguage **P**rocessing
NLU	**N**atural **L**anguage **U**nderstanding
NUC	**N**ext **U**tterance **C**lassification
OCR	**O**ptical **C**haracter **R**ecognition
OOV	**O**ut **O**f **V**ocabulary
PI	**P**olicy **I**teration
POMDP	**P**artially **O**bservable **M**arkov **D**ecision **P**rocess
PPO	**P**roximal **P**olicy **O**ptimization
PU	**P**ositive **U**nlabeled
RCN	**R**econstruct **C**rush **N**etwork
RL	**R**einforcement **L**earning
RNN	**R**ecurrent **N**eural **N**etwork
SAC	**S**oft **A**ctor **C**ritic
SH	**S**mart **H**ome
SPI	**S**oft **P**olicy **I**teration
TRPO	**T**rust **R**egion **P**olicy **O**ptimization
TTS	**T**ext **T**o **S**peech
UI	**U**ser **I**nterface
VI	**V**alue **I**teration

CHAPTER 1

Introduction to Ambient Assisted Living

1.1 Ambient Assisted Living technologies

Ambient Assisted Living (AAL) technologies are gaining immense popularity due to growing demands in healthcare sector for the population around the world. In recent years, the increase in aging population around the world has become a major concern. According to a report from World population prospects [6], the growth of people aged 60 years or older is projected to accelerate in the coming years. This population numbered 962 million in 2017, which is more than double compared to 1980. The number of older persons is expected to double again by 2050 and is projected to reach nearly 2.1 billion. Aging leads to various problems ranging from basic functional disabilities to severe health problems such as diabetes, osteoarthritis, depression, pulmonary diseases, and dementia. Along with medical problems, dependence on family or caregivers for simple daily activities causes embarrassment and sedentariness which leads to poor nutrition [7]. To overcome these problems, various research projects have been introduced and implemented successfully such as the ACROSSING project for developing smart solutions, the Cogni Win project for personalized and adaptable interfaces for older people, the RelaxedCare project which introduced an assistive system to provide informal care, the SUCCESS project which support user through conversation with avatar and many more. This research is a part of the ACROSSING ITN Horizon 2020 project, which combines the efforts of a multi-disciplinary network of 26 leading European research groups, industry partners, and user organizations, to develop an open SH technology infrastructure. The subproject described in this manuscript, is part of the Austrian Institute of Technology's (AIT) contribution to the ACROSSING project. ACROSSING project is designed such that there are four Scientific and Application groups and this work is part of the scientific group, that focuses on Open smart home platforms and service infrastructure. Therefore, as a part of this scientific group, we are focusing on three important components: improving activity recognition, addressing privacy concerns and developing intelligent dialogue systems for AAL systems, with an emphasis on a framework which is flexible and scalable for real-world applications.

AAL solutions provide various functionalities (health monitoring, social participation, fall detection, physical exercise monitoring, home monitoring and robotic support, etc.) that support older people in living independent and confident lives. Various other factors contribute towards developing a successful AAL solutions —a system should meet the needs and demands of the user, provide security and privacy in data, and show timeliness, reliability, interoperability, and robustness.

AAL systems are increasingly complex, consisting of various modules that cater for different functionalities, with the constraint of performing well in a dynamic environment. Among various AAL solutions, smart homes have gained a lot of attention due to the versatile applications in the field of Internet of Things (IoT). Smart homes provide various functionalities that may reduce the need for a caregiver through the development of smart solutions which support older people to maintain their autonomy for many tasks.

1.2 Smart Homes

Smart home (SH) is defined as the living environment where the IoT devices installed in the home environment have the capability to interact with each other and with the resident living inside [8]. These devices include smart electronic appliances such as a television, refrigerator, washer, etc.; safety and security systems (cameras, monitors and emergency alarms) and smart energy equipment (thermostats, sensors, lighting, and heating) which are interconnected using standardized communication protocols [9]. Recent trends show that there has been major growth in the market of SH devices and rise in demands for home health care products, assisted living and energy consumption solutions. According to the report in [10], the global market of SHs is expected to reach USD 53.45 billion by 2022 and industry analysis shows a compound annual growth rate (CAGR) of 14.5% between 2017 and 2022.

SH systems are generally designed as a “bottom-up” process [11] as illustrated in Fig. 1.1. In this framework, SH consists of many different sensors installed in different parts of

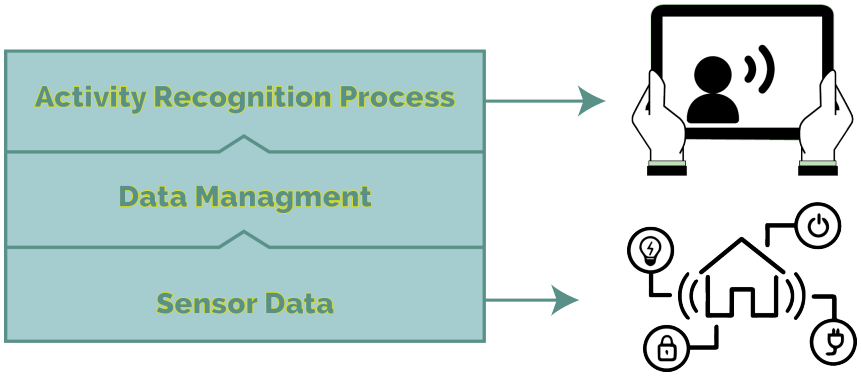


FIGURE 1.1: Smart Home with bottom-up approach.

the living environment. Sensor readings are collected and managed by the data module and shared with activity recognition module, then classified into a predefined set of activities that may occur in daily life. These activities can cover a wide range of frequent (walking, sitting, watching TV, etc.) or less frequent (fall detection, laundry) activities. The activity recognition module provides its predictions about the current activity and state of the user to communication mechanisms that are deployed in the SH setting. The communication mechanism is the user interface which enables humans to interact with SH and vice versa. Different modalities of User Interface (UI) can be implemented in SH. It can be visual UI through a touch screen or an assistant that is capable of processing and producing speech input. More complex UIs would combine many different modalities to give the most flexibility to user.

Although SHs are growing rapidly, there are barriers that need to be overcome [12]. Previous works have introduced various SH frameworks that focused only on one particular component of the system: for example, only activity recognition module, adaptive interface or privacy in SH IoT devices. A SH environment consists of three major modules —activity recognition module, privacy preserving data management module and dialogue manager. An activity recognition module is used to predict the activity of user, the privacy module handles how user data is stored, processed and shared and the dialogue module helps in communication and interaction with the SH. Each component of the system is equally important and integration of these modules in the smart environment will help in developing a robust and secure SH.

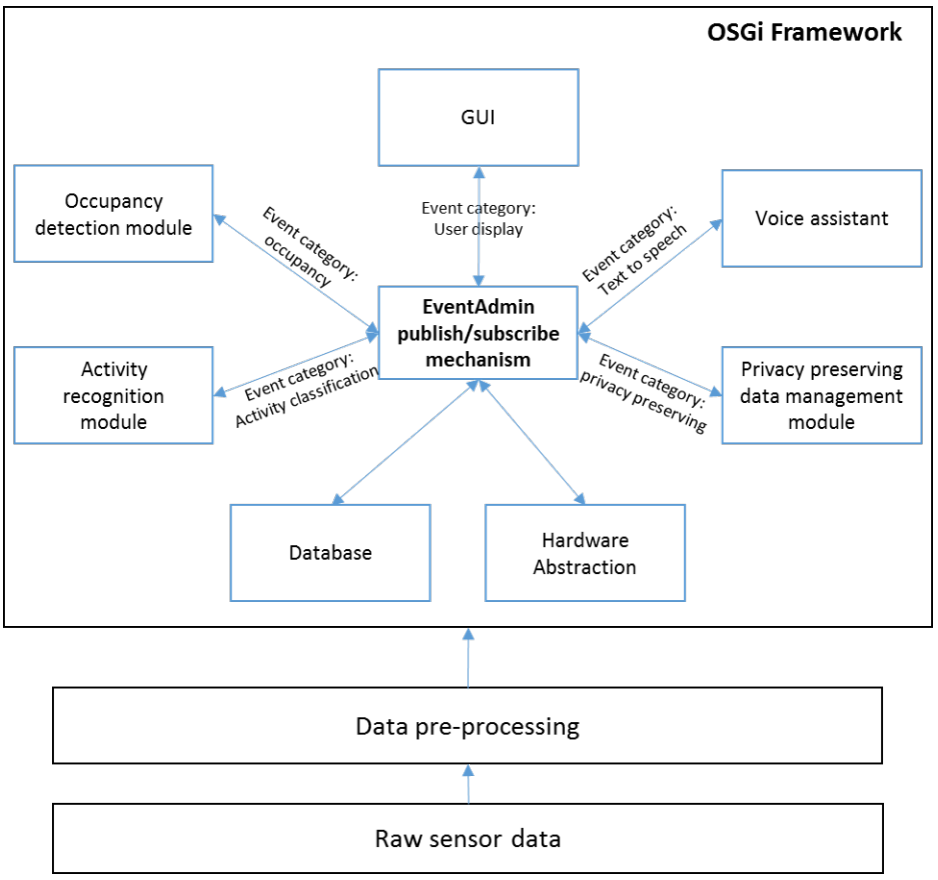


FIGURE 1.2: Integration of activity recognition, privacy preserving data management and dialogue manager in smart home framework.

The research we focused on has three parts: activity recognition, privacy preserving data management, and dialogue management. All these parts are experiencing a shift towards more deep learning models since such models require less human expertise through feature engineering and also achieve state-of-art results. We first started by developing a theoretical framework for a SH which combines main components. This is shown in Fig. 1.2.

In the proposed framework, the Home Event Recognition System (HOMER) [13] acts as a middleware to integrate each module and sensor components of the system. HOMER is an open-source platform based on the Apache Karaf OSGi framework and enables modularity by encapsulating its functionalities in terms of OSGi bundles. Each module of the framework is linked with the centralized module called EventAdmin which performs

communication between the modules by sending and receiving asynchronous events. It is based on the subscribe/publish mechanism in which modules of the framework need to subscribe to EventAdmin in order to send and receive events. A detailed description of the activity recognition, privacy preserving data management and dialogue manager modules are presented in the following section.

1.3 Activity recognition

Human activity recognition (HAR) in SH is one of the most widely researched topic in the field of AAL. It aims at determining the daily living activities of a person or a group of persons using raw data in the home environment. The monitoring systems are categorized mainly into three categories: (1) camera-based systems, (2) wearable devices and (3) sensors. Camera-based monitoring is not preferred by the majority of users as it raises privacy concerns regarding collected data. Wearable devices provide good accuracy in a personalized system and do not raise as many privacy concerns; however, such systems are not practical to use while monitoring long-term activities. As a result, the most preferred solution for activity recognition is using unobtrusive sensors from a smart home environment. A SH consists of various sensors such as motion sensors, door sensors, window sensors, thermostat, passive infrared sensors (PIR), energy tracking switches, light sensors, cameras, and voice-assistants, which communicate with each other and collect data to monitor user activities. Also, installation of sensors depends on the activities to be monitored. The data collected from these embedded sensors and smart devices can provide different services to the residents such as safety and guidance features by user behaviour monitoring, activity recognition and fall detection; home automation by controlling lights, doors, windows, temperature and energy consumption; and security with alarms, lock/unlock of doors and monitoring of outsiders in the absence of resident. There are several steps involved in the process of HAR. These steps include: (1) pre-processing of the raw sensor data by removing noise, outliers and redundant data, performing data aggregation and normalization; (2) identifying the most significant data

segments through segmentation techniques; (3) feature extraction (e.g. temporal and spatial information) from the segmented data; (4) dimensionality reduction that decreases the number of features to increase their quality and reduces the computational effort needed for the classification; and (5) classification of activities using machine learning techniques.

Previous works have used various data-driven approaches and knowledge-driven approaches for activity recognition, e.g. [14, 15]. Currently, deep learning has contributed towards activity recognition systems [16] and tends to overcome the limitations of conventional pattern recognition approaches. In this research, we focus on deep neural networks, since they have the capability to learn features automatically from raw sensor data instead of using manual feature handling, which is a costly and time consuming process. The deep generative networks can also make use of unlabeled data for training model since it is not feasible many times to collect labelled data [16].

1.4 Privacy preserving data management

An SH environment is typically built using IoT technology. In such technology, a communication network extends the existing internet infrastructure by including everyday items and sensors. In the SH scenario, this is done to help monitoring and managing daily home activities and interaction with users through a dialogue manager, as discussed in the previous section. This results in the generation of huge amounts of diverse, heterogeneous, complex, and distributed data from a multitude of applications and sensors. This generated data typically contains private and sensitive information such as the user's profile, location, pictures and daily activities that a user does not want to share. Such information raises privacy and security concerns among the user and also causes barriers for the adoption of SH systems.

Various studies have been performed to understand users' privacy concerns and proposed potential solutions for them [17]. Other studies [18] aim to understand how and if

contextual factors affect users' privacy perceptions in IoT environments. There are a number of different privacy concerns in the SH setting. The majority of concerns are where to store user private data, how data can be shared with third parties and how to handle different privacy levels for third parties (doctors, care givers, family members, etc.) that must be able to access user data.

However, despite the many advances in IoT technology and Activity Recognition methods, there is still a lack of privacy preserving methods able to resolve these issues (data access authorization, processing of large volume of heterogeneous data and secure transmission). One of the main reasons preserving privacy is a challenge is because of the unstructured and heterogeneous data sources which means that data may be in various formats such as text, photo, video, etc. Also, as more data is collected about individuals, it is becoming more challenging to reliably protect the identity of each individual, especially when there are multiple stakeholders that require access to different subsets of the collected data. Another reason behind this is that SHs not only collect personal information but can also monitor the users' activities. Unfortunately, it is increasingly hard to obfuscate the data in a way that it protects sensitive information while maintaining its utility.

Numerous methods have been proposed in previous work, which endeavor to eliminate the privacy risks, dividing the privacy research field into three main areas. The first area focuses on the transformation of data before they are distributed. Research in this area has mostly focused on centralized healthcare systems [19] and not decentralized AAL environments. The second area focuses on private data management and storage in cloud environments [20], while the third area focuses on access control [21]. However, as the applications and environments become more complex, these methods are not always enough to meet all the privacy requirements. Currently popular de-identification techniques are not sufficient. Either personally identifiable information is not sufficiently protected, or the resulting data no longer represents the original data.

Privacy-preserving mechanisms are necessary in the SH environment to protect the sensitive data and the privacy of the users. This protection is even more important in cases where Deep Learning based models are used since they often require a vast amount of data to train the model. It is crucial to develop privacy preserving methods to handle the sharing of user data and how third parties can have access on user data. In this research, we focus on how user data can be processed by third parties without sharing raw data and how handling different privacy levels on user data, by revealing only the permitted amount of data to third parties.

1.5 Dialogue manager

Dialogue is the most natural way to communicate for humans. It is learned early in childhood and gets more and more complex throughout the later stages of life.Even elderly with serious personal disorders retain their face-to-face communication. Dialogue can be conducted between human participants as well as between humans and an intelligent agent. Direct natural interaction as dialogue, with the SH through User Interface (UI), is an important aspect in terms of user experience and user acceptance of SH technologies, especially for older people [22]. Face-to-face communication in SH environments can be used as virtual avatars. These avatars' appearance can either be realistic (human-like), a wooden mannequin character [23], or without any embodiment such as Amazon Alexa or Apple Siri. Avatars can be treated as Embodied Conversational Agents (ECAs) with visual output (such as the avatar showing emotions or gestures) accompanying the speech or textbase natural language output. ECAs have intelligent dialogue managers which are capable of Natural Language Processing (NLP). NLP has a wide range of applications in Human Computer Interaction (HCI) in various components such as Natural Language Generation (NLG), Natural Language Understanding (NLU) and Dialogue Processing (DP) tasks. The dialogue pipeline is composed of main components, which are responsible for understanding the user's natural language input and generating system reply, and complementary components, which are responsible for converting user speech to text and

system text response to speech [24]. Two complementary modules, Automatic Speech Recognition (ASR) and text-to-speech (TTS) synthesizer, are responsible for converting spoken input to text and synthesizing speech from text created by dialogue manager. Core components of the dialogue module are Natural Language Interpreter (NLI), Dialogue State Tracker (DST) and Dialogue Response Selection (DRS). NLI converts the text input to features that are processed by DST to update current dialogue state. Current dialogue state is used by DRS to generate a textual reply to user. First dialogue system that was developed is ELIZA [25] which was a rule-based system. Later, other types of dialogue managers were developed using intelligent dialogue managers which can be trained, rather than using human created rules. Dialogue managers are either designed with hand-crafted rules and dialogue flow or trained with machine learning models, but they are still far from human level dialogue. However, in most tasks for which DMs are deployed, their limited capability of dialogue can be enough, such as for airline booking or for restaurant reservation.

1.6 Report Outline

This manuscript is organized as follows.

Chapter 2 presents different SH architectures [26] and a detailed literature review on activity recognition in the SH setting for single and multiple occupancy, privacy enabled data management and intelligent dialogue systems [27].

- **Merdivan E.**, Singh, D., Hanke S., Holzinger, A.-Dialogue Systems for Intelligent Human Computer Interactions. *Electr. Notes Theor. Comput. Sci.*, 2019
- Singh D., Psychoula I., **Merdivan E.**, Kropf J., Hanke S., Sandner E., Chen L., Holzinger A.-Privacy enabled Smart Home framework with Voice assistant. *to appear in book Smart Assisted Living:Toward An Open Smart-Home Infrastructure*, 2019

Chapter 3 presents the background of human activity recognition (HAR), with an emphasis in activity classification based on SH sensor data for single and multi-occupancy. Until recently, activity recognition in the SH setting was performed using traditional machine learning models such as Hidden Markov Models, Conditional random fields and Naive Bayes. We implemented deep learning based approach using LSTMs and 1D-CNNs for single-occupancy [28, 29]. Deep learning models outperform existing models in activity recognition and do not require any feature engineering. Another advantage of using deep learning models is their scalability to large amounts of data.

- Singh, D.*, **Merdivan, E.***, Hanke, S., Kropf, J., Geist, M. and Holzinger, A., 2017. Convolutional and recurrent neural networks for activity recognition in smart environment. In *Towards integrative machine learning and knowledge extraction* (pp. 194-205). Springer, Cham.

- Singh, D.*, **Merdivan, E.***, Psychoula, I., Kropf, J., Hanke, S., Geist, M. and Holzinger, A., 2017, August. Human activity recognition using recurrent neural networks. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (pp. 267-274). Springer, Cham.

Chapter 4 describes the work done on how to share user private data with different parties, taking into account different privacy settings [30]. In the SH setting training data comes from the user that resides in SH. This user data is highly sensitive and raises privacy concerns for users. We developed a deep learning method which enables users to share encoded data instead of raw data with third parties [31]. Third parties can then use the decoder, which is based on deep neural networks, to decode the encoded data to view amount of private information that users decide. We also show that researchers can work on encoded data instead of raw data in order to train deep learning based models which would reduce the privacy concerns on using data but still allows deep learning models to use data for training.

* denotes equal contribution.

- Psychoula, I., **Merdivan, E.**, Singh, D., Chen, L., Chen, F., Hanke, S., Kropf, J., Holzinger, A. and Geist, M., 2018, March. A deep learning approach for privacy preservation in assisted living. In 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 710-715). IEEE.

- Psychoula, I., **Merdivan, E.**, Singh, D., Chen, L. , Wagner, I. and Geist, M., 2020, March. Privacy Preservation with Deep Learning: The Encoders-Decoders Method, *(article under preparation)*

Chapter 5 focuses on intelligent dialogue systems which can operate in the task oriented or open dialogue settings. It includes dataset collection and designing deep neural networks for open-end dialogue [32]. After dataset collection, we present results on achieving NLP task with different modality [33] as well as deep learning-based energy models for task-oriented dialogue [34]. It also includes a newly developed, sample efficient deep reinforcement learning approach [35], which would be interesting for training dialogue managers since collecting samples for training dialogue managers are very expensive and time consuming.

- **Merdivan E.**,Singh, D.,Hanke S., Holzinger, A., Geist M.-Human Annotated Movie Dialogues Dataset (HUMOD) *(Applied Sciences,MDPI)*, 2020

- **Merdivan E.**, Vafeiadis A., Kalatzis D., Hanke S., Kropf J., Votis K., Giakoumis D., Tzovaras D., Chen L., Hamzaoui R., Geist M.-Image-based Text Classification Using 2D Convolutional Neural Networks, *IEEE Smart World Congress*, 2019

- **Merdivan E.**, Loghmani MR., Geist M. - Reconstruct And Crush Network *Conference on Neural Information Processing Systems (NeurIPS)*, 2017

- **Merdivan E.**, Hanke S., Geist M. - Modified Actor-Critics *(extended abstract in Conference on Autonomous Agents and MultiAgent Systems (AAMAS))*, 2020

Eventually, Chapter 6 concludes and summarizes the report.

CHAPTER 2

Literature Review

2.1 Existing SH Projects

The concept of SHs has gained popularity in the early 2000s. Lutolf [36] defined SH concept as the integration of different services within a home environment by using a common communication system. According to Satpathy [37] a SH provides independence and comfort to the residents by using mechanical and digital devices interconnected in a network with the ability to communicate with the user in order to create an interactive space.

Various SH projects have been conducted in previous works, with varied objectives and desired services. For example, in [38], SH projects were categorized according to the intended services, i.e. comfort, healthcare, and security.

One of the main objectives of SH is to provide comfort to the resident, which can be achieved by two ways: activity recognition with event automation and remote management from different locations. The SH projects which focus on activity recognition systems and remote management services are MavHome (Managing an Adaptive Versatile Home) [39]; Adaptive Control of Home Environment (ACHE) system, developed by Mozer in the USA [40]; House of Matilda, which was designed by University of Florida [41]; and Gator Tech [42], in which middle-ware architecture, based on OSGi framework, has been proposed to enable integration and communication between smart devices. The Easy Living project of Microsoft [43] created a dynamic smart environment to track multiple residents using distributed image processing system. A "Ubiquitous Home" has been constructed by Yamazaki [44], which is a real-life testbed for context-aware services. The system supports new home services by connecting devices, sensors, and appliances in the home network. The most popular system, The Center for Advanced Studies in Adaptive Systems (CASAS) [45], an adaptive SH, has been developed at Washington State University. The project has benefited researchers by making the dataset publicly available to compare different methodologies for activity recognition.

The SH systems which enable users to monitor and control their home environment are Simple Object Access Protocol (SOAP)-based SHs [46], Service-oriented SH architecture based on OSGi and mobile-agent technology [47]. Another solution is to combine CWMP (CPE Wide area network Management Protocol) with UPnP following DSL forum standardization [48].

SH systems that are developed to provide healthcare facilities for patients, elderly and healthy people are based on local monitoring and are implemented on-site as stand-alone solutions. Some also provide remote monitoring to provide emergency support to the users. Some of these projects are: A Health Integrated SH Information System (HIS) [49], ENABLE [50] and SELF (Sensorized Environment for LiFe) [51].

The projects which are more inclined towards security in SH are [52], which is a product-based security model for SH; Intuisec [53], a user-intuitive security framework for SHs and [54], an authentication mechanism for secure remote access to an SH network. The system uses a HMAC-based One Time Password (HOTP), a hashchaining technique, and smart cards based on a strong password approach.

Our contribution. In Chapter 1.2, we presented our theoretical framework, which uses an event-based design to connect different components; activity recognition, privacy-enabled data management and dialogue manager. Different modules are developed as part of ACROSSING Project (EU Marie Curie ITN). Our work uses HOMER as middle and can integrate separate modules easily by linking each module to only centralized module. Our proposed module focuses on how to enable deep learning based activity recognition modules while preserving user privacy and communicate naturally with the user.

2.2 Activity Recognition

Human activity recognition is one of the most prominent research topic in the field of AAL. The goal of activity recognition is to identify and detect simple and complex activities of the user in a smart environment setting. Sensor-based activity recognition has been characterized into four basic tasks [55]. These tasks include: 1) selection and deployment of appropriate sensors to objects and environments in order to monitor and capture users' behavior along with environment; 2) processing of perceived information through data analysis techniques and/or knowledge representation formalism at appropriate levels of abstraction; 3) creating computational activity models in a way that allows software systems/agents to conduct reasoning and manipulation; and 4) development of reasoning algorithms to infer activities from sensor data. The selection of the method for activity recognition depends on the specific task. The input data can be in the form of video or sensor data; therefore, activity recognition systems have been broadly classified into two main categories. The first is vision-based activity recognition, which uses visual sensing technologies such as cameras in order to monitor user behaviour and surroundings. The data generated is in the form of video sequences or digitized visual data. The second type is sensor-based activity recognition, which uses sensor network technologies for activity recognition and monitoring. The generated sensor data from sensor-based monitoring are mainly in the form of time series including various parameter values. In these approaches, sensors, which can be wearable or smartphones, are attached to an actor or attached to the various objects in a SH setting such as a door, window, light, television, etc. Wearable sensors generally use inertial measurement units and radio frequency identification (RFID) tags to collect users' behavioral information. This approach is effective in recognizing physical movements such as physical exercises and in developing personalized solutions. In contrast, sensors installed in SH use multiple multi-modal sensors to generate data for user behavior analysis.

Activity recognition is a challenging task, as the data generated from the sensors are sometimes ambiguous with respect to the activity taking place. This causes ambiguity in

the interpretation of activities. Sometimes the data obtained can be noisy as well. Noise in the data can be caused by humans, sensors or due to an error in the network system which fails to give correct sensor readings. Several solutions have been proposed for activity recognition methods in previous works; these can be classified into two main categories [56]: data-driven and knowledge-driven.

2.2.1 Data-driven approaches

Data-driven methods use machine learning techniques to develop the activities model from the sensor data. These approaches are good in handling uncertainty and noisy data but requires large annotated dataset for learning and training the model [57]. In [55], data-driven approaches are further classified into generative and discriminative based approaches. The generative approaches include Naive Bayes classifier (NBC), Hidden Markov Models (HMM) and Dynamic Bayesian Networks (DBN). Naive Bayes classifiers yield good accuracy when large amounts of sample data are provided but do not explicitly model any temporal information which is usually considered important in activity recognition [58]. The Hidden Markov Model (HMM) is the most frequently used generative approach as it includes temporal information. It is a probabilistic model which can learn and interpret data and is efficient to implement. The HMM model has been implemented in [59, 60] to recognize activities using feature extracted from sensor events according to a sliding window. HMMs are the basis of statistical temporal models and are a special form of general Dynamic Bayesian Networks. In [61], DBNs are used to simultaneously track persons and model their activities using a variety of simple sensors. However, HMMs and DBNs have some limitations. An HMM is not capable of capturing long-range or transitive dependencies of the observations due to its very independence assumptions. Furthermore, without significant training, an HMM is not capable of recognizing all possible observation sequences that can be consistent with a particular activity.

A drawback of the generative approach is that it requires enough data to learn the complete probabilistic representations. Discriminative based approaches focus directly on solving the classification problem instead of the representation problem. These approaches include Neural networks (NN) [62], Support vector machines (SVM) [63] and Conditional Random Fields (CRF) [64]. CRF is a discriminative and generative probabilistic model that allows nonindependent relationships among the observation sequences, thus adding flexibility to the model. CRF is modeled as an undirected acyclic graph, capturing flexibly the relation between an observation variable and hidden state. It has been applied to activity recognition tasks [65].

2.2.2 Knowledge-driven approaches

Knowledge-driven methods are motivated by real-world observations that involve activities of daily living and lists of objects required for performing such activities. In real life situations, even if the activity is performed in different ways, the number and type of objects involved do not vary significantly. For example, it is common that the "make coffee" activity consists of a sequence of actions which involves a coffee pot, a cup, hot water, coffee, sugar, and milk, whereas humans can have different habits or abilities in performing the "make coffee" activity. For instance, one may prefer strong coffee or a specific brand of coffee. For each activity, individuals may prefer different items in different orders. Such domain-dependent activity requires prior knowledge about how activities are performed in specific situations. Knowledge-based activity modeling and recognition intends to make use of rich domain knowledge and heuristics for activity modeling and pattern recognition. The knowledge structure is modeled and represented through forms such as logical axioms, rules or description logic. The most commonly used approaches are using ontologies for activity recognition. Such ontological activity models are not dependent on algorithmic choices, thus facilitating portability, interoperability, and reuse and sharing of both underlying technologies and systems. Ontology-based modeling has been applied in various AAL applications [66, 67]. However, the limitations of these

approaches are that they require complete domain knowledge to build the activity model and that they are weak in handling uncertainty and scalability.

2.2.3 Hybrid approaches

In previous works, there have been a few proposals on hybrid approaches for activity recognition systems, since both statistical and symbolic approaches have some limitations. An interesting study was performed using Markov Logic Networks (MLN), which is a probabilistic first-order logic approach [68]. With a given training set and probabilistic formulas, using MLN, weights can be learned for each formula by repetitively optimizing a pseudo-likelihood measure. These weights are the confidence value of the formula. Addition of deterministic formulas to probabilistic ones are done to express deterministic knowledge about the domain of interest. Different reasoning tasks can be implemented to deduce additional information based on formulas and facts [69]. The advantage of using probabilistic description logic is that it defines complex knowledge-based constraints, which can capture the intrinsic uncertainty of sensor measurements. Thus, by learning the weights of these constraints, the model combines positive features of knowledge-based and data-driven methods and improves the recognition rate. However, these approaches still require a labeled dataset. Various methods [70] have been proposed to derive semantic similarity between sensor events using ontologies, which is further used to segment sensor data, obtain sequential activities patterns and train the clustering model. Such semantic segmentation of sensor data helps in distinguishing the transition between activities without using labeled data. In order to access fall risk of seniors in the home environment, hybrid ontological and statistical reasoning model has been proposed [71], which could be used to reduce the number of false positives obtained from statistical false detection systems.

2.2.4 Our Contribution

In this thesis, we focused on data-driven approaches. Data-driven approaches with the advancement of deep learning achieved state-of-the-art results in vision and expanded to other domains. We present in Chapter 3.2 the concurrently first application of different deep learning models [29, 7] on single-occupant SH dataset. In Chapter 3.3, we present an application of deep learning with methods to handle imbalanced datasets on the multi-occupancy dataset. Then we also developed a privacy-enabled activity classification module for the multi-occupancy dataset.

2.3 Privacy in data management

In recent years, there has been a rapid increase in the deployment of the Internet of Things (IoT) and artificial intelligence technologies, aiming to make users' lives more convenient. These smart devices include smart wearables for health monitoring, smartphones for activity monitoring, smart sensors in the home environment for users' behavior analysis and voice-controlled appliances for users' interaction and assistance. However, these devices also collect and use personal data in order to tailor the offered services to each individual user. These data collection practices will become even more powerful in future IoT environments, given that nearly all of the IoT devices are connected to the Internet and can collectively monitor and gather personal information from users. For instance, the IoT devices may collect users' personal information without asking for their permission, or may not give notice to them when collecting potentially sensitive information which raises security and privacy issues. There is a need for new tools to provide transparency, user control, and ensure that individual privacy requirements are met. To develop these tools, it is important to better understand how people feel about the privacy implications of IoT and the situations in which they prefer to have control of their privacy. The new methods of data collection in the IoT environment have introduced new privacy challenges such as obtaining consent for data collection,

allowing users to control and choose data they would like to share while ensuring that the use of collected data is limited to the mentioned purpose [72]. Several studies have been performed to understand the privacy concerns and attitudes of users towards data control and sharing [18, 73, 74]. These studies showed that users would like to have ownership of their data and privacy concerns vary depending on factors such as age, retention time and perceived value of collected data. The findings of a user study in [18] identified privacy concerns of users in heterogeneous scenarios, which used different types of data collection and formats. Thus, it helped in determining factors (such as regulations and security mechanisms) that have the greatest impact on measures of individual privacy concerns in IoT and data sharing. In addition, such studies help in developing privacy-enabling solutions in IoT environments.

2.3.1 Anonymization Methods

Privacy enhancing technologies protect the users' privacy based on technology and can offer additional levels of protection beyond just relying on laws and policies. In order to address the privacy concerns of the users, several approaches have been proposed by the research community. These approaches include information manipulation, privacy and context awareness, access control and data anonymization. The most popular data anonymization technique is k-anonymity [75]. K-anonymity is achieved by suppressing (deleting an attribute value from the data and replacing it with a random value that matches any possible attribute value) or generalizing the attributes in the data, which means that an attribute is replaced with a less specific but semantically consistent value. Another popular anonymization technique is l-diversity. This method was developed to address the weaknesses of k-anonymity, which does not guarantee privacy against adversaries that use background knowledge or in cases where data lack diversity. For l-diversity, the anonymization conditions are satisfied if, for each group of records sharing a combination of key attributes, there are at least l - "well represented" values for each confidential attribute [76]. The disadvantage of this method is that it depends on the

range of sensitive attributes. If l-diversity is to be applied to a sensitive attribute that does not have many different values, artificial data will have to be inserted. The use of artificial data will improve privacy but may result in problems with the analysis, thus ruining the utility of the data. Also, this method is vulnerable to skewness and similarity attacks so it cannot always prevent attribute disclosure.

2.3.2 Privacy Preserving Methods

In previous work, traditional machine learning techniques have been used for privacy preservation. Among them, differential privacy [77] is the most well-known privacy-preserving method that allows the gathering of statistics from a database without revealing information about individual records. Some of its applications are principal component analysis, boosting, support vector machines, linear and logistic regression, and continuous data processing. Another privacy preserving technique is Secure multi-party computation (SMC) [78], which aims at the protection of intermediate steps of computation when multiple parties perform collaborative machine learning. The applications of SMC are linear regression, decision trees, Naive Bayes, K-means clustering, and association rules. In [79], the authors provide a method for training neural networks while preserving the privacy of the participants. However, it makes use of a two-server model for computation, in which the servers are not trusted.

Recently, deep learning methods have been used for privacy-preserving such as in [80], where authors proposed a practical system for privacy-preserving deep learning by allowing the participants to train independently on their own datasets and selectively share subsets of their models key parameters during training. An alternative method [81] calculates the privacy-loss in each model to select the parameter updates and ensure differential privacy for the stochastic gradient descent. By calculating the privacy loss for each model, they provide a solution that is applicable in cases where the whole model will be used in the target devices. The main elements of their method are a differentially private version of the SGD algorithm, the moment's accountant, and hyperparameter

tuning. A privacy-preserving version of a deep Auto-Encoders is proposed in [82], that enforces ϵ-privacy by modifying the objective function. An attack is preformed on distributed, decentralized deep learning and demonstrated that data of honest participants were leaked [83]. For the attack, generative adversarial networks (GANs) are used [84], with the assumption that there is an insider in the victim's system. The insider uses a GAN to learn to create similar objects for a particular victim's class and injects these images into the learning process under a different class. As a result, the victim has to reveal more gradients about the original class, which results in leaking information about the objects. It is shown that the scheme proposed by [80] can leak data to an honest-but-curious server and this issue can be addressed with the combination of asynchronous deep learning with additively applied homomorphic encryption [85].

An additional approach to privacy preservation in deep learning is Private Aggregation of Teacher Ensembles (PATE) [86], which partitions a sensitive training dataset in order to independently train an ensemble of ML models on each data partition. In the testing phase, the models make predictions by collectively voting for one of the possible labels. This way, in order for an input to be correctly classified, there needs to be agreement among the models that were trained independently on different data partitions. The advantage of this method is that training inputs have small impact on the overall prediction result.

Another approach is using Homomorphic Encryption [87] in combination with neural networks for data encryption. However, there are still limitations in these models, as currently it is not computationally feasible to apply Fully Homomorphic Encryption (FHE) [88] and then perform classification on the encrypted data. The existing methods use Partially Homomorphic Encryption (PHE) to encrypt data, but this type only allows additions or multiplications over data which presents a big issue for classification tasks since the use of activation functions is no longer possible and they have to be approximated with low degree polynomials. Unlike previous approaches, this project will attempt to address some of the gaps in privacy protection with a mechanism for encoding big data that is applicable on heterogeneous raw data and allows the application

of classification on encoded data without any of the records being exposed. In this work, we proposed a novel deep learning privacy preserving method for anonymization, obfuscation, and encoding of data. In this way, we offer a method that is more flexible and allows the use of activation functions in classification tasks while not allowing access to sensitive raw data.

2.3.3 Attacks on Deep Learning Models

Deep learning is often applied to domains that need high levels of security and privacy (such as trading, healthcare, SHs, activity recognition, etc.). Privacy threats in deep learning are categorized into two main categories, the training phase, and the prediction phase. In the training phase, the threats are mainly associated with the structure and deployment of the deep learning model. When learning takes place in a centralized server it puts all of the user data at risk, in case of a successful attack. But even in collaborative learning, during the training phase, a malicious user can deceive the honest users to share their private information. On the other hand, there is the prediction phase where the authors of [89] identify three main attacks, membership inference, training data extraction and model extraction. Usually, deep learning is deployed with the assumption that the training and test distributions are identical, but this is not always true as adversaries may attempt to find attack variants that evade intrusion detection, manipulate the inputs to cause miss-classification or reveal healthcare models to retrieve sensitive data [90]. This means that an attack could target the model by providing it with inputs that have been manipulated during the training, or by asking the model to make predictions on inputs that it has not been trained on. Some of the most well-known attacks of this type include model inversion [91], data poisoning [92], and model evasion [93].

2.3.4 Our Contribution

In this thesis, we focused on two parts for privacy preserving in SH setting. One is how to develop classifiers to work on encoded data instead of raw user data and new method on how user data can be stored, transmitted and transformed. We developed a simple encoder-decoder based approach that encodes user data and decodes according to privacy settings for the third-party user. In 4.2.4, we demonstrated multiple encoder-decoder approaches where user data is encoded to high-dimensional vector on the user side then transmitted to the third party where it is decoded according to privacy settings. We further improved our model in 4.2.5 by using only single encoder for each user and also demonstrated this new approach can easily be extended with additional decoder as shown in 4.2.6.

2.4 Dialogue management

Dialogue is exchanging written or verbal utterances between two or more people. Dialogue skills are acquired from a young age through interaction with other humans and humans retain this skill even in older ages [94]. In dialogue, interlocutors take turns and their output within each turn can be identified as an utterance. It is likely that a dialogue will be perceived as more natural if it has a clear segmentation of participants' utterances and turns [95].

The dialogue manager is a pipeline which is constructed by combining different modules [96]. Full conversation systems can handle spoken dialogue systems as seen in Fig. 2.1. Automatic Speech Recognizer (ASR) and Text-to-Speech Synthesizer exist in Spoken Dialogue Systems in order to convert user input to text and convert Dialogue Manager (DM) output to speech. Early ASRs were based on HMMs and later, deep learning based [97] models started to be deployed. Current ASRs are mostly based on deep learning and they are being used by end-users such as Google API, Microsoft API,

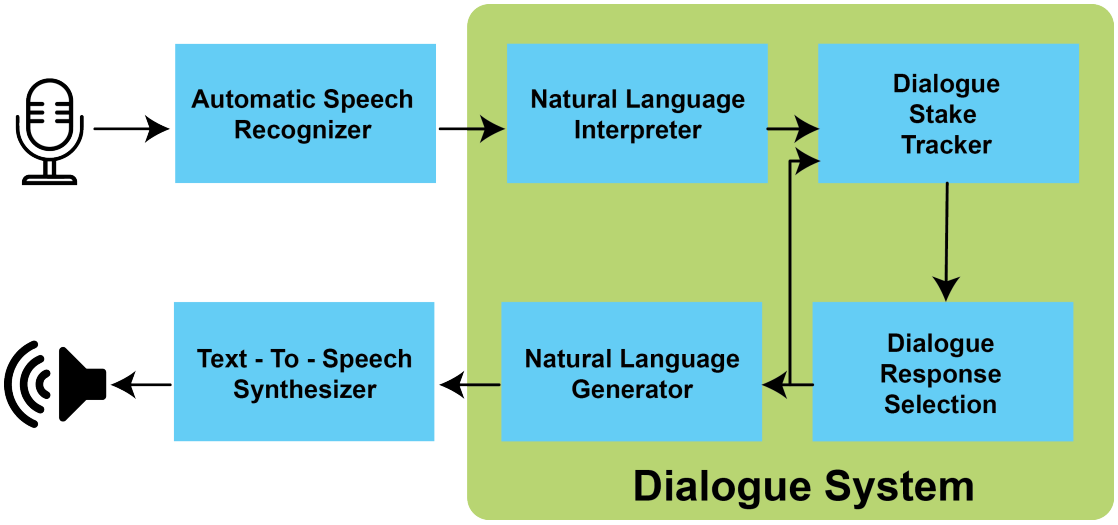


FIGURE 2.1: Dialogue Processing Pipeline.

and Sphinx. Deep learning has also improved the performance of Text-to-Speech (TTS). Tacotron 2 is a neural network which synthesizes speech directly from the text and its performance is very close to human [98]. Although ASR and TTS modules perform at near-human level, core modules are still far away from human-level performance [99].

A dialogue system consists of a Natural Language Interpreter or Natural Language Understanding module, which identifies user intents and extracts information the dialogue manager can process. Early models used support vector machine and maximum entropy models as classifiers for domain and intent classification [100, 101]. Recent deep learning based models have been deployed in natural language understanding in domain and intent classification tasks [102, 103]. Models that are based on RNN and LSTM first applied on two common tasks (addressee classification and user intent classification) by [104]. For a slot filling task, [105] used RNNs. Later, [106] extended LSTM based models for slot filling to use sentence-level information. Segmentation and slot filling were combined by [107] in neural sequence chunking models and achieved very good results. If no context is used in NLU tasks, user utterances can be ambiguous. A context sensitive model [108] was used to predict sequences of dialogue concepts. End-to-end memory networks [109] are based on memory networks [110], which are end-to-end trainable neural networks with a recurrent attention model over a possibly large external memory.

End-to-end memory networks are used in LU by integrating long-term knowledge and short-term dialogue context [111, 112]. A recent work uses time decay attention functions based on an end-to-end contextual language understanding model [113]. Errors in NLU affect the performance of the overall system. Two types of error, (i) slot-level and (ii) intent level, are investigated in task-oriented dialogue systems trained with reinforcement learning [114].

Natural Language Generation focuses on generating natural language text in order to communicate. For example in a dialogue setting, the dialogue manager may choose acts inform (day=June, airline=Turkish) and NLG converts it to a natural language utterance such as "your flight is on June with Turkish Airline". Since NLG is defined for humans, it requires human evaluation and can be very subjective and automatic metrics do not correlate with human judgements [115, 116]. Template-based NLGs map frames to natural language text. It is very simple and error-free. On the other hand, designing all the mappings requires a lot of human effort and is hard to scale. Statistical techniques [117] showed that domain expert knowledge can be used in NLG. Also, the NLG problem can be formulated as a classification problem.

Deep learning models have also been applied in NLG using RNNs. Models based on RNN learn from unaligned data and optimize sentence planning and surface realization [118]. RNN output can be used to sample from in order to create more diverse text. However, semantic repetition is observed on RNN-based NLGs, therefore [119] introduced a gating mechanism. Structural NLG encodes the syntax tree and generates natural language text [120] and later introduces context to encoded representation, which is used to generate natural language text [121]. Hierarchical NLG, proposed by [122], encodes semantics and later uses a hierarchical decoder to output natural language text by decoding different parts of outputs in each level decoder.

DM has two modules, Dialogue State Tracker (DST) and Dialogue Response Selection (DRS). The DST keeps track of the dialogue state which is needed by DRS in order to select the next response. DRS's next response is a feedback provided to DST in order to

update its current dialogue state. The response is also the input to the Natural Language Generator (NLG) module, which converts structured and template-based outputs of DRS to natural language text.

2.4.1 Conversational Agents

Dialogue managers or conversational agents can be divided into two main groups: task-oriented dialogue agents and Chatbots [123, 124]. Task-oriented dialogue agents are designed to complete a task such as restaurant reservation [125, 126], airline ticket booking [127, 99] or bus information searching [128]. In task-oriented settings, the dialogue agent interacts with humans in a limited manner to acquire information for completing the task. Task-oriented dialogue managers are designed to fill the frame which consists of slots and values (which slots can take) [129, 130]. For example in the restaurant reservation problem, the slot can be the nationality of the restaurant and values can be French, Indian, Italian [131]. Task-oriented dialogue managers are designed for specific domains and goals. Dialogue manager performance can be easily evaluated based on completion of task.

Chatbots are the models which communicate with humans using natural language [132], and often refer to the type of dialogue agents that are capable of handling unstructured dialogue without any knowledge of the dialogue structure or specific task of the dialogue. The first dialogue managers were chatbots such as Eliza [25], Parry [133] and Alice [134]. These systems are usually designed in the open domain and without a specific task. For example, Microsoft's XiaoIce [135] is a chatbot designed as an AI companion with an emotional connection to satisfy the human's social interaction needs. Chatbots are designed based on two different methodologies: (i) generative-based methods [136, 137] and (ii) retrieval based methods [138, 139, 140, 141]. Generative models generate responses given dialogue context and can generate responses that may not exist in training data. However, retrieval-based methods select from all available candidate responses, which restrict responses to a predefined set of responses. Main methodologies to train

dialogue managers are rule-based , sequence-to-sequence-based, deep reinforcement learning-based and hierarchical reinforcement learning-based methodologies.

2.4.2 Rule-based Methods

Early conversation systems Eliza [25], Parry [133], and Alice [134] were text-based and mimicked human dialogue by using an extensive set of hand-crafted rules. Having hand-crafted rules requires extensive human expertise and also makes the dialogue system constrained and not flexible. Many publicly available chatbot systems still use rule-based components in practice [96]. Rule-based systems can be also part of more complex dialogue systems as a sub-module, which creates one of the possible candidate answers among other response generator approaches [142], or to warm start dialogue system [143, 144].

2.4.3 Sequence-to-Sequence-based Methods

Sequence to Sequence methods or seq2seq are deep learning methods with encoder and decoder architecture, which transform a given sequence from one domain to another sequence [145]. For seq2seq models, usually a dictionary is defined with all the words from which the model chooses from while being conditioned on previous sequences [146]. These dictionaries can be very large depending on the complexity of dialogue.

Sequence-to-sequence methods rely on recurrent neural networks [147, 148], one-dimensional convolutional units [149, 150], or the transformer architecture [151]. Initially, they were applied on translation tasks such as English to French and achieved astonishing results with very little or no natural language processing of sentences [152, 145]. Later, they were adopted for dialogue tasks, where dialogue history is treated as a sequence and mapped to another sequence which is a dialogue response. In translation tasks, the dataset consists of pairs of sentences in different languages, whereas in the dialogue tasks, there can be many more dialogue replies for the same history.

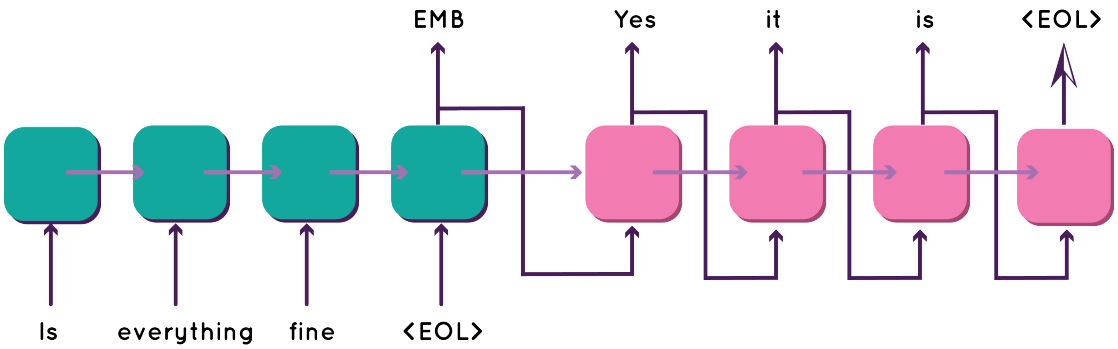


FIGURE 2.2: Sequence-to-Sequence based dialogue manager.

The decoder outputs a probability distribution over this dictionary of words based on context vector provided by the encoder. In end-to-end supervised learning models, in each time step the decoder chooses the word from set of the highest probability (or k highest) conditioned on context vector [137, 153] which consists of every word chosen before given time step and in some cases, additional information as well [154].

Fig. 2.2 demonstrates seq2seq learning applied to generation of responses for a given history. The LSTM encoder takes either full history or last reply and converts it into an encoded feature vector. The LSTM decoder takes this vector and outputs a possible reply conditioned on the encoded feature vector. Seq2seq models are the basis for recent dialogue response generators [155, 156, 157]. In Machine Translation (MT), there are pairs between languages (e.g. English-French); however, in conversational settings, each dialogue context may have many diverse responses. Another difference between dialogue and MT is the frequency of certain sentences. For example, in the dialogue setting, there are certain common sentences such as 'I don't know','Yeah' and 'Nothing' which occur in many dialogue history-response pairs. Adaptation of seq2seq models, originally developed for MT to the dialogue domain can not provide diverse answers as required in dialogue. In [158], different sources and existing approaches to low-diversity of responses of seq2seq models were explored.

When seq2seq models are trained using maximum likelihood, they tend to choose these generic responses unless sampled from output probabilities [119], this keeps their response diversity very low. Choosing a different objective such as Maximum Mutual

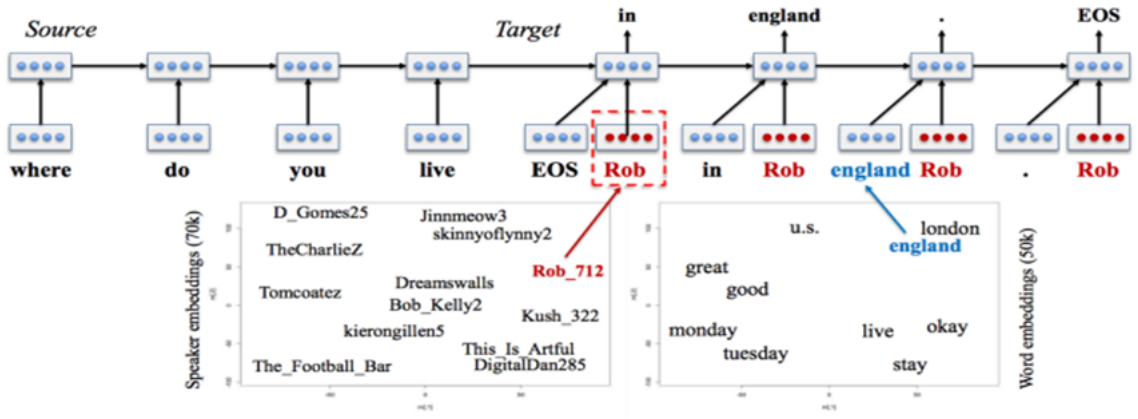


FIGURE 2.3: Persona based neural response generation [1].

Information (MMI) in response generation [159] can lower the selection frequency of such generic responses. It is also often difficult to customize unless additional information is given and may overfit in generating certain frequent replies in the dataset. A solution could be adding embeddings about user information in order to train the network to take user preferences into account [1]. In Fig. 2.3, the seq2seq model is changed to also accept embedding of the user, 'Rob' in this case, which allows the network to generate replies based on user persona.

2.4.4 Deep Reinforcement Learning-based Methods

Dialogue can be viewed as a sequential decision process and a dialogue manager needs to learn how to choose actions based on the dialogue state. Early models viewed dialog as a (fully observable) Markov Decision Process (MDP) [160, 161]. MDP assumes that dialogue action at time a_t is only dependent only on the current state s_t and s_t is fully observable, but MPDs can not handle uncertainty on state. Partially observable Markov Decision Process (POMDP) handles the uncertainty in state when dialogue state is not known specifically but is rather a distribution across all states. The POMDP view is useful especially in spoken dialogue systems, where user utterances can have acquired noise converting from speech to text. Recent advances in deep learning have also led to tremendous improvements in reinforcement learning [162]. Reinforcement learning

with powerful feature generators from deep learning has become able to scale more complex domains such as Atari [163], Go [164] and robotics [165]. Reinforcement learning framework can be deployed to learn dialogue managers where the problem is formulated as an MDP. In dialogue setting, dialogue manager is an agent which learns a strategy for a sequential decision process to maximize rewards given by the environment. An MDP is defined as a tuple $\{S, A, P, r, \gamma\}$. S is the state space, that is the dialogue history. A is the action space, which consists of available actions to the dialogue manager. These actions can be characters, words or sentences. P is the transition kernel ($P(s'|s, a)$ denotes the probability to go from s to s' under action a), $r \in \mathbb{R}^{S\times A}$ the reward function and $\gamma \in (0, 1)$ the discount factor. The dialogue manager chooses an action in each time step t and receives a reward r_t. The dialogue manager is trained to maximize the expected cumulative discounted reward.

Reinforcement learning has been applied for dialogue management for the last two decades and has shown promising results [161, 166, 167, 168, 169, 170, 142, 171]. When dialogue is modeled as an MDP, it requires a defined reward function. In the task-oriented setting, the reward can be given upon completion of task (choosing correct utterance for each dialogue turn) [125] or decoding the placeholder of the requested slot [172]. In open-domain dialogue settings, such as chatbots, it is harder to define a reward function. Heuristic rewards are developed by combining different rewards as in [170, 173]. In order to discover the reward for open-domain dialogue, GAN [84] architecture has been used. A discriminator is used to classify between real dialogue response and a response generated by the generator. The discriminators output is treated as the reward [174]. Due to the complexity of human dialogue, it is very hard to hand-craft a reward for open-domain dialogue.

Deep learning or deep reinforcement learning methods require an extensive amount of data to be trained. In [24], authors surveyed all available dialogue datasets. Datasets differ for different types of dialogue —open-domain and task-oriented. In open-domain dialogues, data are larger in amount [175, 176, 177] but since no task is defined, with diversity of human languages it is harder to train models using this large amount of data

for open domain dialogue systems. In task-oriented dialogue setting, it is often hard to collect the dataset since dataset collection should be tailored for the specific task and it should cover all scenarios that may occur for this task-specific setting. If the task-oriented dataset does not reflect well the different scenarios or users, the agent will perform not well in real task setting. Two different approaches are currently available for such data collection: Wizard-of-Oz (WOZ) [178, 179] and Machines Talking To Machines (M2M) [180]. WOZ approach does not require explicit dialogue act annotations and can acquire diverse dialogues. WOZ approach is fast and simple and requires little development in order to collect training data for task-oriented systems. However, WOZ approaches may require post-processing after data is collected and may not cover all scenarios. M2M, on the other hand, can cover more scenarios depending on the developer. It has more control over the dialogue flow. Humans only need to paraphrase computer generated approaches. M2M's shortcoming is that all system behavior is expected and planned.

2.4.5 Our Contribution

As we mentioned before, there is a need for a human annotated dataset for rating dialogue history-reply pairs. In Chapter 5.1, we present Human Annotated Movie Dialogue Dataset (HUMOD), which we collect human rating on dialogue-reply pairs align with diverse human replies for given dialogue histories. In Chapter 5.2, we present [33] a new way of processing text visually. Our work may help NLP tasks where textual data has a visual format or where dialogue management has low language variation but more choosing the right information to present user. RL-based methods are getting more popular in dialogue management. However, two main parts of RL frameworks should be improved such as dialogue reward function and more versatile and sample efficient RL frameworks. In Chapter 5.1 and chapter 5.3, we focused on dialogue reward functions for non-task-oriented and task-oriented settings. In Chapter 5.4, we presented MoSoPI which is an abstract framework for creating different configurations for RL algorithms.

We experimented with our modified version of Proximal Policy Optimization (PPO), called MoPPO, in continuous control tasks. MoPPO is a lot more sample efficient than counterparts. Versatility and sample efficiency of our RL model make our model a good framework to be used in a dialogue setting.

CHAPTER 3

Activity Recognition with Deep Learning

3.1 Motivation

Recent advances in ambient assisted technologies have resulted in a wide range of applications for a smart living for older people. Older people will be able to self-manage their health, using these new technologies. The number of older people is expected to reach about 2 billion by 2050 [181] and it is important to provide services to maintain their lifestyle in their homes.

SHs consist of many different sensors (photocell, pressure, contact, temperature, distance, etc.). These different sensors collect information that can be used to provide different services such as health care and well being to the residents. Machine learning models can be deployed to analyze a large amount of data collected by sensors for different applications. The most common applications detect Activities of Daily Living (ADL), such as cooking, sleeping, taking a shower and walking. Several models have been proposed to recognize the activities inside SHs using intrusive and non-intrusive approaches. Ethically, the non-intrusive approaches are preferable over intrusive approaches such as video-cameras in SH setting since non-intrusive approaches do not violate user privacy.

Deep learning gained popularity in recent years due to its performance on different challenging tasks such as Computer Vision and Natural Language Processing. Deep learning models are useful and powerful feature extractors for classification and regression tasks [182]. Deep learning-based feature extractors are capable of extracting features from a large amount of data, and their performances increase with the amount of available data. Sensor data collected for activity recognition is, in principle, sequential data. The most suitable architectures for sequential data processing are Long Short Term Memory (LSTM) [145], and Convolutional Neural Network (CNN) [183] applied temporally instead of spatially. We developed models based on LSTM and CNN for activity recognition in SH setting for the first time for the single-occupancy dataset [184] setting and compared performance to traditional models such as Naive Bayes, Hidden Markov Model (HMM), Hidden Semi-Markov Model (HSMM) and Conditional Random

TABLE 3.1: Details of the Kasteren datasets.

	House A	House B	House C
Age	26	28	57
Gender	Male	Male	Male
Setting	Apartment	Apartment	House
Rooms	3	2	6
Duration	25 days	14 days	19 days
Sensors	14	23	21
Activities	10	13	16
Annotation	Bluetooth	Diary	Bluetooth

Fields (CRF). In the multi-occupancy dataset [185], we compared different LSTM based architectures for activity classification performance with the emphasis on imbalanced dataset. We also showed first time for the multi-occupancy dataset [185] how over-sampling, undersampling and cost-sensitive approaches handles imbalanced data in SH setting. Both datasets are publicly available.

3.2 Activity Recognition in Single-Occupancy

In a single-occupancy SH, there is always a single user in the home and all sensor data is related to activities of the single user. This setting is very suitable for older people living alone. The Kasteren dataset [60], which we used for single-occupancy experiments, was collected from three different houses, each with a different amount of sensors installed on each house. The dataset includes many varieties between houses as listed in Table 3.1. These differences are very important to note since they show how different each SH setting can be and how hard is to train a single model that fits all different houses available in real-life. Each different configuration of the house and sensor requires a different model to be trained.

The dataset contains sensor information for every minute in each day as well as an activity label for that minute, which totals 1440 length input for each day. The data in the experiments are represented in two different forms. The first is raw sensor data, which

are the data received directly from the sensor. The second form is last-fired sensor data. The last firing sensor gives continuously 1 and changes to 0 when another sensor changes its state. For each house, we performed leave-one-out cross validation and repeated this for every day and for each house.

3.2.1 LSTM Model

The models described in this chapter were influenced by several deep neural network models including LSTM and CNN. LSTM, proposed by [147], is a recurrent neural network architecture which is capable of learning long term dependencies. LSTM has been developed in order to deal with gradient decay or gradient blow-up problems and can be seen as a deep neural network architecture when unrolled in time. The LSTM layer's main component unit is called a memory cell. A memory cell is composed of four main elements: an input gate, a neuron with self-recurrent connection, a forget gate and an output gate. The input provided to the LSTM controls the operations to be performed by the gates in the memory cell: write (input gate), read (output gate) and reset (forget gate). The following equations explain the way a layer of memory cells is updated at each timestep *t*,

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i),\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f),\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o),\\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c),\\
h_t &= o_t \tanh c_t,
\end{aligned}
$$

where W_i, W_f, W_o are the weight matrix and x_t is the input to the memory cell layer at time *t*, σ being the sigmoid and $\tanh$ is the hyperbolic tangent activation function. The terms i, f and o are the input gate, forget gate and output gate. The term c represents the memory cell and b_i, b_f, b_c, and b_o are bias vectors.

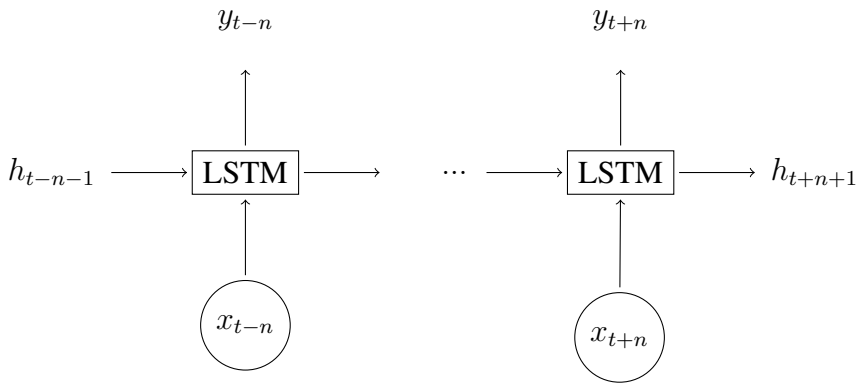


FIGURE 3.1: Illustrations of an LSTM network with x being the binary vector for sensor input and y being the activity label prediction of the LSTM network.

Fig. 3.1 illustrates an LSTM single cell layer at time t where x_t, h_t, and y_t are the input, hidden and output state.

3.2.2 CNN Model

Convolutional neural network [186] is a type of deep neural network, consisting of multiple hidden layers which can be either convolutional, pooling or fully connected. A single convolutional layer of CNN extracts features from the input signal through a convolution operation of the signal with a kernel. The activation of a unit in a CNN represents the output of the convolution of the kernel with the input signal. CNNs are able to learn hierarchical data representations for fast feature extraction and classification. The CNN model has advantages over other models when used for the activity recognition task [7] in terms of execution performance. It can capture local dependencies of the activity signal and preserve the feature scale invariant, and so is able to capture variations in similar activity efficiently through feature extraction. Fig. 3.2 shows the structure of CNN for Activity Recognition.

The 1D temporal convolutional model used in this work has four layers: 1) an input layer, 2) a convolution layer with multiple feature widths and feature map, 3) a fully connected layer and 4) the output layer.

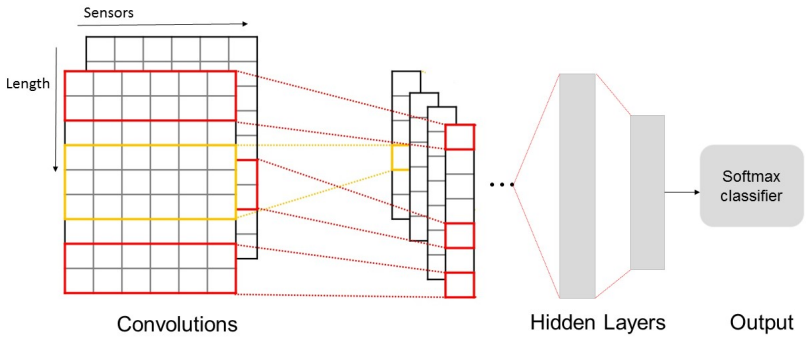


FIGURE 3.2: CNN for Activity Recognition.

3.2.3 Experiments

The results presented in Table 3.2 show the performance (accuracy) of the 1D-CNN model together with LSTM on raw sensor data and Table 3.3 shows the results of the last-fired sensor data in comparison with the results of Naive Bayes, HMM, HSMM and CRF [184]. We calculated the accuracy of the model, which represents the correctly classified activities in each timeframe. For the LSTM model, a time slice of (70) with hidden state size (300) was used. We implemented 1D (temporal) convolution with a time slice of (15). 128 filters were used for each layer and 1D kernel sizes were 5, 5, 3, 3, 3, 3 with a fully connected layer of 128 in the end. Dropout of 0.5 was used in order to reduce the overfitting in the data. We also tested longer timeslices but they tend to overfit considerably. The Adam method [187] was used with a learning rate of 0.0004 for optimization of the networks and Tensorflow library of Python has been used to implement the CNN and LSTM networks. The training took place on a Titan X GPU and the time required to train one day for one house was 4 minutes for CNN and approximately 30 minutes for LSTM, but training time differed amongst the houses. Since different houses have a different number of days of data, we calculated the average accuracy amongst all days. The training was performed using a single GPU but the trained models can be used for inference without losing performance when there is no GPU.

TABLE 3.2: Results of raw sensor data.

Model	House A	House B	House C
$1D-CNN$	88.2 ± 8.6	79.4 ± 20.1	49.2 ± 25.6
$LSTM$	$\mathbf{89.8 \pm 8.2}$	$\mathbf{85.7 \pm 14.3}$	$\mathbf{64.22 \pm 21.9}$
Naive Bayes	77.1 ± 20.8	80.4 ± 18.0	46.5 ± 22.6
HMM	59.1 ± 28.7	63.2 ± 24.7	26.5 ± 22.7
HSMM	59.5 ± 29.0	63.8 ± 24.2	31.2 ± 24.6
CRF	89.8 ± 8.5	78.0 ± 25.9	46.3 ± 25.5

TABLE 3.3: Results of last-fired sensor data.

Model	House A	House B	House C
$1D-CNN(Ours)$	95.3 ± 2.8	86.8 ± 12.7	86.23 ± 12.4
$LSTM(Ours)$	95.3 ± 2.0	88.5 ± 12.6	85.9 ± 10.6
Naive Bayes	95.3 ± 2.8	86.2 ± 13.8	87.0 ± 12.2
HMM	89.5 ± 8.4	48.4 ± 26.0	83.9 ± 13.9
HSMM	91.0 ± 7.2	67.1 ± 24.8	84.5 ± 13.2
CRF	$\mathbf{96.4 \pm 2.4}$	$\mathbf{89.2 \pm 13.9}$	$\mathbf{89.7 \pm 8.4}$

Each model for each house was trained with a leave-one-day out strategy. If the house has k days of data k-1 days were used to train and 1 day was used to test and this processed is repeated for each day. In order to compare, models average accuracy with standard deviation were calculated. Table 3.2 shows the average accuracy with the standard deviation of the accuracy of different models on raw data from three different houses. Among all the models, the LSTM performs the best for all three datasets and 1D-CNN performs second best. In House B and House C, LSTM improves the best result significantly, especially in House C where the improvement is approximately 40% from CRF and 30% from CNN. This small difference is actually very high if one thinks there can be millions of classifications done every day once SH becomes widely popular. Raw input data is preferred in real-life applications since it requires less resources (pre-processing or human expertise) to be used in applciations. Deep learning-based models perform best in raw data when compared to other models; however, it is important to notice that simpler models are also very close to more complex deep learning models.

Table 3.3 shows the accuracy on the last fired data from the three different houses. The 1D-CNN matches the best performance achieved by CRF in the case of House A but

drops slightly in case of House B and C. In comparison to LSTM, 1D-CNN performs similar except a slight decrease in case of House B. It is also important to notice the high standard deviation in all models. Standard deviation is halved for the last-fired sensor data compared to raw sensor data.

Lower standard deviation on House A can be related to having a higher number of training days and lower number of activities in House A comparing to House B and C. House C has very high standard deviation in both raw and last-fired sensor data which is reasonable since it has a higher number of activities and rooms.

3.3 Activity Recognition in Multi-Occupancy

We've shown that various deep learning based models can accurately recognize activities in single user homes - however, multi-occupancy poses a greater challenge. Many older people may live together, thus it is also crucial to develop systems for a multi-occupancy SH setting. Multi-occupancy is a harder problem than single-occupancy since sensor information gathers activities from all residents in the SH and it is harder to distinguish which sensors should be used for each user.

3.3.1 ARAS Dataset

The ARAS (Activity Recognition with Ambient Sensing) dataset contains data that were collected from two real houses with multiple residents during two months (Table 3.4). There are 2 houses in ARAS dataset each with 2 residents. Although both House A and House B use the same number of 20 sensors, they deploy different sensors. Sensor data was collected from residents over 30 days and annotated with activity labels (such as going out, having breakfast, sleeping, watching TV, having a shower, talking on the phone, brushing teeth, having guests, etc.) [185].

TABLE 3.4: Details of the ARAS Dataset.

	House A	House B
Personal Area Networks	2	1
Number of Sensors	20 of 7 different types	20 of 6 different types
Size of the House	$50m^2$	$90m^2$
House Information	one bedroom, one living room, one kitchen, one bathroom	2 bedrooms, one living room, one kitchen, one bathroom
Residents	Two 25 year old males	Married couple, 34 years old
Duration	30 days	30 days
Number of Activities	27	27

The ARAS dataset contains multi-occupancy, therefore we designed a different output layer. Instead of single output layer, we used two output layers, one for each resident, and classify into a set of possible activities. We also used different metrics since there are two resident activities to classify. For general classification (with regarding class imbalance), Exact Match Ratio (EMR) and balanced accuracy for each person were used to evaluate the models. In the EMR metric, both activities the classifier predicts for the two users should match the ground truth. Balanced accuracy is widely used in class imbalance problems and it is the average of recall obtained on each class. There are various techniques designed for class imbalance. Some of the most common ones are oversampling, undersampling and cost sensitive classification since they can be applied to any problem setting without any modification to the classifier, and can be easily applied to sequential data. Since our problem is a multi-occupant problem, we modified how we defined classes. In single-sampling approach, we define majority class and minority class for each person separately while in multi-sampling approach, we combine activities for each person and create majority and minority classes as tuples. For instance, if User-1's activity is Cooking and User-2's activity is watching TV, we define an activity tuple as Cooking, Watching TV. In both sampling methods, we defined a ratio coefficient which determines over or under sampling of minority class samples. In sampling methods (over or under), we used a threshold to not allow minority class to

TABLE 3.5: Test results on House A after 10 epochs of training. OS and US represent Oversampling and Under-sampling. Best results of EMR and Balanced accuracy are reported.

Model	**Total EMR**	**Balanced ACC1**	**Balanced ACC2**
Baseline	56.347% (+/-4.035)	35.283% (+/-2.919)	28.187% (+/-3.124)
OS Single(2)	55.157% (+/-4.144)	34.701% (+/-2.622)	29.098% (+/-5.039)
OS Multi(2)	56.196% (+/-4.135)	34.697% (+/-1.910)	29.129% (+/-4.258)
OS Single(5)	53.991% (+/-4.434)	32.871% (+/-2.251)	27.854% (+/-3.869)
OS Multi(5)	54.724% (+/-4.646)	35.175% (+/-2.837)	27.936% (+/-4.207)
US Single(0.1)	56.276% (+/-4.730)	34.416% (+/-2.041)	27.636% (+/-3.688)
US Multi(0.1)	55.933% (+/-4.185)	34.636% (+/-1.718)	28.520% (+/-4.028)
US Single(0.25)	**56.463% (+/-4.156)**	35.001% (+/-2.412)	28.782% (+/-3.089)
US Multi(0.25)	55.895% (+/-4.244)	33.964% (+/-2.631)	27.686% (+/-3.097)
CS Single(2)	55.247% (+/-3.809)	34.299% (+/-2.012)	29.772% (+/-4.386)
CS Multi(2)	56.054% (+/-4.203)	34.148% (+/-2.915)	29.058% (+/-4.432)
CS Single(5)	53.520% (+/-4.022)	**36.553**% (+/-1.689)	**30.583% (+/-3.570)**
CS Multi(5)	55.945% (+/-4.663)	35.953% (+/-2.298)	30.022% (+/-5.120)

TABLE 3.6: Test results on House B after 10 epochs of training. OS and US represent Oversampling and Under-sampling. Best results of EMR and Balanced accuracy are reported.

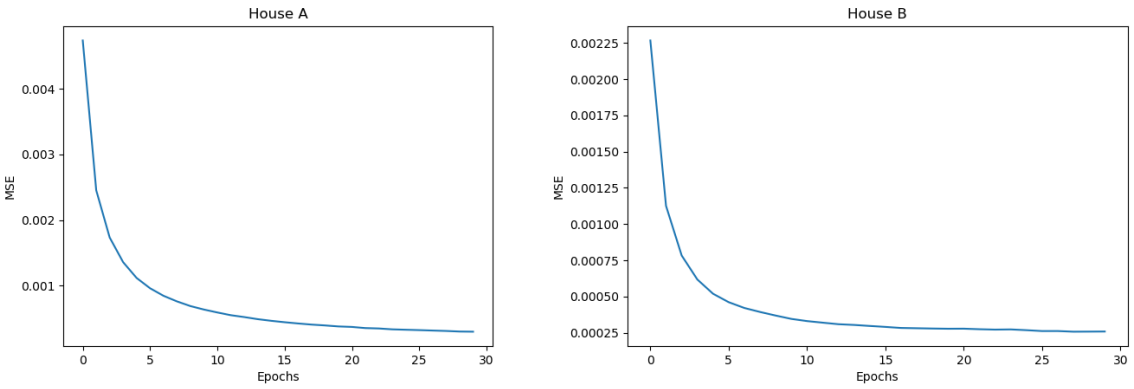
Model	**Total EMR**	**Balanced ACC1**	**Balanced ACC2**
Baseline	87.308% (+/-2.138)	34.785% (+/-4.686)	42.115% (+/-5.379)
OS Single(2)	87.128% (+/-2.451)	36.018% (+/-3.600)	41.188% (+/-4.240)
OS Multi(2)	87.116% (+/-2.015)	35.576% (+/-5.853)	41.722% (+/-5.316)
OS Single(5)	86.740% (+/-2.086)	34.717% (+/-4.122)	39.057% (+/-3.580)
OS Multi(5)	86.901% (+/-1.971)	36.890% (+/-4.490)	40.912% (+/-4.995)
US Single(0.1)	87.237% (+/-2.139)	35.124% (+/-4.369)	40.225% (+/-2.211)
US Multi(0.1)	**87.464% (+/-2.255)**	34.307% (+/-4.613)	41.727% (+/-3.176)
US Single(0.25)	87.204% (+/-1.900)	34.303% (+/-5.464)	40.172% (+/-4.607)
US Multi(0.25)	87.301% (+/-2.223)	33.692% (+/-4.669)	40.345% (+/-4.635)
CS Single(2)	87.372% (+/-1.756)	35.323% (+/-5.652)	39.487% (+/-3.380)
CS Multi(2)	87.268% (+/-2.292)	**37.046% (+/-5.363)**	41.749% (+/-4.062)
CS Single(5)	86.768% (+/-2.069)	36.020% (+/-5.870)	**42.310% (+/-4.281)**
CS Multi(5)	87.367% (+/-2.248)	35.702% (+/-5.099)	41.146% (+/-3.983)

become majority. For the cost sensitive case, cost coefficient is calculated according to majority and minority class ratio.

Single-layer LSTM was trained for 10 epochs with 5-fold cross-validation. 30 days of training data were divided into 5 different groups each with 6 days. In each run, we kept

6 days out for testing and kept the rest for training. In Tables 3.5 and 3.6, it is shown that the cost-sensitive method achieves 4 out of 6 best results while Under-sampling achieves 2 out of 6 best results. Oversampling does not perform well, which may be due to over-fitting. Moreover, it increases the dataset size which increases training time and computational cost. Cost-sensitive is better in terms that it does not remove any data such as Under-sampling while making the model more robust in the imbalanced setting. The cost-sensitive approach also offers more flexibility in terms of activity recognition in SH setting where certain activities are of high importance such as falling, turning oven, leaving the door open and eating.

Activity classification systems based on deep learning models described above require high amounts of raw user data and labels for training. User data from different houses needs to be combined to develop robust and high-performance models. It is not efficient to deploy separate training systems in each user's SH using only single user data. It is more efficient to deploy such systems outside of SH and merge different users' data. However, external training of machine learning models of user data creates privacy concerns such as how data will be transmitted, where it will be stored and how it will be processed. For example, if user sensor data is shared with third parties, all user private home information can be accessed by external parties and very sensitive information can be relieved since sensors can be read and understood (such as which sensors are on, if someone visited the person, etc.) by humans. We propose a simple encoder-decoder approach for eliminating privacy concerns while preserving the advantages of deep learning models for activity recognition. In our approach, we first trained an autoencoder to convert each human readable sensor information to encoded space and decode it back to the original feature space. This basic auto-encoder approach allows the encoding of the data, which can be easily transferred to third parties instead of real sensor information. Using autoencoding also preserves the actual information in the data while making it hard to decode it back to the original feature space since no mapping between encoded data and original input is given. In Fig. 3.3, we show the Mean Square Error between the output of decoder and original input.



(a) MSE House A (b) MSE House B

FIGURE 3.3: Mean Square Error (MSE) for House A and House B.

3.3.2 Classification on Encoded Data

It is challenging to design practical deep learning models that can be trained on encrypted data since cryptographic schemes have constraints. The main drawback of Homomorphic Encryption (HE) is the limitation on operations that exist in practical schemes; HE's operations are limited to addition or multiplication. In deep neural networks, many different activation functions are used such as Rectified Linear Unit (ReLU), Sigmoid, and hyperbolic Tangent, which needs to be approximated with functions that only use addition and multiplication so that they are compatible with HE [188].

Our methodology differs from classification on encrypted data since we can use any activation function. We demonstrated the classification of encoded data for the ARAS multi-occupancy dataset.

For this experiment, we considered a multi-label classification problem. The ground truth occupancy data was labeled for each of the two residents. We sub-sampled with 60 seconds intervals and performed one day leave-out experiments for House A and House B. For each activity classification, we used a 30 minute window for history. Fig. 3.4 shows the two networks we used for these experiments where the model on

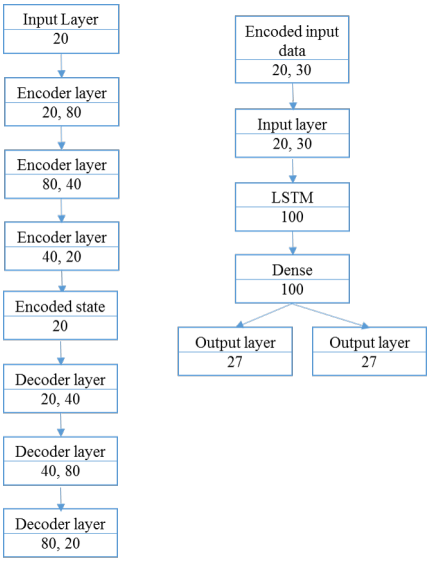


FIGURE 3.4: Classification on Encoded Data.

TABLE 3.7: Results on encoded data and raw data.

	House A		House B	
	EMR	Recall	EMR	Recall
Raw	56.3% (+/-10.0)	71.3%+/-8.4)	86.7% (+/-8.8)	90.8% (+/-6.3)
Encoded	54.9% (+/-9.6)	70.8% (+/-7.8)	86.9% (+/-8.5)	90.9% (+/-6.1)

left encodes the data, and the model on the right classifies on either the encoded or raw sensor information.Two separate output layers are used for each resident in the house.

In addition to EMR, we also used Recall, a match ratio metric in which the total number of matches is counted, so when one activity is classified correctly, it is counted as 1 and if both activities are classified correctly, it is counted as 2. Table 3.7 shows the results of House A and House B on both Raw Data and Encoded Data. We notice that the results are similar for both Raw and Encoded data. For House A, the accuracy with EMR is 56.3% for raw data and 54.9% for encoded data while for House B it is 86.7% for raw data and 86.9% for encoded data. One of the reasons for the differences in accuracy between House A and House B is that the houses are not identical; different types of sensors were used in each house to create the dataset.

3.4 Conclusion

SH systems require robust and accurate activity recognition modules in order to be deployed widely and accepted by users. Also, another important aspect of SH systems is communication with user requires understanding sensory information of SH. There are many challenges open in activity classification, of SH setting. We investigated how results may vary between different days, sensor structure, participant demographics and sensor installation architecture. Deep learning models outperform other models in raw sensory information and they are also capable of processing very large amount of data and benefit from it while other methods are hard to scale. In our published work [29, 7] we showed the benefits of applying deep learning models concurrently for the first time and compared them to classical machine learning models such as CRF and HMM, which were state-of-the-art before the deep learning models.

Classification of SH activities should use appropriate metric for the given task. For example, if fall detection is in question using traditional accuracy may not suffice since it would be a very imbalanced dataset. Class imbalance also occurs with less severity due to the nature of activities in the home such as sleeping takes longer then preparing meal which takes longer than brushing teeth. Balanced accuracy may be more feasible in these settings. Current work on multi occupant setting may use metrics which takes into account while comparing performances. However, to best to our knowledge, there is no work on how to handle imbalanced multi occupant dataset. In this thesis, we showed how to classify events in a multiple occupancy setting and how to use most common methods for class imbalance performs in multiple occupant setting. Applying different methods to vanilla deep learning models leads to classifiers that handle class imbalance dataset better.

CHAPTER 4

Privacy Preservation with Deep Learning in Smart Home

4.1 Motivation

There are many applications that can be developed for SH users using deep learning systems. For each application, using deep learning models requires a high amount of data to achieve good performance. This data requirement raises privacy concerns, especially for sharing raw user data for each application separately. User' privacy concerns would be relieved if needs for raw user data is reduced or different format of user data is used for training deep learning applications.

One privacy concern is how user data can be used to train the machine learning models. The most straightforward solution is to share user raw data and allow third parties to develop models on them. However, users may not want to share private data which is collected from their private home. Another concern is raised when user data must be readable by the external parties about how data is transferred, stored and viewed by third parties. In this chapter we analyze how deep learning models can be deployed for sharing user data while reducing these concerns.

4.2 Anonymization Model for Multiple Stakeholders

Currently, there is not one privacy definition that is able to encompass all the different aspects of privacy, but there are guidelines that are usually followed as standards, especially in regards to anonymization and personal identifiers. According to HIPAA [189] and GDPR [190], private data are those that could be used to identify a person from a group. In its guidelines, HIPAA lists the following identifiers as those that could be used to identify a person from a group [189]:

1. Names.
2. All geographical subdivisions smaller than a state.

3. Dates (other than year).
4. Phone Numbers.
5. Fax Numbers.
6. Electronic mail addresses.
7. Social Security numbers.
8. Medical record numbers.
9. Health plan beneficiary numbers.
10. Account numbers.
11. Certificate/license numbers.
12. Vehicle identifiers and serial numbers (including license plate numbers).
13. Device identifiers and serial numbers.
14. Web Uniform Resource Locators (URLs).
15. Internet Protocol (IP) address numbers.
16. Biometric identifiers, including finger, retinal and voice prints.
17. Full face photographic images and any comparable images.
18. Any other unique identifying number, characteristic or code.

However, the above list of identifiers is not exhaustive; as technology advances, more potential identifiers could emerge.

In the previous section, we discussed the different privacy concerns on user data collected in SH setting. One major concern is how much data is shared with external parties. These external parties can be:

- Caregivers who need to monitor patient health and also require access to patient information.
- Family members may already know personal data and may need restricted access to user data and also users may not be willing to share all their personal data.
- Doctors who require less personal data but need a certain amount of data to make a decision on user's health.
- Researchers who require no personal data but need access to user data in order to develop models.

Besides the external parties requirements on amount of data, the user can also select the amount of data they are willing to share.

4.2.1 Use case

Here we define more specifically a sample use case for an older person in SH setting.

John is 80 years old and lives in an ambient assisted living environment. He is widowed and was recently diagnosed with Alzheimer's disease. Currently, he lives alone but he likes to stay in touch with his family and friends. The AAL environment he lives in gives him independence and allows him to control his home automation system; for example, he can remotely open and close windows/doors, control the lighting, heating, and the alarm system. Also, it allows the monitoring of his vital signs and offers him reminders about medication and appointments. The sensors deployed in the home send the collected information to the cloud, offering access to family members, caregivers, doctors, and researchers.

In this use case scenario, four different views of the data are being created (Fig. 4.1), depending on the access level of the receiver and the preferences of the user (Table 4.1). The user in this scenario has a very close relationship with his family and trusts his

caregiver so he has selected almost all his information to be accessible to them, especially because he feels safer knowing they will be able to help him in case of emergency. With regards to doctors, he has allowed only some basic personal and medical information to be visible to them so a different view is created for them. And finally for research purposes, the view that is created does not show any explicit personal data and most of the other sensitive attributes are generalized.

4.2.2 Encoder-Decoder Model

We developed a simple encoder-decoder seq2seq model similar to a machine translation setting. Model's (LSTM) encoder maps the input sequence to a vector of fixed dimensionality, and then another LSTM is used to decode the target sequence from the vector. This can be thought of in terms of translation where task is the translate from English to French. In our case, the task is to transform user information based on privacy setting and show the transformed information to external parties. These transformations can be removing user info (replacing the address with * or just leaving the city), generalizing user info (such as converting the weight of 75 kg to 70) or categorizing certain information (such as converting numerical blood pressure to high or low categories).

The encoder network is the part of the network that takes the input sequence and maps it to an encoded representation of the sequence. The encoded representation is then used by the decoder network to generate an output sequence. This makes the framework have a lock and key analogy, where only someone with the correct key (decoder) will be able to access the resources behind the lock (encoder). The multiple hidden layers of neural networks have characteristics that enable this kind of learning [191], along with the mapping characteristic of the encoder-decoder models, which are able to create corresponding pairs that could make them appropriate for privacy-preserving frameworks.

4.2.3 Simulated Smart Home Dataset

To further illustrate our method, we used the paradigm of a smart home assisted living environment with simulated data. The kind of data that was used was divided into three categories: personal, medical and smart home sensor attributes. The personal data contain attributes like name, address, and phone number. The potentially identifiable attributes include gender and birth date. The last data type category includes sensitive medical information and sensor data such as blood pressure, medical history, presence sensors, and energy consumption. There are 10000 users in the dataset, with each user having 100 entries. For our obfuscation scheme, the data from each entry can be generalized, deleted or fully disclosed. Four separate views are created for the different stakeholders, which are family member, doctor, caregiver, and researcher. Each stakeholder has a different decoder output due to their different access levels on user information as specified in Table 4.1.

4.2.4 Multiple Encoder Multiple Decoder

We first designed multiple encoder and multiple decoder approach as shown in Fig. 4.1. Each end-user has a dedicated encoder and a decoder. Each encoder-decoder pair is trained using user data and target output data which is acquired by applying the given privacy transformations to user data.

After training, the encoder is deployed to the user side and the decoder is deployed to external parties and the only encoded information is passed between them. In case an attacker receives this information as a high-dimensional vector, he/she would still require a decoder to decode the information back. Without correct weight configuration, it is not possible to decode the data back and since data is encoded without any interpretable structure, it is very hard to decode the encoded data. This approach takes advantage of the high amount of data that is needed in order to train deep learning models and the high number of weights that are deployed in deep neural networks which makes

TABLE 4.1: Data Attributes and Access to Information.

Attribute	Family Member	Doctor	Caregiver	Researcher
Name	F	F	F	D
Age	F	F	G	G
Gender	F	F	F	F
Height	F	F	G	G
Weight	F	F	F	F
Address	F	G	F	D
Phone Number	F	F	F	D
Occupation	F	G	G	G
Marital Status	F	G	G	G
Timestamp	F	F	F	F
Blood Pressure	G	F	F	G
Glucose level	G	F	F	G
Disease	F	F	F	G
Wearable Pedometer	G	F	F	G
Presence Sensor	G	D	F	G
Temperature Sensor	G	G	F	G
Light Sensor	F	D	D	F
Window Sensor	F	D	F	D
External Door Sensor	F	D	F	D
Energy Consumption	G	D	D	G

Abbreviations F: fully disclosed, G: generalized, D: deleted

them very hard to be guessed by attackers. The mapping characteristic of the encoder-decoder models makes them suitable for privacy preservation since they are able to create corresponding pairs. This makes the method have a lock and key analogy, where only someone with the correct key (decoder) will be able to access the resources behind the lock (encoder) [30].

4.2.5 Single Encoder Multiple Decoder

Having a separate encoder for each end-user can be inefficient. Therefore, we investigated if only one encoder can be used with multiple decoders. As shown in Fig. 4.2, our model has one encoder which outputs encoded data and multiple decoders decode it back

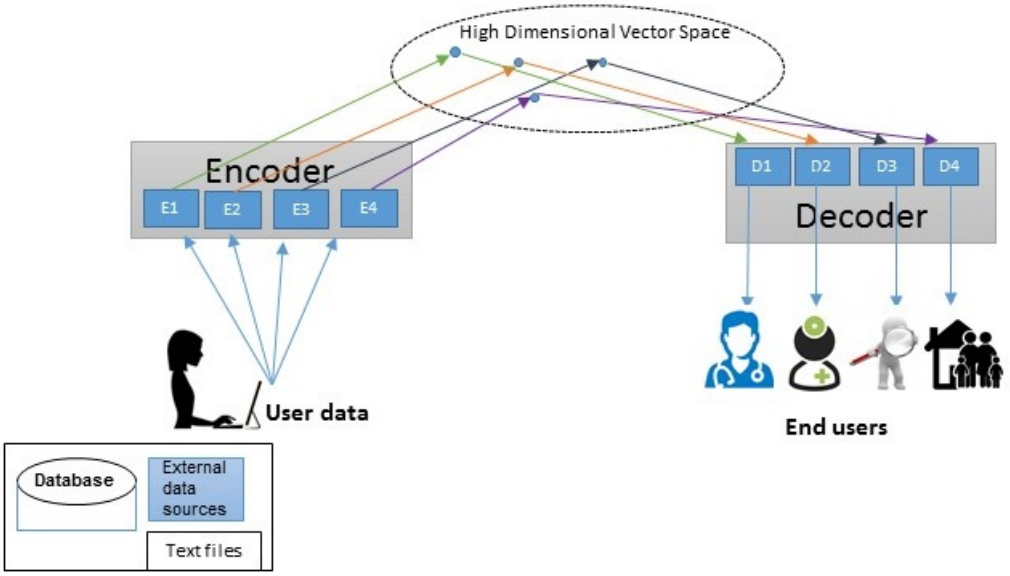


FIGURE 4.1: Conceptual System Architecture with Multiple Encoder-Decoder.

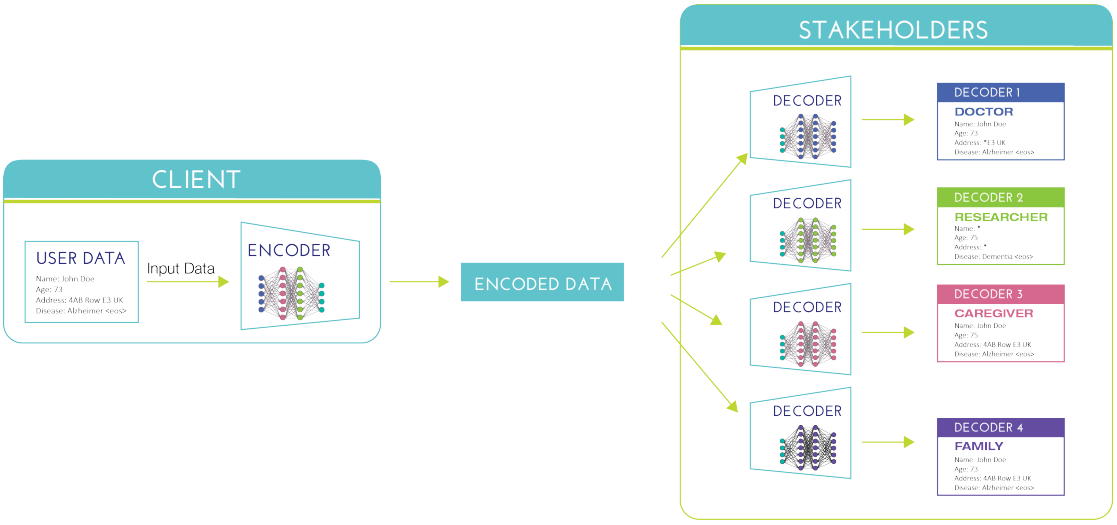
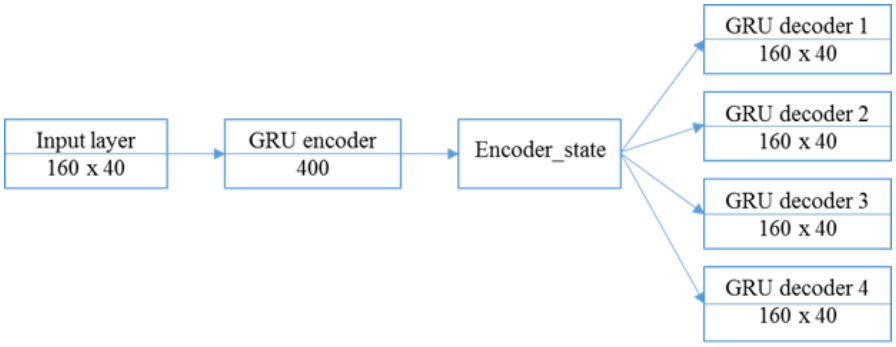
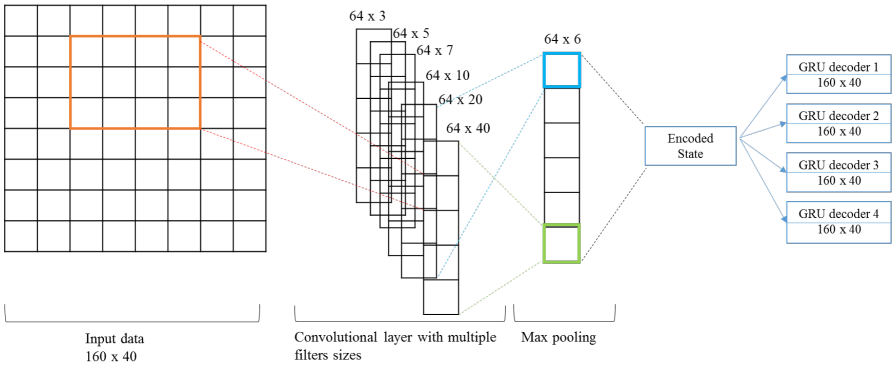


FIGURE 4.2: Encoder-Decoder based Privacy Preservation Process with Single Encoder.



(a) GRU Network Model

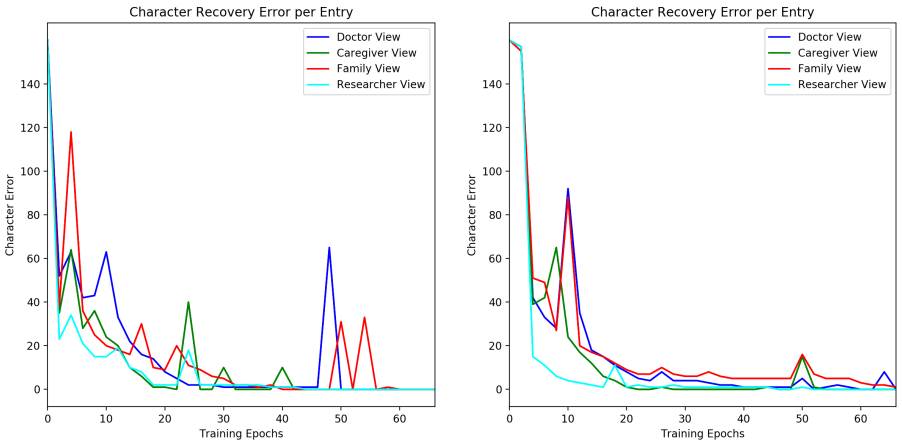


(b) 1D CNN Network Model

FIGURE 4.3: Models for One Encoder with Multiple Decoders.

according to their privacy settings. Having one encoder lowers the number of parameters to be trained significantly while allowing new decoders to be added. For the encoder, there are different deep neural networks available. In Fig. 4.3, we have two different types of encoder, one with Gated Recurrent Unit (GRU) [192] as encoder and the other model is with 1D CNN. 1D CNNs recently gained popularity due to their speed over RNN-based models. We tokenized user data into characters and used one-hot encoding for each character. We chose character level encoding since user data contains many numeric values and text which may not exist as word vectors. The decoder is kept same for both type of encoders.

Three types of operations are applied to the encoder input; keeping the data as it is, removing the data, or generalizing them (*i.e.* replacing the value with a less specific but semantically consistent value). The neural network learns which type of operation



(a) GRU (b) 1D-CNN

FIGURE 4.4: Training of the models: (a) Multiple Decoders with GRU (b) Multiple Decoders with 1D-CNN

it should perform based on the type of data and the user preference which is shown in Table 4.1. The preferences are designed and set initially and then the model is trained on those parameters. For example, the data about energy consumption are generalized for the family member and the researcher but deleted for the doctor and caregiver.

The models for the experiments were trained with 800,000 data entries and tested with 200,000 unseen entries. Each user entry has 160 characters at most. The different attributes were separated with '|' in order to easily distinguish them and the entries end with the special token 'eos'. The dictionary used has a set of 40 characters. In order to handle different sequence lengths, each entry is zero padded to the maximum number of characters which in our case is 160. To also test if the model is resilient against missing data, we inserted missing values into the simulated dataset on the attribute of Glucose Levels.

The details of the neural network for the model with one encoder and multiple decoders are shown in Fig. 4.3(a) and Fig. 4.3(b). Fig. 4.4(a) and Fig. 4.4(b) show the number of character errors in the output.

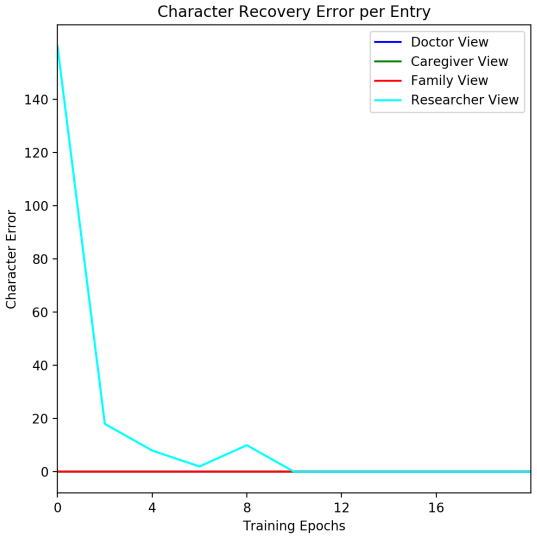


FIGURE 4.5: Addition of Decoder on Existing Decoders Set with GRU.

4.2.6 Addition of Stakeholder on the Existing Encoder-Decoders

To examine the effectiveness and practicality of the method, we tested the addition of an extra decoder to the existing encoder-decoders set to investigate how easy it would be to create another view of the data for new stakeholders requesting access to the data. In more detail, the training was done first with 3 stakeholders (Doctor, Caregiver and Family Member) and then the Researcher view was added. As can be seen in Fig. 4.5 the reconstruction of the existing views is done with minimum loss and the added Researcher's Decoder quickly reaches 0 loss, which demonstrates the easiness with which new views can be added to the existing ones. In order to guarantee that encoder encodes all the information from input data, one can add a decoder, during training the encoder, to decode all user information back perfectly.

4.3 Conclusion

We developed a simple solution that can relieve privacy concern by sharing encoded data rather than raw data with external parties, published in [30], and in this thesis,

we expanded to single encoder-multi decoder setting with experiments to flexibility of adding additional decoder. Our sequence to sequence model is trained similarly to machine translation while translating raw user data to privacy enabled data. By using this method, users can transfer their data to third parties such as doctors and caregivers but only share the amount of information required for the professional to provide their service to user. Also, user data is transferred or stored encoded, which is more resilient to attackers. In encryption, if it is cracked attacker would have human readable data. However, in encoder-decoder case, attackers would not only need to have access to encoded data but they would also need the exact configuration of decoder parameters to decode the encoded data.

CHAPTER 5

Deep Learning Models for Dialogue

In this chapter, we gathered our work that spans different areas. All parts in this chapter are related to research that may potentially improve user interaction in SHs. We start with a dataset which is collected to investigate the human perception of dialogue history and reply as well as to use diverse replies for better dialogue managers. We continue with a new approach for NLP, where the text is processed visually. It may benefit, user interaction systems where textual information is presented with a format. Deep reinforcement learning-based models are very suitable for dialogue management. If dialogue reward function is accurate with a sample efficient RL algorithm, dialogue managers would benefit very much from RL. Therefore, in the last two sections, we presented work on dialogue reward function and sample efficient RL framework.

5.1 Human Annotated Movie Dialogues Dataset (HUMOD)

5.1.1 Motivation

For open-domain unstructured dialogue manager, there are larger size datasets available than for task-oriented systems. However, these datasets are often low quality and very noisy. The description of available datasets are listed extensively in [24]. The datasets which are often used for non-task oriented dialogue managers are Ubuntu [175], Cornell [177], Twitter [176] and Open Subtitles [193]. These datasets are constructed by crawling online chats or movie subtitles. However, there is no benchmark metric to compare dialogue managers against each other. Generally, in the Ubuntu dataset, retrieval-based dialogue managers are compared against different recall metrics, or humans are used to evaluate generator-based dialogue managers.

The human evaluation of a dialogue manager is an expensive process and subjective in small amounts. Recall metrics are often considered from a small number of candidate replies; for example, the original answer is in among 10 candidate answers while the

possible candidate answers are around 100k. Lack of an automatic metric for translation (such as BLEU [194], ROUGE [195] and METEOR [196]) for dialogue history-reply makes it difficult to evaluate and compare dialogue managers in the non-task-oriented setting. Furthermore, there is no publicly available dataset with human annotations on quality of dialogue-reply pairs which is required to develop and test such metrics.

Human dialogue is also very diverse and abstract due to natural language and external knowledge bases. It is very unlikely that the diversity of human dialogue can be captured by having only a single response [197] for a given dialogue history. Existing datasets only include the original dialogue history and reply pairs which means only a single response is matched to one history unless the same history is repeated in the dataset. This diversity can be acquired through templates to some degree, but it will still be far from human generated diversity and robustness. Also, the creation of such templates for a wide range of dialogue contexts will be very challenging. Therefore, human perception of the dialogue is very important and useful in order to acquire diverse and unique dialogue replies depending on the dialogue context. These diverse replies answered by humans would also reflect individual's language, knowledge and preferences.

Movie language has been shown to be a potential source for teaching and learning human language [198] with quantitative and qualitative analyses. Movie subtitles without any dialogue segments such as the Open Subtitles dataset and with dialogue fragments (i.e. Cornell Movie Dialogue dataset) are used to train dialogue managers. However, Cornell Movie Dialogue dataset was never rated by humans in terms of how dialogue history and reply pairs fit. Another missing aspect is that there is only one correct reply for any given dialogue history. In order to enhance the Cornell dialogue dataset to overcome the challenges in human dialogue, we prepared a human annotated multi-turn movie dialogue dataset, HUMOD. We first selected a subset of the Cornell movie dialogue dataset [177] and constructed two different tasks from the selected subset of dialogues.

In the first task, we created pairs of dialogue history and replies to be rated by humans on how good they fit together using a Likert scale (between 1-5, where 1- Irrelevant,

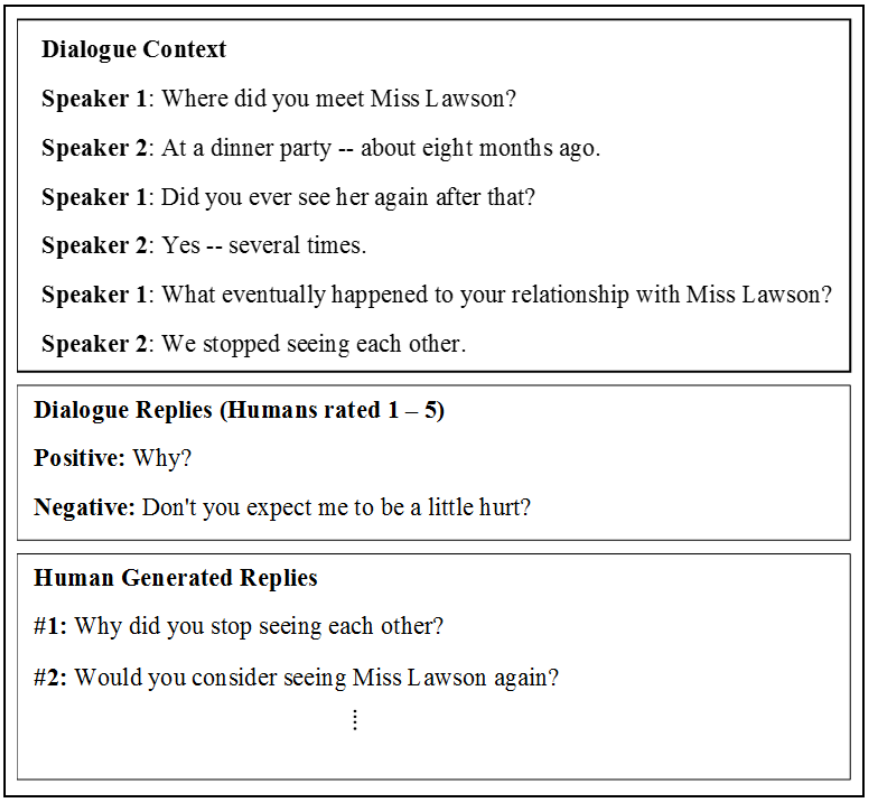


FIGURE 5.1: A sample of 6-turn dialogue context with positive (actual) reply and candidate negative (sampled) reply and two examples of human generated replies for the dialogue context.

2- Weakly irrelevant, 3-Neutral, 4- Fairly relevant and 5- Highly relevant). The second task was designed to create alternative and diverse replies for one dialogue history. We used the Amazon Mechanical Turk (AMT) platform to collect human annotation and no ethical concerns were raised. An example from the HUMOD dataset can be seen in Fig. 5.1 in a three block structure. The first block represents the dialogue context. The second block represents dialogue replies that are named positive and negative replies, similar to [199]. A positive reply is the actual reply of dialogue context from the Cornell Movie Dialogue dataset. A negative candidate reply is sampled uniformly from the set of all possible replies. In the third box, the two out of six selected human generated alternative replies to dialogue context are shown.

Next Utterance Classification (NUC) is used to train dialogue managers to discriminate positive and negative replies as in [199, 125]. Although negative replies are sampled

Dataset	# of dialogues	Human annotated	Diverse Replies	Description
Cornell Movie-Dialogue [177]	220K	No	No	Conversation from the movie scripts
Movie-DiC [200]	132K	No	No	American movie scripts
Movie-Triples [155]	245K	No	No	Dialogues of three turns between two interlocutors
OpenSubtitles [201]	36M	No	No	Movie subtitles which are not speaker-aligned
HUMOD	28.5K*	Yes	Yes	Conversation from the movie scripts with 1 to 5 ratings and six diverse replies from humans

TABLE 5.1: A comparison of existing movie dialogue datasets with HUMOD dataset. (*) denotes that HUMOD dataset can be extended by replacing the diverse replies with the original reply as explained in Fig. 5.2.

from all possible replies, it is important to have humans annotate on their goodness. Since sampled negative replies could also be good fit for the dialogue history. This also applies to positive replies. Even though, actual dialogue context and reply pair are likely to be highly related, it is important to rate the relatedness. Nonetheless, we asked a human annotator to rate both (positive and candidate negative) the dialogue replies from 1-5 (1: Irrelevant, 2: Weakly relevant, 3: Neutral, 4: Fairly relevant, 5: Highly relevant). This evaluation was highly important as it might help researchers to design dialogue evaluation score that can be compared against human judgements. As we mentioned before, humans were also asked to provide their own reply for the dialogue context and each context had six possible replies for a given dialogue context. These diverse replies can be used to train dialogue managers to create and handle diverse replies.

We also compared the HUMOD dataset with the existing movie dialogue datasets as shown in Table 5.1. One of the most important aspect of HUMOD is that it is rated by humans rather than collecting written text from movie dialogues. In actual size, the HUMOD is a smaller dataset. However, it is extendable by replacing the original positive reply with human generated alternatives, keeping in mind that it may have some noise. This extension is shown in Fig. 5.2.

Dialogue Context

{Turn 1} Speaker 1: Where did you meet Miss Lawson?

{Turn 2} Speaker 2: At a dinner party -- about eight months ago.

{Turn 3} Speaker 1: Did you ever see her again after that?

{Turn 4} Speaker 2: Yes – several times.

#Turns	Original reply from Cornell movie dataset	Human Generated Replies			
{Turn 5} Speaker 1:	What eventually happened to your relationship with Miss Lawson?	Where did you see her?	And so, what happened?	Where and when is the last time you saw Miss Lawson?	Did anything happen during those meetings?
{Turn 6} Speaker 2:	We stopped seeing each other.	She is my best friend but I am not able to contact her since a month	Nothing came of it.	After few dates, our relationship went sore when we spoke about our career.	I lost track of her.
{Turn 7} Speaker 1:	Why?	Why did you stop?	Would you consider seeing Miss Lawson again?	Don't you expect her to be a little hurt?	You guys were so good for each other.

FIGURE 5.2: The extendability approach of HUMOD dataset.

5.1.2 Dataset design and collection

The dataset consists of 4,750 multi-turn dialogue histories ranging from two to the maximum of seven turns from the Cornell movie dataset. Then, we appended each dialogue history with two replies (i) actual reply from the movie dialogue dataset and (ii) candidate negative reply sampled uniformly from possible set of replies. A total of 9500 dialogues were rated by humans. In AMT, each user was asked to complete two tasks, the first was rating the dialogue reply pair and the second was providing their own reply, for 20 dialogue history and reply pairs. We only approved AMT users who met the quality standards and removed the low quality tasks such as gibberish text, text in a

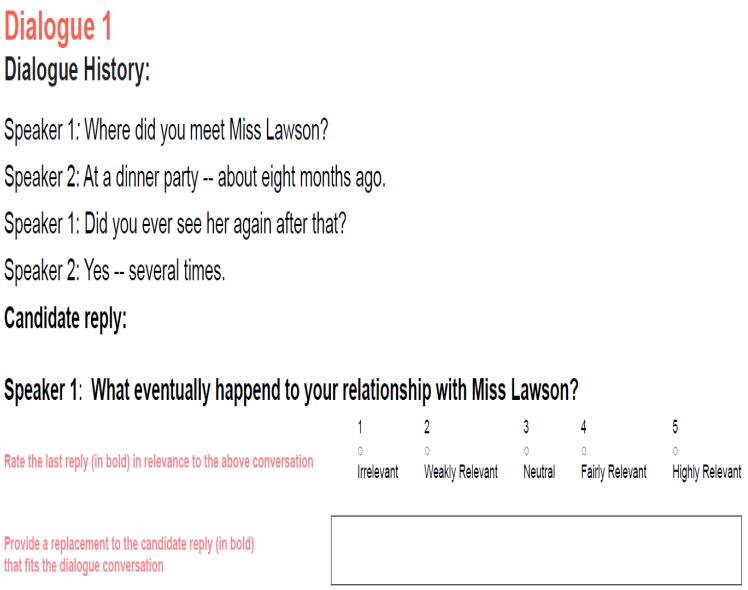


FIGURE 5.3: Screenshot of dialogue context with positive (actual) reply.

language other than English, very short and generic replies and gave the same rating to all 20 pairs. A total of 1425 AMT tasks were completed with each task containing 20 dialogue pairs. Thus, each dialogue context-reply pair was rated by three different users and each dialogue context was provided with six unique responses from humans. Screen shots from our custom website are provided in Fig. 5.3 for a positive pair and 5.4 for a negative pair.

Fig. 5.5 presents the histogram of user perception for positive and negative replies. Positive replies were generally rated high (4 or 5) by users at 74%. However, although it happened rarely, positive replies were also rated low by humans. Negative replies were rated low (1 or 2) by users at 73%, but again, some negative candidate replies were rated high by humans, which means that all sampled negative replies are not irrelevant and can actually be relevant. It is crucial to take into account the number of low scores in positive replies and high scores in negative replies which introduce noise in the dataset. There is an utmost need for human rating on dialogue context-reply pairs since positive replies may be negative and negative replies may be perceived as positive. Furthermore, in the case of noise in training, it can be harmful in a binary text classification [202] problem, and next utterance classification settings in dialogue managers can be considered as such.

FIGURE 5.4: Screenshot of dialogue context with candidate negative (sampled) reply.

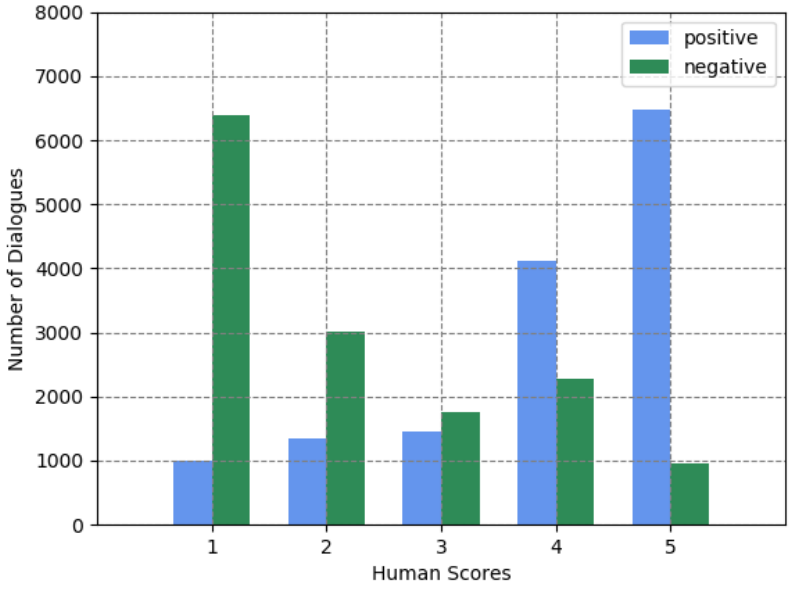


FIGURE 5.5: Human scores vs Positive and Candidate Negative Dialogue Pairs.

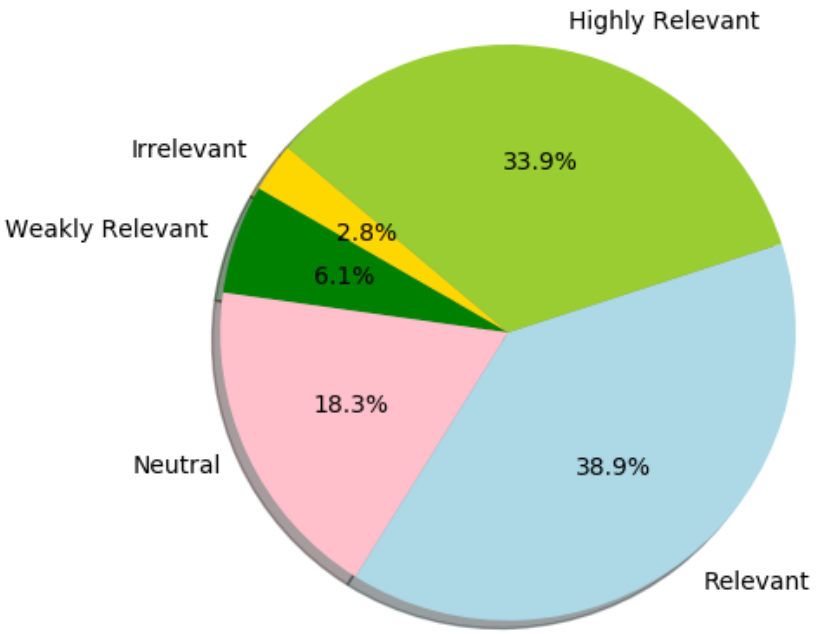


FIGURE 5.6: Evaluation of human responses on selected HUMOD dataset.

In order to evaluate human generated replies, we randomly selected 30 dialogue reply pairs with six diverse replies of different turns (for example, five dialogues for each turn i.e 2-turn, 3-turn, 4-turn, 5-turn, 6-turn, 7-turn) from the HUMOD dataset. Each diverse reply set was rated by two AMT users. Figure 5.6 presents the evaluation of human generated replies on the selected HUMOD dataset. As can be seen from the figure, 2.8% of the AMT users rated Irrelevant whereas, in original dataset the Irrelevant score was around 3.3% for the same dialogue-reply pairs.

5.1.3 Model

We compared supervised models based on RNNs and word-overlap metrics on HUMOD dataset. Word-overlap metrics are commonly used in machine-translation tasks and correlate highly with human judgment for translation tasks. However, dialogue setting is

different from translation due to the high diversity of replies for one dialogue context. Additionally dialogue context is highly relevant when evaluating replies, but word-overlap metrics do not take into account the context. Advantages of word-overlap metrics that can be used off-the-shelf without any training are that they are very easy to deploy to new datasets and require only ground truth. Another approach we developed for evaluating dialogue history-reply pairs is using a supervised model based on Hierarchical Attention Networks (HAN) [203] with a slight modification to incorporate dialogue reply as shown in Fig. 5.7. Our model can also be seen as an ADEM [204] with additional attention layers. We designed our model such that it does not require a reference reply since it is not practical to have a reference reply in real-life settings. ADEM is evaluated with and without a reference reply. We chose to use the ADEM model without the reference reply because it represents a more realistic scenario. Even though performance is higher with a reference reply, it does not take into account the multiple possible replies, and assumes only one accurate. We also experimented with Bidirectional Encoder Representations from Transformers (BERT) [205] which achieved state-of-the-art in various NLP tasks. The advantage of a supervised model is that it can be trained to evaluate as humans do in an ideal case, where there are high amount of data. However, training should be done for different datasets and if there is not enough data, the model may not generalize well. We used common translation metrics such as BLEU, METEOR, and text summarization metric ROUGE.

5.1.3.1 Hierarchical Attention Network-based Models

We implemented two separate networks, one for dialogue context and another for the dialogue reply, where we used the output of each network with different losses. The context encoder was the same as Hierarchical Attention Network (HAN) as shown in figure 5.7, which encodes dialogue context to a dialog context vector c. Reply encoder network is a biLSTM [206] network with attention. The reply encoder encodes a reply to a reply vector $\hat{r}$ (Figure 5.8).

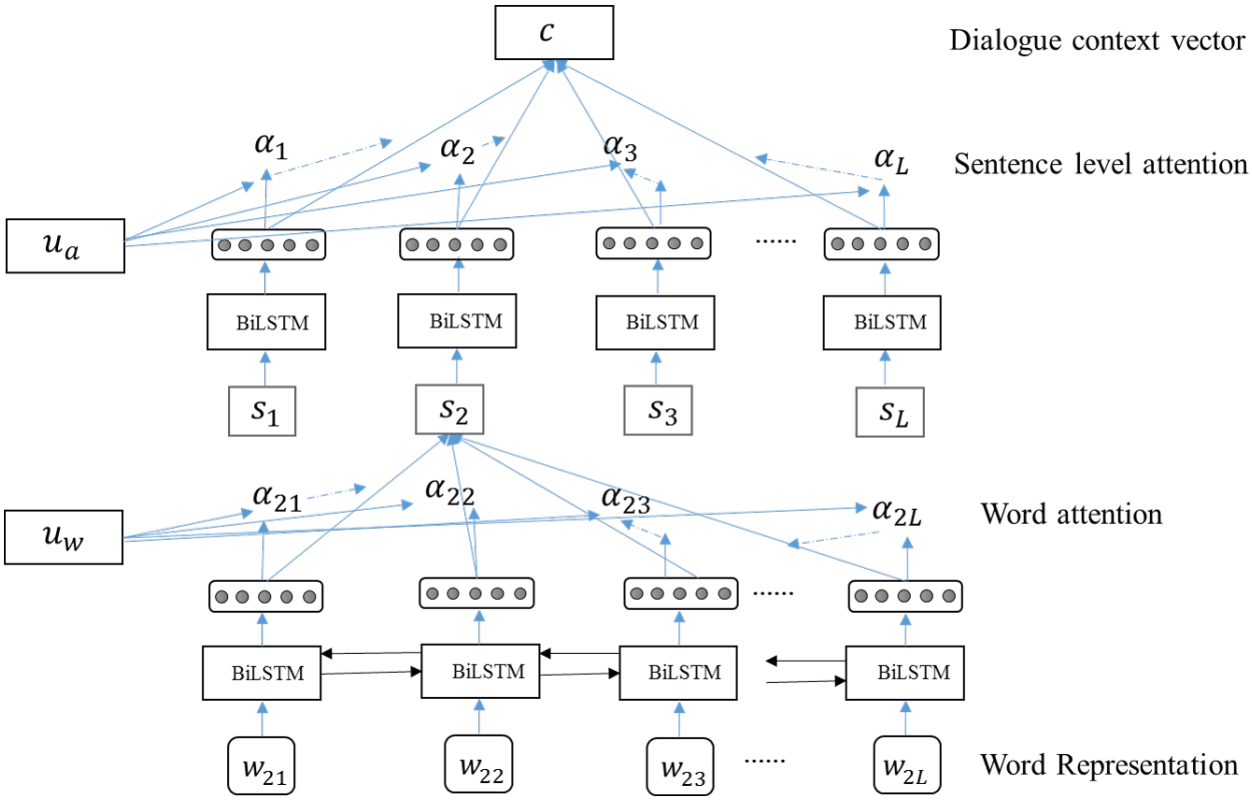


FIGURE 5.7: Hierarchical attention network for dialogue context.

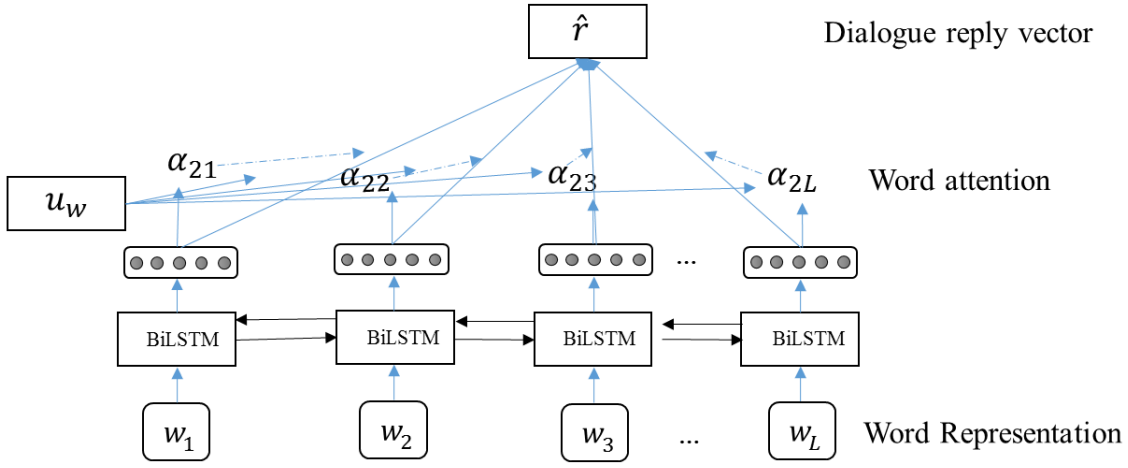


FIGURE 5.8: Dialogue reply encoder.

Two different loss functions were implemented that use dialogue context vector and reply vector. The first loss is the cross-entropy loss which classifies concatenated vector into five classes which are constructed using human ratings (1-5). The second loss, as in Eq. (5.1), was trained with mean squared error against the human score and uses concatenated vector.

$$\text{HAN-R(MSE)}(c, \hat{r}) = \sum_i [FC([c_i, \hat{r}_i]) - h_i)]^2 \tag{5.1}$$

Where h represents average human scores, c represents the context and $\hat{r}$ is reply vector. FC is the fully connected layer which outputs score for the given context and reply.

We did not use cosine similarity between dialogue context and reply since [207] showed that ADEM, which is similar to ours, creates response embedding with very low vector spread in the embedding space. In dialogue, there are many alternative replies for the same context, and the same reply can be a good fit to very different contexts. Further, common replies which occur frequently also fit a lot of different contexts. Due to these observations, when the dialogue manager is trained using cosine distance and makes embedding of context and reply similar, eventually very different texts become similar and may collapse to a very small region in embedding space.

5.1.3.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT [205] is a bidirectional language model that allows the model to learn both left and right context in all layers. BERT is pretrained with two methods that are ”Masked Language Model (MLM)” and ”Next sentence prediction” and uses a Transformer [151] with attention instead of recurrent networks like LSTM. The Next sentence prediction method is more related to our task since it trains BERT to learn the relationship between sentences, which may improve performance in the case of dialogue context and reply scoring. The BERT model has achieved state-of-the-art performance on various Natural language processing (NLP) tasks and has also outperformed human performance in a

Models	**Correlation (p-value)**
BLEU-4	0.055 (0.08)
ROUGE	-0.035 (0.26)
METEOR	-0.017 (0.59)
HAN-R(CE)	0.138 (<0.001)
HAN-R(MSE)	0.128 (0.003)
BERT	0.602 (<0.001)

TABLE 5.2: Correlation of different models with human scores.

question-answering task. Therefore, we fine-tuned BERT (from tensorflow-hub) on the HUMOD dataset in order to compare performance with other approaches.

5.1.4 Experiments

We performed a comparison of supervised and word-overlap metrics to see how they were correlated with human judgments. The dataset was divided into 8,500 context-reply pairs for the train set and 1000 context-reply pairs for the test set.

No dialogue context was shared between train and test sets. Since for each dialogue, we had scores from three judges, we took the average score of three judges. Both supervised and word-overlap approaches were evaluated on the same test set. For word-overlap metrics, we normalized average human scores into the 0-1 range and calculated metrics using NLGEval toolkit [208]. Both the context encoder and the reply encoder uses 50-dimensional glove word vectors [209] and the dimension of 100 was chosen for each biLSTM hidden state.

Results of the correlation of the HUMOD dataset with existing supervised models and word-overlap metrics is shown in Table 5.2.

We provided the benchmark results for the overall correlation of human judgment with different models. Although supervised models are correlated to some degree, it is still far from applicable to use in dialogue reply scoring as widely as translation scores are

Dialogue turn	**Correlation (p-value)**
2-turn	0.52 (<0.001)
3-turn	0.58 (<0.001)
4-turn	0.67 (<0.001)
5-turn	0.61 (<0.001)
6-turn	0.66 (<0.001)
7-turn	0.59 (<0.001)

TABLE 5.3: Correlation of BERT against different turns with human scores.

used to evaluate machine translation models both in terms of human correlation and ease of use.

As can be seen from Table 5.2, the BERT model outperformed the word-overlap metrics and HAN model, likely because BERT model takes advantage of the language model and can be fined-tuned according to other dataset. In this experiment, we used BERT with pretrained weights and fine-tuned it for HUMOD dataset, which may explain the performance difference compared to the supervised models. In addition, the correlation of BERT with human judge was performed to investigate the behavior of the network against different dialogue turns (shown in Table 5.3). 2-turn dialogue correlation was found to be lowest, which could be due to the difficulty of evaluating the dialogue reply score for very short dialogues, since they contain very little context which may increase human judgment and bias. Similarly, for long dialogues conversation it is slightly harder for system to contain the context and make good understanding of the fitness of dialogue context and reply pairs.

5.1.5 Conclusion

This section presented the Human annotated movie dialogue dataset (HUMOD), currently under review, for research to develop a benchmark metrics for comparison of different models on human scores and generated replies. The detailed description of the dataset construction and statistics was provided. Dataset will also be released for public for

further research. Different models can be compared on HUMOD for correlation with human scores, which is currently lacking. The availability of the HUMOD dataset opens up various possibilities for research in dialogue systems. To our knowledge, HUMOD is the first dataset which has alternative human generated dialogue replies for given dialogue history in open-domain dialogue setting. HUMOD also differs from the point that humans rate both actual replies for given dialogue history as well as sampled candidate reply (used as negative samples). It allows researches to investigate how humans rate dialogue history and reply pairs instead of using positive and negative (it can be harmful when negatives are not actually negative).

Different replies for the same context as well as dialogue ratings can be used to develop a metric to compare to methods such as BLEU. Another interesting use of unique diverse replies is to train generative models for dialogue systems. We presented empirical results to provide baselines with supervised and word-overlap metrics. HAN provides better results in comparison to word-overlap metrics since these approaches do not use any context. BERT outperforms HAN and provides good correlation to human dialogue score. However, it is harder to train and needs tuning for each different dataset.

5.2 Image-Based Natural Language Processing

5.2.1 Motivation

All Natural Language Processing (NLP) methods are applied on textual data, which is either acquired from original source data which is stored as text or converted to textual data from a document using Optical Character Recognition (OCR). However, most information is lost if only text is captured from the document but not the format. An example would be an Excel file which has information in a spreadsheet, which would be very hard to process only as text without any table format, even for humans. Also, most documents have information in text as well as images or visual signals accompanying

this information. Although OCR models are getting more robust, they still have a fundamental issue of only capturing text as characters but no other formatting such as paragraphs, bullet points and formatted text.

NLP models rely on embedding of natural language tokens. These tokens can be characters, n-grams or word embedding. Using different embedding has advantages and disadvantages. While character-based models reduce the dictionary size significantly they can not take advantage of language models and in certain languages such as Chinese dictionary size can be quite high. Word level tokenization can use the benefit of pretrained word vectors and use the Language Model (LM), which can be acquired training on vast amount of documents without any human annotation. These LM and word level relations can be transferred to different tasks and achieve very good results. However, human language is very rich and there can be a very high number of words so it would be hard to acquire a word embedding for each word, especially for proper nouns and numerical combinations of text (date, money amount, phone numbers, etc.).

To tackle the above problems, we use CNNs to process the entire text at once as an image. In other words, we convert our textual datasets into images of the relevant documents and apply our model on raw pixel values. This allows us to sensibly apply 2D convolutional layers on text, taking advantage of advances in neural network models designed for and targeting computer vision problems. Doing so allows us to bypass the issues stated earlier relating to the use of 1D character-level CNNs and RNNs, since now the processing of documents relies on parallel extraction of visual features of many lines (depending on filter size) of text. Regarding the vanishing gradient problem, we can take advantage of recent CNN architectural advances [210, 211, 212], which specifically aim to improve its effects. In terms of linguistics, our approach is based on the distributional hypothesis [213], where our model produces compositional hierarchies of document semantics by way of its hierarchical architecture. Beyond providing an alternative computational method to deal with the problems described above, our approach is also motivated by findings in neuroscience, cognitive science and the medical sciences where the link between visual perception, recognition of words, and semantic processing

of language has long been established [214, 215]. Our approach is robust to textual anomalies, such as spelling mistakes, unconventional use of punctuation (e.g., multiple exclamation marks), etc., which factor in during feature extraction. As a result, not only is the need for laborious text preprocessing removed, but the derived models are able to capture the semantic significance of the occurrence of such phenomena (e.g., multiple exclamation marks to denote emphasis), which proves to be especially helpful in tasks such as text classification and/or sentiment analysis. Moreover, our approach can work with any text (Latin and non-Latin), text font, misspellings and punctuation. It also reduces the need for pre-processing real-world documents (and thus the need for optical character recognition, spell check, stemming, and character encoding correction).

Processing text as image also allows models to get formatted information if it is required for the task. Such task could be getting information from an Excel table and presenting this information to user. Another task can be processing health record documents and using this information in dialogue manager to adjust the answers of manager.

In real life settings, mostly task-based dialogue managers are deployed for tasks such as restaurant booking, information retrieval or flight booking. In the Facebook bAbI dialogue dataset [125], a restaurant booking task is covered with different subtasks. Different subtasks include offering the user information about the restaurant, gathering tokens to fill slots for curating an API call or modifying the API call. All these tasks contain little human language variation and mostly focus on how to handle wide amounts of different restaurant information and how to gather information from the user to search the database. In Task 4, the knowledge base is inputted to dialogue manager as text and dialogue manager's task is to give information to the user.

Text-based models work well on this task if given restaurant information is in training data and performance drops significantly if Out Of Vocabulary (OOV) words are used. In a realistic setting, it would not be feasible to have all restaurant names or information in the training set and models would require training in case a new restaurant is added. It is also not realistic that all information is given as text but not in a more structured format

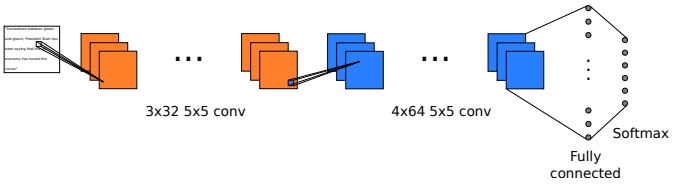


FIGURE 5.9: Proposed model: 3 convolutional layers consisting of 32 5x5 filters each, are followed by 4 convolutional layers consisting of 64 5x5 filters each. A linear fully connected layer and a classification output layer complete the model.

such as a database or Excel, and restaurant information may also include information which is not text such as a logo, map information or ratings as stars. It would be quite difficult to do the same task if information was presented as an Excel table since model would need to understand row and column relation. This row and column relation would be easier to process using visual and textual information than only processing textual information. Also, in terms of dialogue management, it could be easier to complete the task if visual information is added as well. If Task 4 was performed visually it would only require the agent to match visual patterns in KB with the answer and it is expected to work on any amount of tokens.

Adding extra information would benefit the dialogue manager training. In most tasks, dialogue manager benefits from LM and performance of the manager can be increased with additional information gathered from the vision model of document. We first did experiments to see if processing text as image was actually possible and the model could acquire the sentiment of the text. We then moved to see if a better vision model could also lead to better results on sentiment analysis of the text from the image. For Facebook tasks, we investigated the benefits of processing dialogue as image and how this can help to train dialogue managers.

5.2.2 Method

In our approach, we treated text classification as learning of context-dependent semantic conventions of language use in a given corpus of text. We treated this complex problem

as an image processing problem, where the model processes an image with the text body (Fig. 5.10), learning both the local (word- and sentence-level) and the global semantics of the corpus. In this way, the domain or context dependent meaning of sentences is implicitly contained in the variations of the visual patterns given by the distribution of words in sentences. As such, the problem is that the model needs to observe as many variations of in-domain text as possible to be able to generalize adequately. This process is similar to the analytical method of learning to read [216], where the global meaning of a body of text is acquired first and learning of the text's meaning moves to hierarchically lower linguistic units. In our case, this translates to capturing the structure and context of the whole corpus first, then the sentences, and finally the words that constitute these sentences.

5.2.2.1 Models

For the tasks of English and Chinese text classification we used a vanilla CNN and also the Xception architecture [212] to check whether better vision deep networks could increase performance.

The vanilla CNN consists of seven convolutional layers, a fully connected layer and an output layer containing as many units as classes (e.g., for a classification problem with four classes, the output layer would contain four units). All filters in the convolutional layers are 5x5 with stride 2. The first three layers use 32 filters, while the rest use 64 filters. The fully connected layer consists of 128 units. All units in all layers use the rectifier function, apart from the output layer, which uses a softmax output. Fig. 5.9 shows the architecture of the model.

For the task of dialog modeling, we used version 4 of the recently proposed deep Inception network (Inception-V4) [217]. Our choice was motivated by the fact that the vanilla CNN model was too simple to effectively model the dialog structure, as well as its pragmatics (i.e., the use of language in discourse within the context of a given domain), a problem which Inception-V4 seems to have tackled, at least to a certain extent. We

selected the Inception-V4 against the Xception because we wanted to experiment with different advanced architectures for similar tasks.

5.2.2.2 Data Augmentation

Data augmentation has been shown to be essential for training robust models [3, 218]. For image recognition, augmentation is applied using simple transformations such as shifting the width and the height of images by a certain percentage, scaling, or randomly extracting sub-windows from a sample of images [219].

For the task of English and Chinese text classification, we used the *ImageDataGenerator* function provided by Keras[220]. The input image was shifted in width and height by 20%, rotated by 15 degrees and flipped horizontally, using a batch size of 50. For the task of dialogue modeling, we applied the same augmentation techniques and random character flipping. Character flip and in particular changing the rating of a restaurant improved the per-response and per-dialog accuracy, especially for difficult sentences, such as booking a 4 star restaurant.

5.2.3 Experiments

To validate our approach, we ran experiments for two separate tasks: text classification and dialog modeling, using a single NVIDIA GTX 1080 Ti GPU.

5.2.3.1 Text classification

In this task we trained our model on an array of datasets which contained text related to news (AG's News and Sogou's News), structured ontologies on Wikipedia (DBPedia), reviews (Yelp and Amazon) and question answering (Yahoo! answers). Details about the datasets can be found in [2]. For this task, Zhang et al. [2] tested CNNs that use 1D convolutions in the task of text classification, which may more broadly include natural

TABLE 5.4: Results of Latin and Chinese text classification in terms of held-out accuracy. Worst-Best Performance reports the results of the worst and best performing baselines from Table 4 of Zhang et al. [2] and Conneau et al. [3]. Results reported for *TI-CNN* were obtained in 10 epochs.

Dataset	**Worst-best Performance (%)**	**TI-CNN (%)**	**Xception (%)**	**Number of Classes**
AG's News	83.1-92.3	80.0	91.8	4
Sogou News (Pinyin)	89.2-97.2	90.2	94.6	5
Sogou News (Chinese)	93.1-94.5	-	**98.0**	5
DBPedia	91.4-98.7	91.7	94.5	14
Yelp Review Polarity	87.3-95.7	90.3	92.8	2
Yelp Review Full	52.6-64.8	55.1	55.7	5
Yahoo! Answers	61.6-73.4	57.6	73.0	10
Amazon Review Full	44.1-63	50.2	57.9	5
Amazon Review Polarity	81.6-95.7	88.6	94.0	2

TABLE 5.5: *TI-CNN* sentiment prediction for human-generated input text. The model was trained on Amazon Review Polarity dataset.

Sample No.	**Text Sample**	**Positivity Score**
1	'this product is mediocre'	0.60
2	'this product is excelent'	0.91
3	'this product is excellent'	0.96
4	this product is excellent!!!!	0.98
5	'I love this product it is great'	0.99
6	'I like this product it is ok'	0.78
7	'I don't know'	0.56
8	'as;kdna;sdn nokorgmnsd kasdn;laknsdnaf'	0.51
9	'I recommended this product to every one in the beginning, but it turned out to be horrible later'	0.64
10	'I recommended this product to every one, in the beginning it was working great, I was in love it'	0.96

language processing, as well as sentiment analysis. While the model in [2] uses text as input vectors, our proposed method uses image data of text. In other words, whereas Zhang et al. [2] use one-hot vector representations of words or word embeddings, we use binarized pixel values of grayscale images of text corpora.

Table 5.4 shows our method's held-out accuracy in the task of Latin and Sogou News in Chinese text classification for each of the datasets. All baselines are derived from Table 4 of Zhang et al. [2] and Conneau et al. [3]. We denote the vanilla CNN by *TI-CNN*

(Text-to-Image Convolutional Neural Networks). The column *Worst-Best Performance* shows the worst and best held-out accuracy achieved by the baseline models. Our approach achieved comparable results to most of the best performing baselines. The Amazon datasets were large and we did not have enough computational resources to achieve comparable results to the state-of-the-art Xception.

Table 5.5 shows human generated text (not included in the training set) used for testing. For these examples, the table shows predictions after the model was trained on the *Amazon Review Polarity* dataset [221], which contains reviews of products in various product categories. The dataset is used for binary (positive/negative) sentiment classification of text and the metric *(positivity score)* is the probability of the positive class. The model was able to discriminate between words expressing different degrees of the same sentiment (e.g., samples 1,6 compared to samples 2-5). Sample 2 (compared to samples 3-4) illustrates our method's robustness to anomalies like spelling mistakes. In a traditional NLP setting, the misspelled word would have a different representation from the respective correctly-spelled word. Unless the model was trained on data that contained many of these anomalies, or engineered by a human, it would not necessarily correlate the misspelled word with the sentiment it expressed. In our model, the misspellings are handled naturally. We note that while this can be alleviated by preprocessing procedures or character-level models, these require more pre-processing or human intervention than our method.

As discussed before, the model builds these visual representations in a bottom-up fashion, creating a semantic hierarchy which is derived from language use within the context of the corpus domain. Sample 4 shows another interesting characteristic of our model, which is capturing the effect of punctuation (exclamation marks), even if used informally. The exclamation marks used in sample 4 generated the highest prediction score for positive sentiment among all variations of the same phrase (samples 2-4). Samples 5 and 6 have a similar structure but the different choice of words to describe positive sentiment affects the prediction score. This also exhibits the model's capacity to build meaningful hierarchical representations, as it has learned to discriminate between the

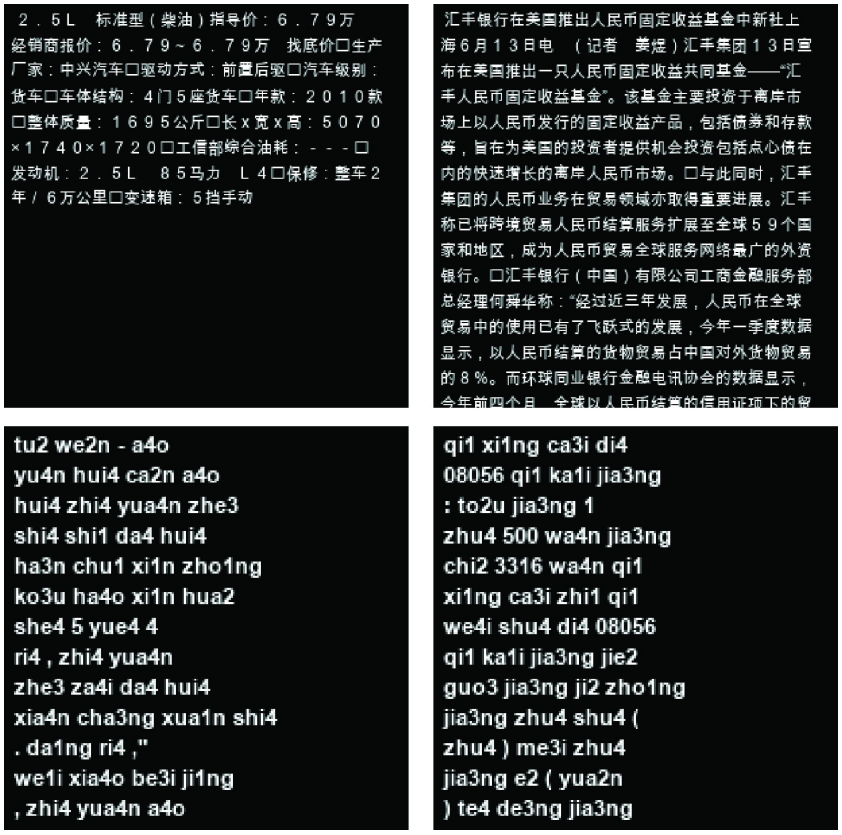


FIGURE 5.10: Top: Sogou News dataset with Chinese characters. Bottom: Sogou News dataset with pinyin.

small nuances (e.g., choice of words) encountered in (visually and semantically) similar textual structures (sentences). Interestingly, an input which expresses a "neutral" sentiment, such as sample 7, has an analogous prediction score (0.56) that is closer to random guessing in a model that was trained in binary sentiment prediction, which is reasonable behavior. The model is also robust to nonsensical text such as sample 8. Finally, we applied the Xception architecture to the Sogou News dataset, using the original Chinese characters (Fig. 5.10). Huang and Wang [222] used 1D CNNs for text classification with Chinese characters and showed that the accuracy recognition was higher than the traditional conversion to the pinyin romanization system. We extended this work by using the Xception architecture in the 2D image to achieve almost the same result (Table 5.4). This proves that regardless of how many Chinese words we fit in a 300x300 or a 200x200 image, our approach outperformed the NLP sequential CNNs. Furthermore, the performance improved when using the Chinese characters instead of

the pinyin.

5.2.3.2 Dialogue modeling

For the dialog modeling task, we tested our Inception-V4-based agent in task 4 of the bAbI dialog dataset [125], since it requires knowledge base information when choosing the replies to the user (e.g., address, phone number). The bAbI dialog dataset consists of 1000 training, 1000 validation and 1000 test dialogues in the restaurant booking domain. Each dialog is divided in four different tasks. Here we focus on task 4, where the dialog agent should be able to read entries about restaurants from a relevant knowledge base and provide the user the requested information, such as restaurant address or phone number. We note that restaurant phone numbers and addresses have been delexicalized and replaced by tokens representing this information. We chose to focus on this task to demonstrate the increased effectiveness of visual processing of dialog as opposed to purely linguistic processing, due to the high number of different lexical tokens. In our approach, the agent needs to correlate the visual pattern of a knowledge base entry to the relevant request. While in principle this should be easy to achieve using artificial delexicalized tokens, as in this benchmark task, it would be far more difficult to do so in the real world, with non-standard sequences of words (such as restaurant names, addresses etc). However, given the results of the text classification tasks, we hypothesize that given enough data, our visual approach can create semantic models that encapsulate such correlations.

As in text classification, we trained the model with images of dialog text taken from the bAbI corpus. So the agent learns the expected user utterances and their corresponding responses on the system side by processing images of in-domain dialog text. The agent learned visual representations of text meaning and structure both at word-level (implicitly, through the optimization process) and utterance-level (explicitly, through labeling of correct and incorrect responses given a user utterance).

TABLE 5.6: Facebook bAbI Dialog Task 4.

Metrics	**Inception-V4 (%)**	**Memory Networks w/o Match Type (%)**
Per-response Accuracy	63.3	59.5
Per-dialog Accuracy	11.4	3.0

Table 5.6 shows the Inception-V4 performance against the MemNets used in [125]. The table shows that our approach is competitive with MemNets when the latter does not use match types. Bordes et al. [125] introduced match types to make their model rely on type information, rather than exact match of word embeddings corresponding to words that frequently appear as containing out of vocabulary (OOV) entities (restaurant name, phone number, etc). This is because it is hard to distinguish between similar embeddings in a low-dimensional embedding space (e.g., phone numbers) as they lead to full scores in all metrics. In real life, match types would require a lexical database to identify every word type, which is not realistic.

5.2.4 Conclusion

We presented a proof of concept that natural language processing can be based on visual features of text. For non-dialog text, images of text as input to CNN models can build hierarchical semantic representations which let them detect various subtleties in language use, irrespective of the language of the input data. For dialog text, we showed that CNN models learn both the structure of discourse and the implied dialog pragmatics implicitly contained within the training data. Although our model is trained in an Next Utterance Classification (NUC) setting, it could be expanded as a generative model by using an image-based encoder for dialogue history and a language-based model for decoding. Crucially, unlike traditional NLP applications, our approach does not require any preprocessing of natural language data, such as tokenization, optical character recognition, stemming, or spell checking. Our method can easily be trained to work using different computer fonts, background colors and can be expanded to human handwriting. It is capable of performing NLP tasks on real-word documents that include

tables, bold, underlined and colored text, where traditional NLP methods, as well as language agnostic models (1D CNN) fail.

Our work [33] is a first step towards expanding the methods for natural language processing, exploiting recent advances in image recognition and computer vision versions are promising for a wide range of NLP tasks, such as text classification, sentiment analysis, dialog modeling and natural language processing. Our model achieved state-of-the-art results for Chinese which showed that our model can benefit similar languages such as Japanese.

5.3 Dialogue Reply Assessment

5.3.1 Motivation

Dialogue managers are often trained using seq2seq approaches or with reinforcement learning approaches. The issue with the sequence to sequence approach is that the manager is trained to produce the most probable answer and often generates text that is common and hard to personalize. Reinforcement learning can be deployed to train dialogue managers but it requires a reward function. In task oriented dialogue, this reward can be designed as completion of the task with some extra bonuses such as completing the task in fewer turns.

Dialogue reward is less defined for non-task oriented systems. Another drawback to applying deep reinforcement learning to dialogue manager training is sample inefficiency. Current high performing deep RL algorithms require a large amounts of data, which is a very expensive in the dialogue setting. Therefore, two very important research questions need to be investigated in order to use deep RL in order to train dialogue managers; one is how to designate a reward function, and the second is how to develop sample-efficient RL algorithms which can be deployed in real life.

It is quite challenging to define a "good" dialogue reply and what defines a "bad" dialogue reply. Defining the goodness of a dialogue reply may be easier in the task oriented setting, or choosing a right reply from available human created possible replies. It can be beneficial to develop a model which could learn intrinsic characteristics of different classes in the data. We developed a model that achieves this by explicitly learning features that define the intrinsic characteristics of a given class of data rather than features that discriminate between different classes. The aim is to distinguish between samples of a positive class (A) and samples that do not belong to this class ($\neg A$), even when test samples are not drawn from the same distribution as the training samples. We achieve this goal by introducing an energy-based model that is adversarial regarding data: it minimizes the energy for a given data distribution (the positive samples) while maximizing the energy for another given data distribution (the negative or unlabeled samples). The model is instantiated with autoencoders because of their ability to learn data manifolds.

5.3.2 Reconstruct and Crush Model

Let define p_{pos} as the probability distribution producing positive samples, $x_{\text{pos}} \sim p_{\text{pos}}$. Similarly, write p_{neg} the distribution of negative samples, $x_{\text{neg}} \sim p_{\text{neg}}$. More generally, these negative samples can be *unlabeled* samples (possibly containing positive samples). This case will be considered empirically, but we keep this notation for now.

Let N denote a neural network that takes as input a sample x and outputs a (positive) energy value E:

$$N(x) = E \in \mathbb{R}^+.$$

The proposed approach aims at learning a network N that assigns low energy values to positive samples ($N(x_{\text{pos}})$ small for $x_{\text{pos}} \sim p_{\text{pos}}$) and high energy values for negative samples ($N(x_{\text{neg}})$ high for $x_{\text{neg}} \sim p_{\text{neg}}$).

Let $m > 0$ be a user-defined margin, we propose to use the following loss $\mathcal{L}_N$ and associated risk $\mathcal{R}_N$:

$$\begin{aligned}\mathcal{L}(x_{\text{pos}}, x_{\text{neg}}; N) &= N(x_{\text{pos}}) + \max(0, m - N(x_{\text{neg}})) \\ \mathcal{R}(N) &= \mathbb{E}_{x_{\text{pos}} \sim p_{\text{pos}}, x_{\text{neg}} \sim p_{\text{neg}}} \mathcal{L}(x_{\text{pos}}, x_{\text{neg}}) \\ &= \mathbb{E}_{x_{\text{pos}} \sim p_{\text{pos}}}[N(x_{\text{pos}})] + \mathbb{E}_{x_{\text{neg}} \sim p_{\text{neg}}}[\max(0, m - N(x_{\text{neg}}))].\end{aligned}$$

Ideally, minimizing this risk amounts to having no reconstruction error over positive samples and a reconstruction error greater than m (in expectation) over negative samples. The second term of the risk acts as a regularizer that enforces the network to assign a low energy only to positive samples. The choice of the margin m will affect the behavior of the network: if m is too small, a low energy will be assigned to all inputs (both positive and negative); while if m is too large, assigning a large energy to negative samples can prevent from reconstructing the positive ones.

We specialize our model with autoencoders, which are a natural choice to represent energy-based models. An autoencoder is composed of two parts, the encoder (Enc) that projects the data into an encoding space, and the decoder (Dec) that reconstructs the data from this projection:

$$\begin{aligned}\mathrm{Enc} &: \mathcal{X} \rightarrow \mathcal{Z} \\ \mathrm{Dec} &: \mathcal{Z} \rightarrow \mathcal{X} \\ &\underset{\mathrm{Enc},\mathrm{Dec}}{\operatorname{argmin}} \|x - \mathrm{Dec}(\mathrm{Enc}(x))\|^2.\end{aligned}$$

Here, $\mathcal{X}$ is the space of the input data (either positive or negative) and $\mathcal{Z}$ is the space of encoded data. In this setting, the reconstruction error of a sample x can be interpreted as the energy value associated to that sample:

$$N(x) = \|x - \mathrm{Dec}(\mathrm{Enc}(x))\|^2 = E.$$

Our resulting reconstruct & crush network (RCN) is thus trained to assign a low reconstruction error to x_{pos} (*reconstruct*) and an high reconstruction error to x_{neg} (*crush*).

5.3.3 Experiments

In this section, we experiment with the proposed RCN on various tasks with a PU learning setting for the Amazon reviews dataset (section 5.3.3.1) and a dialogue completion setting for the Facebook bAbI dataset (section 5.3.3.2).

5.3.3.1 Amazon review

Amazon reviews is a dataset containing product reviews (ratings, text, helpfulness votes) and meta-data (descriptions, category information, price, brand, and image features) from Amazon, including 142.8 million reviews spanning [223]. Here, we only use the ratings and text features.

This set of experiments belong to the PU learning setting: the training set contains positive and unlabeled data. The positive training set contains 10k "5-star" reviews and the unlabeled training set contains 10k unlabeled reviews (containing both positive and negative reviews). The test set is composed of 10k samples: 5k "5-star" (positive) reviews and 5k "1-star" (negative) reviews. The aim here is to show that RCN performs well in the PU learning setting with unlabeled sets with different positive/negative samples ratio.

For handling the text data, we used the pretrained Glove word-embedding [224] with 100 feature dimensions. We set the maximum number of words in a sentence to 40 and zero-padded shorter sentences.

For our autoencoder, we used a 1-dimensional (1D) convolutional network defined as: (128)7c1s-(128)7c1s-(128)3c1s-(128)3c1-(128)3c1s-2048f-4000f, where "(128)7c1s" denotes a 1D convolution layer with 128 output feature maps, kernel size 7 and stride 1.

TABLE 5.7: F-measure of positive samples obtained with Roc-SVM [4], Roc-EM [5], Spy-SVM [5], NB-SVM [5], NB-EM [5] and RCN (ours). The scores are obtained on two different configuration of the unlabeled training set: one containing 5% of positive samples and one containing 50% of positive samples.

Method	F-measure for pos. samples (%5-%95)	F-measure for pos. samples (%50-%50)
Roc-SVM [4]	0.92	0.89
Roc-EM [5]	0.91	0.90
Spy-SVM [5]	0.92	0.89
NB-SVM [5]	0.92	0.86
NB-EM [5]	0.91	0.86
RCN	0.90	0.83

ReLU activation functions are used for all the layers. The margin m was set empirically to achieve highest test performance.

Table 5.7 shows the results of different well-established Positive Unlabeled (PU) learning methods, together with ours (RCN), on the Amazon review dataset. PU learning treats dataset as a combination of samples labeled as Positive and Unlabeled data which may contain positive and negative data. PU setting is more suitable for dialogue modelling since some negative training data may actually be positive. Indeed, one dialogue history may have many possible good replies as shown in HUMOD dataset. Note that, despite the fact that the architecture of our method is not specifically designed for handling the PU learning setting, it shows comparable results to the other methods, even when unlabeled training data with a considerable amount of positive samples (50%) are used.

Table 5.8 presents some examples from the test set. Note that positive comments are assigned a low reconstruction error (energy) and vice-versa.

5.3.3.2 Facebook bAbI dialogue

Facebook bAbI dialogue is a dataset containing dialogues related to 5 different tasks in which the user books a table in a restaurant with the help of a bot [225]. For each task, 1k training and 1k test dialogues are provided. Each dialogue has 4 to 11 turns between

TABLE 5.8: Examples of positive (5/5 score) and negative (1/5 score) reviews from Amazon review with the corresponding reconstruction error assigned from RCN.

Review	**Score**	**Error**
excellent funny fast reading i would recommend to all my friends	5/5	0.00054
this is easily one of my favorite books in the series i highly recommend it	5/5	0.00055
super book liked the sequence and am looking forward to a sequel keeping the s and characters would be nice	5/5	0.00060
i truly enjoyed all the action and the characters in this book the interactions between all the characters keep you drawn in to the book	5/5	0.00066
this book was the worst zombie book ever not even worth the review	1/5	1.00627
way too much sex and i am not a prude i did not finish and then deleted the book	1/5	1.00635
in reality it rates no stars it had a political agenda in my mind it was a waste my money	1/5	1.00742
fortunately this book did not cost much in time or money it was very poorly written an ok idea poorly executed and poorly developed	1/5	1.00812

the user and the bot for a total of $\sim$6k turns in each set (training and test) for task 1 and $\sim$9.5k turns in each set for task 2. Here, we consider the training and test data associated to tasks 1 and 2 because the other tasks require querying Knowledge Base (KB) upon user request; this is out of the scope of the section.

In task 1, the user requests to make a new reservation in a restaurant by defining a query that can contain from 0 to 4 required fields (cuisine type, location, number of people and price range) and the bot asks questions to fill the missing fields. In task 2, the user requests to update a reservation in a restaurant between 1 and 4 times.

The training set is built in such a way that, for each turn in a dialogue, together with the positive (correct) response, 100 possible negative responses are selected from the

candidate set (set of all bot responses in the Facebook bAbI dialogue dataset with a total of 4212 samples). The test set is built in such a way that, for each turn in a dialogue, all possible negative responses are selected from the candidate set. More precisely, for task 1, the test set contains approximately 6k positive and 25 million negative dialogue history-reply pairs, while for task 2, it contains approximately 9k positive and 38 million negative pairs.

For our autoencoder, we use a gated recurrent unit (GRU) [192] with 1024 hidden units and a projection layer on top of it in order to replicate the input sequence in output. An upper limit of 100 was set for the sequence length and a feature size of 50 was selected for word embeddings. The GRU uses ReLU activation and a dropout of 0.1. This model is implemented in Tensorflow and trained with the adam optimizer (learning rate of 0.0004) and a mini-batch size of 100 samples.

In these experiments, our network equals the state-of-the-art performance of memory networks presented in [225] by achieving 100% accuracy both for next response classification and for dialogue completion where dialogue is considered as completed if all responses within the dialogue are correctly chosen.

5.3.4 Conclusion

We have introduced a simple energy-based model, published in [34] (that provides additional experiments on vision tasks), adversarial regarding data by minimizing the energy of positive data and maximizing the energy of negative data. The model is instantiated with autoencoders where the specific architecture depends on the considered task, thus providing a family of RCNs. Such an approach can address various covariate shift problems, such as not-in-training and positive and unlabeled learning and various types of data.

The efficiency of our approach has been studied with exhaustive experiments on the Amazon reviews dataset and the Facebook bAbI dialogue dataset. These experiments

showed that RCN can obtain state-of-the art results for the dialogue completion task and competitive results for the general $A/\neg A$ classification problem. These outcomes suggest that the energy value provided by RCN can be used to asses the quality of response given the dialogue history. Future works will extend the RCN to the multi-class classification setting.

In this thesis, we also compared different models for dialogue reward assessment in task-oriented dialogue setting and showed different models that can be used for dialogue reply assessment.

5.4 Modified Actor-Critic

5.4.1 Motivation

Reinforcement learning models interact with the environment to collect data, which is used to train agent to maximize its cumulative discounted reward. In certain tasks, the environment can be fully known such as chess or Go. In these environments, it is very cheap to simulate episodes for the RL agent and simulations are perfectly accurate since all environment dynamics are known. In continuous control tasks, it is a little harder to simulate compared to game environments, but physics simulators can be used such as in Mujoco. However, in dialogue systems it is very hard to simulate episodes that have high resemblance to the real life setting. Real users differ a lot in terms of their communication and language in dialogue and it is very hard to simulate them. A common approach to collecting data is either a user simulator with highly scripted dialogue structures or to hire actual people to perform dialogues. Both of these approaches are time-consuming, and while a user simulator requires a human expert to design dialogue flows, hiring actual humans requires human participation and human experts to preprocess the data. It is clear that the real human dialogues are the best to train a dialogue manager since it is the actual environment that the agent is trained for. Regardless of how data is

obtained, it is very beneficial to have a RL algorithm which is sample efficient. Sample efficiency would benefit the models in data collection and also would perform better in more low-resource dialogue environments.

5.4.2 Positioning

Many deep reinforcement learning (RL) algorithms are based on approximate dynamic programming. Pure critic approach such as DQN [163] is based on approximate value iteration and can be only deployed to tasks with finite action spaces. However, many real life tasks require RL agents to handle discrete and continuous action spaces such as continuous control and dialogue management. Actor-critic architectures, where both the value function and the policy are represented, are a versatile approach for such tasks. Most recent approaches are either variations of policy gradient [226, 227, 228], inspired by conservative policy iteration [229, 230, 231], or make use of entropy regularization [232, 233, 234].

Approximate policy iteration has already been the building block of actor-critics in the past [235], but to our knowledge, it has not been considered with deep learning approximators. Greedy operator is unstable which may be the reason for not using approximate policy iteration with deep learning approximators. Conservative Policy Iteration (CPI) [236], addresses the instability of greedy operator, by a stochastic mixture of the current policy and of the greedy policy. This softens greediness and stabilizes learning.

Stochastic mixture of all past policies is not a practical approach as it requires to keep all past policies. Different more practical softened greediness algorithms have been developed. For example, Trust Region Policy Optimization (TRPO) [229] or Actor-Critic using Kronecker-factored Trust Region (ACKTR) [231] add a constraint on the greedy step, imposing that the average Kullback-Leibler (KL) divergence between consecutive policies is below a given threshold, and Proximal Policy Optimization (PPO) [230]

modifies the greedy step with a clipping loss that forces the ratio of action probabilities of consecutive policies to remain close to 1.

We will call generally "Soft Policy Iteration" (SPI) any approach that combines full evaluation of policy with a softened greedy step. Full evaluation makes SPI approaches naturally on-policy. Full evaluation of policy can be replaced by partial policy evaluation as in Modified Policy Iteration (MPI) [237] and in approximate setting (Approximate MPI, or AMPI [238]). Partial evaluation can be realised by using Temporal Difference (TD) learning which allows for off-policy learning.

In this work, we propose an abstract actor-critic framework that brings together MPI and SPI, by mixing the partial evaluation of MPI with the softened greediness of SPI. We name the resulting approach Modified Soft Policy Iteration (MoSoPI).

5.4.3 Background

We mentioned briefly how MDP is defined in Chapter 2.4.4. Yet, we define here once more with more details. A Markov Decision Process (MDP) is a tuple $\{S, A, P, r, \gamma\}$, with S the state space, A the action space, P the transition kernel ($P(s'|s,a)$ denotes the probability to go from s to s' under action a), $r \in \mathbb{R}^{S\times A}$ the reward function and $\gamma \in (0,1)$ the discount factor. A (stochastic) policy π is a mapping from states to distribution of actions ($\pi(a|s)$ denotes the probability of choosing a in s). The quality of a policy is quantified by the value function,

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) | s_0 = s \right],$$

where $\mathbb{E}_\pi$ denotes the expectation respectively to the trajectories sampled by the policy π and the dynamics P.

Write T_π the Bellman operator, defined for any function $v \in \mathbb{R}^S$ as

$$\forall s \in S, \quad [T_\pi v](s) = \mathbb{E}_{a\sim\pi(.|s)}[r(s,a) + \gamma v(s')].$$

The value function v_π is the unique fixed point of the operator T_π. The aim of RL is to maximize either the value function for each state or an average value function. To do so, the notion of Bellman optimality operator is useful: for any $v \in \mathbb{R}^S$, $Tv = \max_\pi T_\pi v$. The optimal value function v_* is the unique fixed point of T. The notion of greedy operator can be derived from T. We say that π is greedy respectively to $v \in \mathbb{R}^S$ (that is not necessarily a value function) if

$$\pi \in \mathcal{G}(v) \Leftrightarrow Tv = T_\pi v.$$

We consider Q-function instead of state value function, since it does not require knowing the dynamics of environment to be greedy,

$$Q_\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right].$$

Similarly to the value function, we can define the associated T_π, T and $\mathcal{G}$ operators. Value and Q-functions are linked by $v_\pi(s) = \mathbb{E}_{a\sim\pi(.|s)}[Q_\pi(s,a)]$, and the advantage function is defined as $A_\pi(s,a) = Q_\pi(s,a) - v_\pi(s)$ (it is the state-wise centered Q-function).

5.4.4 Modified Soft Policy Iteration

In this section, we present the abstract variations of policy iteration that lead to MoSoPI, as well as briefly how they can be transformed into practical algorithms.

5.4.4.1 Policy Iteration

Policy iteration (PI) alternates policy improvement and policy evaluation:

$$\begin{cases} \pi_{k+1} = \mathcal{G}(v_k) \\ v_{k+1} = v_{\pi_{k+1}} \end{cases} . \tag{5.2}$$

In the exact case, everything can be computed analytically (given finite and small enough state and action spaces), and this PI scheme will converge in finite time. In an approximate setting, one has to approximate both the value function and the policy (possibly implicitly), and to learn them from samples.

We focus on approximate case (both value and policy) since the exact case is only practical in finite and small enough state and action spaces. In approximation of policy evaluation, let Q_θ be a parameterized Q-function, Q_π can be estimated using rollouts. $\hat{\mathbb{E}}$ stands for an empirical estimation, assuming that a set of state-action couples $(s_i, a_i)_{1 \leq i \leq n}$ is available, and that we can simulate the return R_i (the cumulative discounted reward from a rollout starting in (s_i, a_i) and following the policy afterwards), then the Q-function can be estimated by minimizing

$$J(\theta) = \hat{\mathbb{E}} \left[(R_i - Q_\theta(s_i, a_i))^2 \right] .$$

If the action space is finite, the greedy policy can be deduced from the estimated Q-function $\hat{Q}_k$:

$$\pi_{k+1}(a|s) = \begin{cases} 1 & \text{if } a = \operatorname{argmax}_b \hat{Q}_k(s, b) \\ 0 & \text{else} \end{cases} .$$

Generally, one can also adopt a parameterized policy π_w and solve the greedy step as maximizing the following optimization problem:

$$J(w) = \hat{\mathbb{E}} \left[\mathbb{E}_{a \sim \pi_w(.|s_i)} [\hat{Q}_k(s_i, a)] \right] . \tag{5.3}$$

Notice that this would correspond to solving $\mathbb{E}_{s\sim\mu}[[T_{\pi_w}v](s)]$ for some distribution μ instead of the greedy step in (5.2). Adding a state-dependant baseline to $\hat{Q}_k$ does not change the minimizer, and one consider usually an estimated advantage function $\hat{A}_k$ to reduce the variance of the gradient. With discrete actions, this corresponds to a cost-sensitive multi-class classification problem [235].

5.4.4.2 Soft Policy Iteration

As we mentioned earlier CPI proposed by [236] to soften greedy greedy step is not very practical. In order to have more practical approach, [229] proposed to soften the greediness with a KL penalty between consecutive policies, that leads to minimize:

$$\hat{\mathbb{E}}\left[\mathbb{E}_{a\sim\pi_w(.|s_i)}[\hat{Q}_k(s_i,a)]\right] \text{ s.t. } \hat{\mathbb{E}}[\mathrm{KL}(\pi_w(.|s_i)||\pi_k(.|s_i))] \leq \epsilon.$$

In another approach, PPO combines the approximate greedy step 5.3 with importance sampling and a clipping of the ratio of probabilities:

$$J(w) = \hat{\mathbb{E}}\left[\mathbb{E}_{a\sim\pi_k(.|s_i)}\left[\mathrm{clip}_\epsilon\left(\frac{\pi_w(a|s_i)}{\pi_k(a|s_i)}\hat{A}_k(s_i,a)\right)\right]\right]. \tag{5.4}$$

The clip_ϵ operator saturates the ratio of probabilities when it deviates too from 1 (at $1+\epsilon$ if the advantage is positive, $1-\epsilon$ else), without it it would be equivalent to (5.3).

In this work, we call SPI any policy iteration combined with a soft greedy step, that we frame as satisfying $T_{\pi_{k+1}}v_k \geq T_{\pi_k}v_k$ (so, we ask the policy to provide some improvement, without being the greedy one). In that sense, even a policy gradient step can be seen as softened greediness.

5.4.4.3 Modified Policy Iteration

If SPI modifies the greedy step, MPI [237] modifies the evaluation step. The operator T_{π_k} being a contraction, we can write $v_{\pi_k} = (T_{\pi_k})^\infty v$, for any $v \in \mathbb{R}^S$, so notably for

$v = v_{k-1}$. MPI does partial evaluation by iterating the operator a finite number of times. Let $m \geq 1$, MPI iterates

$$\begin{cases} \pi_{k+1} = \mathcal{G}(v_k) \\ v_{k+1} = (T_{\pi_{k+1}})^m v_k \end{cases}.$$

For $m = \infty$, we retrieve PI, and for $m = 1$ we retrieve value iteration (VI): as $T_{\mathcal{G}(v)} v = Tv$, with $m = 1$ it reduces to $v_{k+1} = Tv_k$, that is VI.

We have that

$$(T_\pi)^m v = \mathbb{E}_\pi \left[\sum_{t=0}^{m-1} \gamma^t r(s_t, a_t) + \gamma^m v(s_m) | s_0 = s \right].$$

This suggests two ways of estimating a value function (or next, directly a Q-function). First, consider the case $m = 1$ and a parameterized Q-function. The classical approach consists in solving the following regression problem:

$$J(\theta) = \hat{\mathbb{E}} \left[(y_i - Q_\theta(s_i, a_i))^2 \right] \text{ with } y_i = r_i + \gamma \mathbb{E}_{a' \sim \pi_{k+1}(.|s')} [\hat{Q}_k(s', a')]. \quad (5.5)$$

With $m > 1$, a solution is to perform an m-step rollout (using π_{k+1}) and to replace y_i in Eq. (5.5) by

$$y_i^m = \sum_{t=0}^{m-1} \gamma^t r_{i+t} + \gamma^m \mathbb{E}_{a' \sim \pi_{k+1}(.|s_{t+m})} [\hat{Q}_k(s_{t+m}, a')].$$

This can be corrected for off-policy learning, using for example importance sampling or Retrace [239].

Another approach is to solve m times the regression problem of Eq. (5.5), replacing $\hat{Q}_k$ by the newly computed Q_θ after each regression but keeping the policy π_{k+1} fixed over the m regressions. In other words, solving Eq. (5.5) is one application of an approximate Bellman evaluation operator, and this amounts to applying it m times. Although using m-step returns is pretty standard in deep RL, the second approach has never been experimented in a deep RL context, to the best of our knowledge.

5.4.4.4 Modified Soft Policy Iteration

MoSoPI simply consists in bringing together a soft policy step of SPI (so any kind of soft greediness) and the partial evaluation step of MPI:

$$\begin{cases} \text{find } \pi_{k+1} \text{ s.t. } T_{\pi_{k+1}} v_k \geq T_{\pi_k} v_k \\ v_{k+1} = (T_{\pi_{k+1}})^m v_k \end{cases}.$$

To get a practical algorithm, one just has to choose a soft greedy step (eg., one of those presented in Sec. 5.4.4.2) and to estimate the partial evaluation of the Q-function, eg. with one of the approaches depicted in Sec. 5.4.4.3. We present in more detail such an instantiation in Sec. 5.4.5, that uses the greedy step of PPO and applies m times the approximate Bellman operator for evaluation.

5.4.5 Modified Proximal Policy Optimization

We instantiate our abstract framework with same greedy step as PPO (5.4) [230] and partial evaluation step depicted in Sec. 5.4.4.3. We call this algorithm Modified PPO (MoPPO). We use a replay buffer to store gathered transitions, and we evaluate the Q-function, in an off-policy manner, by solving m times the regression problem (5.5).

MoPPO and PPO differ on how advantage is calculated. Advantage function of PPO is estimated in an on-policy manner by combining successive TD errors with eligibility traces. Let I be the length of the trajectory, the advantage is estimated as

$$\delta_i = r_i + \gamma \hat{v}(s'_i) - \hat{v}(s_i).$$

$$\hat{A}(s_i, a_i) = \sum_{t=0}^{I-i+1} (\gamma\lambda)^t \delta_{i+t}.$$

Advantage function of MoPPO is estimated in off-policy manner by subtracting a Monte-Carlo empirical average of the state-action values from the estimated Q-function. If

$\hat{Q}$ has been estimated based on the policy π, we sample $a_1, \ldots, a_{N_{\text{pol}}} \sim \pi(.|s)$ and we estimate

$$\hat{A}(s,a) = \hat{Q}(s,a) - \frac{1}{N_{\text{pol}}} \sum_{j=1}^{N_{\text{pol}}} \hat{Q}(s,a_j). \tag{5.6}$$

Algorithm 1 MoPPO

Init. replay buffer D to capacity N
Init. Q function with random weights θ
Init. Q^{targ} function with random weights θ^-
Init. policy function π with random weights w
Init. policy function π_{old} with random weights w^-
Set clipping ratio ϵ
for $t = 1$ **to** max_steps **do**
  Sample action $a_t \sim \pi(s_t; \omega)$
  Execute a_t and get reward r_t and next state s_{t+1}
  Store transition (s_t, a_t, r_t, s_{t+1}) in D
  Set $s_{t+1} = s_t$
  if $t\%$train_freq $= 0$ **then**
    for $i = 1$ **to** m **do**
      for $j = 1$ **to** q_steps **do**
        Sample a minibatch from D and do a gradient step on (5.5) (with $\hat{Q}_k = Q^{\text{targ}}$ and $\pi = \pi_w$)
      end for
      Update $Q^{\text{targ}} = Q_\theta$
    end for
    for $i = 1$ **to** pol_steps **do**
      Sample a minibatch from D and do a gradient step on (5.4), using the advantage as estimated in (5.6) (with π_w unchanged, $\hat{\pi}_k = \pi_{old}$ and $\hat{Q} = Q^{\text{targ}}$)
    end for
    Update $\pi_{\text{old}} = \pi$
  end if
end for

MoPPO pseudo-code can be found in Alg. 1. MoPPO algorithm could be divided into two phase (i) interacting with the environment to collect samples to store in replay buffer (ii) training the Q and policy networks. Number of interactions depends on a hyperparameter which is a lot smaller than the size of the replay buffer. We update the Q-network m times by doing m optimizations of (5.5). For target Q-, we sample actions using the same policy π_w and we update the target network after each optimization. After training of Q network, we optimize the policy by using stochastic gradient ascent

on (5.4). Advantage function (5.6) used in policy optimization uses transitions from replay buffer which makes MoPPO off-policy.

5.4.6 Experiments

We conducted two experiments. In first experiment, we compared PPO and its modified version, using MoSoPI framework, MoPPO. PPO and MoPPO share same greedy step and only differ on how state(-action) value functions are (partially) estimated. In second experiment, we compared MoPPO to the state-of-the-art off-policy actor-critic, Soft Actor-Critic (SAC) [233] using the results from authors*. Both experiments are conducted on Mujoco tasks [240], with the OpenAI gym framework [241]. On most benchmarks, MoPPO performs better and/or faster than both PPO and SAC.

The policy was evaluated every 1000 steps by using the mean action (instead of sampling) as in [233] for the first experiment (PPO vs MoPPO). In second experiment, we followed [231] where we average the 10 best evaluation scores acquired so far, that every 1000 steps (that requires keeping track of the best past 10 policies). Results are averaged over 5 seeds†.

Fig. 5.11 present the results of two experiments. We used early stopping since the policy learned by MoPPO became too deterministic and performance degraded. On left, it can be seen that MoPPO was a lot more sample efficient and also learned competitive or better policies faster comparing to PPO (up to 5 to 10 times faster, eg. Hopper or Walker). Since MoPPO is off-policy and PPO is on-policy, it was expected from MoPPO to be a lot more sample efficient. On the right side of Fig. 5.11, MoPPO is compared to PPO and SAC by the average of the past top ten evaluation runs. MoPPO outperformed PPO as in the first experiment and performed very similar to SAC but used much less sample on most experiments.

*https://sites.google.com/corp/view/soft-actor-critic

†For SAC, we used the provided results, corresponding to five seeds, but we do not know their values and how they have been chosen. For PPO and MoPPO, we took the best 5 seeds over 8 seeds, of values evenly spaced between 1000 and 8000.

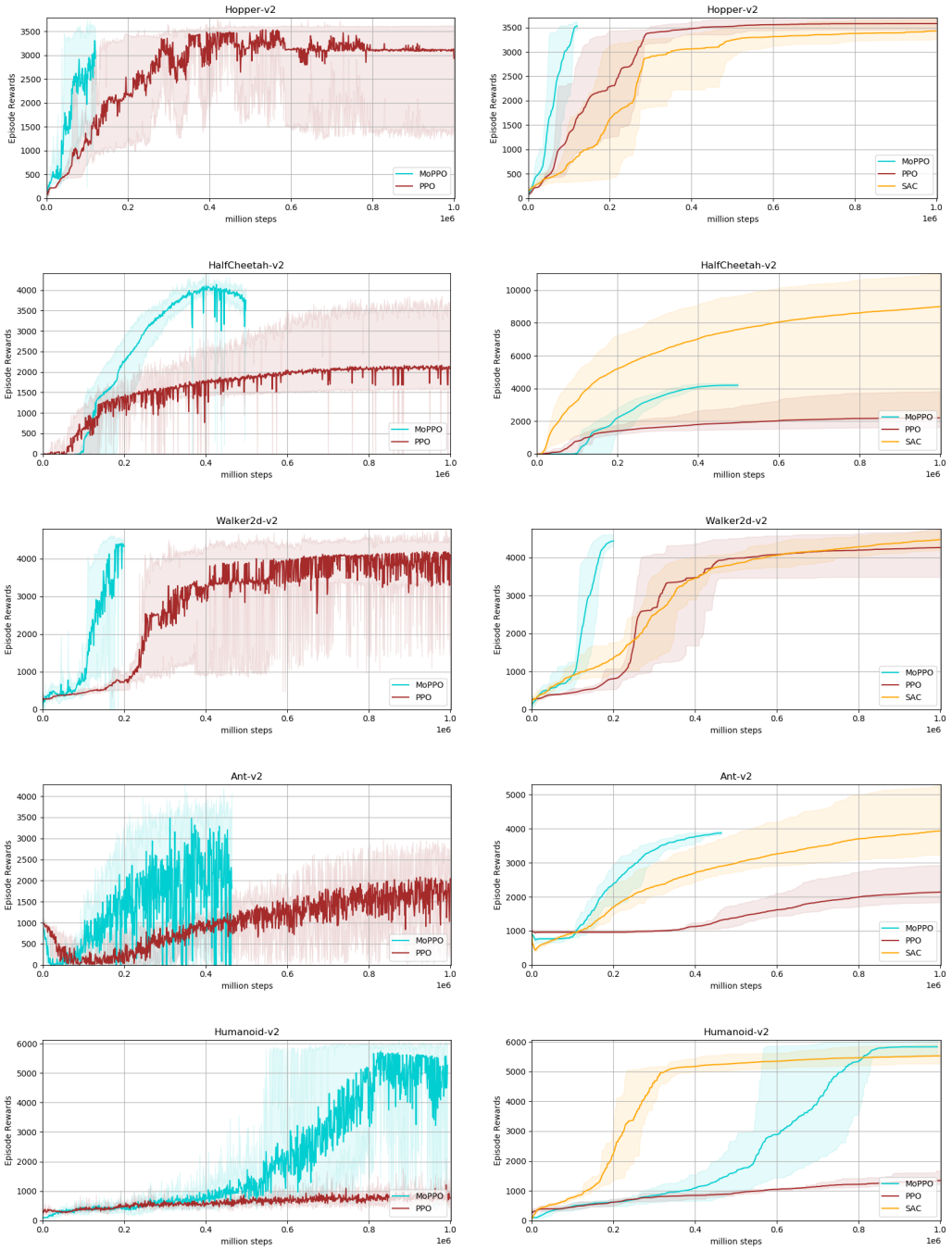


FIGURE 5.11: Evaluation results at every 1000 interaction (left) PPO vs MoPPO single evaluation result, (right) PPO vs SAC vs MoPPO the average of the past top ten evaluation runs.

5.4.7 Conclusion

In this section, we proposed MoSoPI, a general framework that mixes the general idea of soft greediness, initiated by [236], with the partial evaluation approach of MPI [237]. We instantiated MoPPO, which is a modification to PPO. MoPPO uses the same greedy step as PPO but the advantage was calculated differently, in an off-policy manner. Empirical results showed that MoPPO learns faster and better policies than PPO and achieved similar results to SAC while being more sample efficient in most tasks. MoPPO is sample efficient to learn good policy and keep best policy to deploy after training but due to instability, it is not an ideal solution for continuous learning. Since MoSoPI is an abstract framework, we can modify different parts to improve stability such as with more advanced entropy control for the policy in [242] or [234]. We also noticed in our experiments that the replay buffer also affects performance, and a smarter way to choose what experience to keep in the replay buffer [243] would help to improve stability.

We demonstrated the sample efficiency of our method on continuous control tasks on Mujoco where most actor-critic methods are compared on. Also, it is easier to simulate the physical environment and reward function compared to dialogue management. As we mentioned earlier, actor-critic methods are more adaptable as they can handle discrete and continuous action spaces. This versatility enables our MoSoPI framework to be easier to apply to dialogue management rather than Value Iteration-based methods.

CHAPTER 6

Conclusion

Developing technology is prolonging human life all around the world, especially in Europe. With the aging population, certain health problems arise such as physical injuries and mental diseases. It is not feasible to monitor and maintain the health of the older population in dedicated facilities and humans prefer to continue their life style in their homes rather than nursing homes if not medically necessary. Besides maintaining healthy lifestyles, people from all age groups would benefit from houses with intelligent systems. SH systems face many challenges to be overcome in order to be more widely applied. These challenges range from construction, installation of sensors, creating an interface, communication with user and optimal function in for maximum improvement in quality of life.

SH systems monitor and react to user activities which occur inside the home. These activities vary from user to user and house to house. Traditional machine learning models do not generalize as well as deep learning models. Deep learning models outperform traditional machine learning models and, more importantly, they get better in performance with increasing amounts of data. The amount of data available will only increase with increasing connectivity of people and sensors. Deep learning models are capable to benefit from increasing amount of data and their performance also gets better with more training data. Also, deep learning models outperform traditional machine learning models and are increasingly applied to different domains. We applied different deep learning neural networks on SH benchmark datasets with single and multi-occupant settings. Deep learning models outperformed the traditional models and required less preprocessing and feature engineering in our experiments. We also investigated how the same algorithm can deliver different results between house and user settings, which is very important for models to be applied widely. Transformers have been recently proposed and achieved state-of-art results on NLP tasks which is also processing sequential data. Therefore, as as future work it will be interesting to apply transformers on SH data and certain pretraining methods that are applied for Language Model may also benefit to be applied on Activity Model for activities in SHs.

In the multi-occupancy setting, we also showed how important metrics are and the importance of designing algorithms for class imbalance. Class imbalance naturally occurs in SHs since certain activities occur more often and for longer in the home. A model may achieve very high accuracy due to class imbalance but balanced accuracy would reveal the actual performance of the model. Likewise, in some machine learning problems, not every mistake is treated equally. This is very true in the SH setting; for example, if the system makes a mistake on detecting a user fall, it is much more harmful than making a mistake on detecting if a user is brushing his teeth. Training with equal importance for each activity in the home environment is not suitable to provide high user experience and satisfaction. In the multi-occupancy setting, common methods to help with class imbalanced dataset, needs to be altered since instead of one classification there are multi classifications (for each user). We implemented different ways of oversampling and undersampling. We first treated each user separately for minor and major class over and undersampling. Later we combined User One and User Two activities together and applied sampling methods for combined activities being major or minor class. Besides sampling, weighted classification is also a common solution that can be used in class imbalance without changing model or data distribution. It is also suitable in the SH setting where certain activities are of more importance. Using cost as weights in classification loss improves performance in balanced accuracy compared to the other two sampling methods. We will also investigate more advanced methods that can work on feature level for handling class imbalance and can be applied on sequential data in the future work.

SH systems are expected to deliver good performance on a variety of tasks while maintaining privacy of user data. Privacy concerns are increasing with the demand for user data. Deep learning models require large amounts of data to be trained well. SH data is incredibly sensitive data since it records a user's private life in the house setting. We investigated if it is possible to run the deep learning models on encoded data instead of raw data and we achieved very similar performance on encoded data. Using encoded

data would relieve the privacy concerns of users, since they would not be sharing their raw data but an encoded version instead.

Other privacy concerns arise when user data needs to be delivered to different external parties. This data sharing is for the user's own benefit and involves for example doctors, carers, researchers and family members. User data should be readable by external parties after transformation according to the privacy preference of user. Our developed model can achieve this using deep learning and creating different views for different users so the system guarantees that only the allowed amount of private information will be shared with external parties. Using our model should enable the external parties to perform their work while maintaining user privacy. Our model is shown without integrating encryption. Adding encryption would make our model stronger and more appealing to users. One possible idea is to supply the encryption key as an input to encoder and decoder and only encode or decode data if correct key is used. Also, we will investigate in future work how to enable device-level encryption for decoder and encoder neural network parameters rather than storing encrypted weight files.

The most important aspect of the SH system is to communicate with the user. SH which can communicate back and forth with user is preferred since dialogue is the natural way of communication of humans. Although humans learn dialogue from very early ages without explicit instruction, it is a very complex ability. There are many different ideas regarding what should be addressed in order to improve dialogue management systems. In this thesis, we first worked on collecting a new dataset which addresses two very important questions about dialogue. One is diversity in dialogue replies; although the same dialogue context may have many good replies, common datasets only have one reply for each context. The other important feature is availability of a good dialogue metric to evaluate how good a reply is, given the dialogue context. A tool like this has utility for many other applications such as comparing different generator models, training RL algorithms or designing new dialogue managers. In future work, task-oriented dialogue dataset will also be collected to investigate diversity of replies in different dialogue settings and how humans react to diverse replies in task-oriented

and non-task-oriented dialogues. Our supervised models provide some correlation with human ratings for dialogue reward. However, it requires improvement and also it is not very practical to train separate supervised model for each different dialogue dataset. Another interesting future work is to develop generative dialogue managers that leverage diverse replies for one dialogue history in our dataset.

Recent advances in deep learning have also improved the deep reinforcement learning field; this is highly applicable to dialogue management since dialogue is more sequential decision planning rather than supervised learning. Deep RL methods, which use deep learning approximators, also require a high amount of data which is expensive to obtain. For this reason, we developed an abstract model which is a lot more sample efficient than state-of-the-art RL methods. Our abstract framework could be instantiated with different settings to fit different dialogue tasks as well. Our deep RL instantiation would be more attractive if stability is improved. Stability can be improved by modifying the sample buffer or applying more advanced entropy control for the policy.

In order to compare and train dialogue managers, a robust dialogue reward function is needed for task-oriented and non-task-oriented. For non-task-oriented we applied LSTM and Transformer based models while for task-oriented we investigated different models of autoencoders, image-based models, character based seq2seq models and transformers with LM. Image-based methods also enable dialogue managers to use different modalities for NLP tasks, which may extend the performance of text-based NLP models. Image-based models we applied process all text as single image which is computationally very expensive since image that holds whole text can be very large. Instead of one single image, many small images can be processed visually separately and further processed with an LSTM or Transformer. Transformers with LM outperform other models in both task-oriented and non-task-oriented dialogue setting and have good correlation with human judgment, this indicates that it may be promising to use transformers as reward function in RL-based dialogue managers. All the models we applied or implemented requires training but a metric, which does not require a training, can be used out-of-box such as translation metrics (BLEU, METEOR, etc.). Therefore, as a future work we will

work on how to develop such metrics which use dialogue history and reply and assess a score on goodness of fit without any training.

Advances in deep learning will eventually improve the user experience and performance of SH systems. SHs with more advanced technologies will be adopted by users to maintain their health and independence. Especially for older people, SHs are crucial and in allowing the individuals to maintain their health in their most comfortable private environment. With the ever-growing aging population in the world, especially in Europe, the need for robust high performance SH continues to grow. Continued advancements in device capabilities and increasing user demand for ambient assisted living technologies will drive more and more widespread utilization of SH technology, and increase quality for this population globally.

Bibliography

[1] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.

[2] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, 2017, pp. 1107–1116.

[4] L. X. and L. B., "Learning to classify text using positive and unlabeled data," *IJCAI*, 2003.

[5] B. Liu, Y. Dai, X. Li, W.-S. Lee, and P. Yu., "Building text classifiers using positive and unlabeled examples," *ICDM*, 2003.

[6] "United nations, department of economic and social affairs, population division (2017). world population ageing 2017 (st/esa/ser.a/408)." 2017. [Online]. Available: https://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2017_Report.pdf

[7] D. Singh, J. Kropf, S. Hanke, and A. Holzinger, "Ambient assisted living technologies from the perspectives of older people and professionals," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2017, pp. 255–266.

[8] D. Singh, I. Psychoula, J. Kropf, S. Hanke, and A. Holzinger, "Users' perceptions and attitudes towards smart home technologies," in *International Conference on Smart Homes and Health Telematics*. Springer, 2018, pp. 203–214.

[9] D. J. Cook, "How smart is your home?" *Science*, vol. 335, no. 6076, pp. 1579–1581, 2012.

[10] "Smart home market (smart kitchen, security & access control, lighting control, home healthcare, hvac control and others): Global industry perspective, comprehensive analysis and forecast, 2016-2022," 2017.

[11] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, "From activity recognition to intention recognition for assisted living within smart homes," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 368–379, 2017.

[12] N. Balta-Ozkan, R. Davidson, M. Bicket, and L. Whitmarsh, "Social barriers to the adoption of smart homes," *Energy Policy*, vol. 63, pp. 363–374, 2013.

[13] T. Fuxreiter, C. Mayer, S. Hanke, M. Gira, M. Sili, and J. Kropf, "A modular platform for event recognition in smart homes," in *The 12th IEEE International Conference on e-Health Networking, Applications and Services*. IEEE, 2010, pp. 1–6.

[14] L. Chen, C. Nugent, and G. Okeyo, "An ontology-based hybrid approach to activity modeling for smart homes," *IEEE Transactions on human-machine systems*, vol. 44, no. 1, pp. 92–105, 2014.

[15] G. Okeyo, L. Chen, and H. Wang, "Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes," *Future Generation Computer Systems*, vol. 39, pp. 29–43, 2014.

[16] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[17] R. Chow, S. Egelman, R. Kannavara, H. Lee, S. Misra, and E. Wang, "Hci in business: A collaboration with academia in iot privacy," in *International Conference on HCI in Business*. Springer, 2015, pp. 679–687.

[18] I. Psychoula, D. Singh, L. Chen, F. Chen, A. Holzinger, and H. Ning, "Users' privacy concerns in iot based applications," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2018, pp. 1887–1894.

[19] J. Wang, Z. Zhang, K. Xu, Y. Yin, and P. Guo, "A research on security and privacy issues for patient related data in medical organization system," *International Journal of Security and Its Applications*, vol. 7, no. 4, pp. 287–298, 2013.

[20] Y. Sun, J. Zhang, Y. Xiong, and G. Zhu, "Data security and privacy in cloud computing," *International Journal of Distributed Sensor Networks*, vol. 10, no. 7, p. 190903, 2014.

[21] H. Wang, L. Sun, and E. Bertino, "Building access control policy model for privacy preserving and testing policy conflicting problems," *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1493–1503, 2014.

[22] P. Robert, A. König, H. Amieva, S. Andrieu, F. Bremond, R. Bullock, M. Ceccaldi, B. Dubois, S. Gauthier, P.-A. Kenigsberg *et al.*, "Recommendations for the use of serious games in people with alzheimer's disease, related disorders and frailty," *Frontiers in aging neuroscience*, vol. 6, p. 54, 2014.

[23] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch, "The effect of avatar realism in immersive social virtual realities," in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. ACM, 2017, p. 39.

[24] I. V. Serban, R. Lowe, P. Henderson, L. Charlin, and J. Pineau, "A survey of available corpora for building data-driven dialogue systems," *arXiv preprint arXiv:1512.05742*, 2015.

[25] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[26] D. Singh, I. Psychoula, E. Merdivan, J. Kropf, S. Hanke, E. Sandner, L. Chen, and A. Holzinger, "Privacy enabled smart home framework with voice assistant," 2019.

[27] E. Merdivan, D. Singh, S. Hanke, and A. Holzinger, "Dialogue systems for intelligent human computer interactions." *Electr. Notes Theor. Comput. Sci.*, 2019.

[28] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, and A. Holzinger, "Convolutional and recurrent neural networks for activity recognition in smart environment," in *Towards integrative machine learning and knowledge extraction*. Springer, 2017, pp. 194–205.

[29] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, "Human activity recognition using recurrent neural networks," in *Machine Learning and Knowledge Extraction*. Springer, 2017.

[30] I. Psychoula, E. Merdivan, D. Singh, L. Chen, F. Chen, S. Hanke, J. Kropf, A. Holzinger, and M. Geist, "A deep learning approach for privacy preservation in assisted living," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018.

[31] I. Psychoula, E. Merdivan, D. Singh, L. Chen, A. Holzinger, and M. Geist, "Privacy preservation with deep learning: The encoders-decoders method," 2019, (article under preparation).

[32] E. Merdivan, D. Singh, S. Hanke, A. Holzinger, and M. Geist, "Human annotated movie dialogues dataset," 2019, (article under review).

[33] E. Merdivan, A. Vafeiadis, D. Kalatzis, S. Henke, J. Kropf, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, R. Hamzaoui *et al.*, "Image-based natural language understanding using 2d convolutional neural networks," *Smart World Congress*, 2019.

[34] E. Merdivan, M. R. Loghmani, and M. Geist, "Reconstruct & crush network," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4548–4556.

[35] E. Merdivan, S. Hanke, and M. Geist, "Modified actor-critics," 2019, (article under review).

[36] R. Lutolf, "Smart home concept and the integration of energy meters into a home based system," in *Metering Apparatus and Tariffs for Electricity Supply, 1992., Seventh International Conference on*. IET, 1992, pp. 277–278.

[37] L. Satpathy, "Smart housing: Technology to aid aging in place: New opportunities and challenges," Ph.D. dissertation, Mississippi State University, 2006.

[38] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—past, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012.

[39] S. K. Das, D. J. Cook, A. Battacharya, E. O. Heierman, and T.-Y. Lin, "The role of prediction algorithms in the mavhome smart home architecture," *IEEE Wireless Communications*, vol. 9, no. 6, pp. 77–84, 2002.

[40] M. C. Mozer, "The neural network house: An environment hat adapts to its inhabitants," in *Proc. AAAI Spring Symp. Intelligent Environments*, vol. 58, 1998.

[41] S. Helal, B. Winkler, C. Lee, Y. Kaddoura, L. Ran, C. Giraldo, S. Kuchibhotla, and W. Mann, "Enabling location-aware pervasive computing applications for the

elderly," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).* IEEE, 2003, pp. 531–536.

[42] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, "The gator tech smart house: A programmable pervasive space," *Computer*, vol. 38, no. 3, pp. 50–60, 2005.

[43] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Proceedings Third IEEE International Workshop on Visual Surveillance*. IEEE, 2000, pp. 3–10.

[44] T. Yamazaki, "Beyond the smart home," in *2006 International Conference on Hybrid Information Technology*, vol. 2. IEEE, 2006, pp. 350–355.

[45] P. Rashidi and D. J. Cook, "Keeping the intelligent environment resident in the loop," 2008.

[46] T. Perumal, A. R. Ramli, and C. Y. Leong, "Design and implementation of soap-based residential management for smart home systems," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 453–459, 2008.

[47] C.-L. Wu, C.-F. Liao, and L.-C. Fu, "Service-oriented smart-home architecture based on osgi and mobile-agent technology," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 2, pp. 193–205, 2007.

[48] A. E. Nikolaidis, S. Papastefanos, G. A. Doumenis, G. I. Stassinopoulos, and M. P. K. Drakos, "Local and remote management integration for flexible service provisioning to the home," *IEEE Communications Magazine*, vol. 45, no. 10, pp. 130–138, 2007.

[49] G. Virone, N. Noury, and J. Demongeot, "A system for automatic measurement of circadian activity deviations in telemedicine," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 12, pp. 1463–1469, 2002.

[50] T. Adlam, R. Faulkner, R. Orpwood, K. Jones, J. Macijauskiene, and A. Budraitiene, "The installation and support of internationally distributed equipment for people with dementia," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 3, pp. 253–257, 2004.

[51] Y. Nishida, T. Hori, T. Suehiro, and S. Hirai, "Sensorized environment for self-communication based on observation of daily human behavior," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 2. IEEE, 2000, pp. 1364–1372.

[52] D. Pishva and K. Takeda, "Product-based security model for smart home appliances," *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, no. 10, pp. 32–41, 2008.

[53] D. N. Kalofonos and S. Shakhshir, "Intuisec: a framework for intuitive user interaction with smart home security using mobile devices," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2007, pp. 1–5.

[54] B. Vaidya, J. H. Park, S.-S. Yeo, and J. J. Rodrigues, "Robust one-time password authentication scheme using smart card for home network environment," *Computer Communications*, vol. 34, no. 3, pp. 326–336, 2011.

[55] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[56] J. Ye, S. Dobson, and S. McKeever, "Situation identification techniques in pervasive computing: A review," *Pervasive and mobile computing*, vol. 8, no. 1, pp. 36–66, 2012.

[57] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic

reasoning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1–12.

[58] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods of information in medicine*, vol. 48, no. 05, pp. 480–485, 2009.

[59] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 838–845.

[60] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 1–9.

[61] D. H. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors," in *International Conference on Pervasive Computing*. Springer, 2005, pp. 62–79.

[62] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International conference on pervasive computing*. Springer, 2004, pp. 1–17.

[63] O. Brdiczka, J. L. Crowley, and P. Reignier, "Learning situation models in a smart home," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 56–63, 2008.

[64] C. Sutton, A. McCallum, and K. Rohanimanesh, "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," *Journal of Machine Learning Research*, vol. 8, no. Mar, pp. 693–723, 2007.

[65] M. Mahdaviani and T. Choudhury, "Fast and scalable training of semi-supervised crfs with application to activity recognition," in *Advances in Neural Information Processing Systems*, 2008, pp. 977–984.

[66] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, and X. Hong, "Semantic smart homes: towards knowledge rich assisted living environments," in *Intelligent Patient Management*. Springer, 2009, pp. 279–296.

[67] L. Chen and C. Nugent, "Ontology-based activity recognition in intelligent pervasive environments," *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009.

[68] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[69] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2015, pp. 149–154.

[70] J. Ye, G. Stevenson, and S. Dobson, "Usmart: An unsupervised semantic mining activity recognition technique," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 4, no. 4, p. 16, 2015.

[71] F. De Backere, F. Ongenae, F. Van Den Abeele, J. Nelis, P. Bonte, E. Clement, M. Philpott, J. Hoebeke, S. Verstichel, A. Ackaert *et al.*, "Towards a social and context-aware multi-sensor fall detection and risk assessment platform," *Computers in biology and medicine*, vol. 64, pp. 307–320, 2015.

[72] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big data privacy in the internet of things era," *IT Professional*, vol. 17, no. 3, pp. 32–39, 2015.

[73] D. Barua, J. Kay, and C. Paris, "Viewing and controlling personal sensor data: what do users want?" in *International Conference on Persuasive Technology*. Springer, 2013, pp. 15–26.

[74] P. Klasnja, S. Consolvo, T. Choudhury, R. Beckwith, and J. Hightower, "Exploring privacy concerns about personal sensing," in *International Conference on Pervasive Computing*. Springer, 2009, pp. 176–183.

[75] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[76] M. Ashwin, K. Daniel, G. Johannes, and V. Muthuramakrishnan, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–52, 2007.

[77] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

[78] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure multiparty computation*. Cambridge University Press, 2015.

[79] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 19–38.

[80] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 1310–1321.

[81] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[82] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: an application of human behavior prediction," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[83] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM*

SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 603–618.

[84] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[85] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[86] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[87] P. Xie, M. Bilenko, T. Finley, R. Gilad-Bachrach, K. Lauter, and M. Naehrig, "Crypto-nets: Neural networks over encrypted data," *arXiv preprint arXiv:1412.6181*, 2014.

[88] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data," *Journal of biomedical informatics*, vol. 50, pp. 234–243, 2014.

[89] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[90] N. Papernot, "A marauder's map of security and privacy in machine learning: An overview of current and future research directions for making machine learning secure and private," in *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*. ACM, 2018, pp. 1–1.

[91] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1322–1333.

[92] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[93] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[94] T. W. Bickmore, L. Caruso, and K. Clough-Gorr, "Acceptance and usability of a relational agent interface by urban older adults," in *CHI'05 extended abstracts on Human factors in computing systems*. ACM, 2005, pp. 1212–1215.

[95] "Utterance segmentation and turn-taking in spoken dialogue systems," 2005.

[96] J. Gao, M. Galley, L. Li *et al.*, "Neural approaches to conversational ai," *Foundations and Trends® in Information Retrieval*, vol. 13, no. 2-3, pp. 127–298, 2019.

[97] A.-r. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," 2009.

[98] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[99] W. Wei, Q. Le, A. Dai, and J. Li, "Airdialogue: An environment for goal-oriented dialogue research," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3844–3854.

[100] C. Chelba, M. Mahajan, and A. Acero, "Speech utterance classification," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 1. IEEE, 2003, pp. I–I.

[101] P. Haffner, G. Tur, and J. H. Wright, "Optimizing svms for complex call classification," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 1. IEEE, 2003, pp. I–I.

[102] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, "Deep belief nets for natural language call-routing," in *2011 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 5680–5683.

[103] G. Tur, L. Deng, D. Hakkani-Tür, and X. He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 5045–5048.

[104] S. Ravuri and A. Stolcke, "Recurrent neural network and lstm models for lexical utterance classification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[105] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu *et al.*, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2015.

[106] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentence-level information with encoder lstm for semantic slot filling," *arXiv preprint arXiv:1601.01530*, 2016.

[107] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, "Neural models for sequence chunking," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[108] C. Hori, T. Hori, S. Watanabe, and J. R. Hershey, "Context sensitive spoken language understanding using role dependent lstm layers," 2015.

[109] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[110] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

[111] Y.-N. Chen, D. Hakkani-Tur, G. Tur, A. Celikyilmaz, J. Gao, and L. Deng, "Knowledge as a teacher: Knowledge-guided structural attention networks," *arXiv preprint arXiv:1609.03286*, 2016.

[112] Y.-N. Chen, D. Hakkani-Tür, J. Gao, and L. Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding." 2016.

[113] S.-Y. Su, P.-C. Yuan, and Y.-N. Chen, "How time matters: Learning time-decay attention for contextual spoken language understanding in dialogues," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2133–2142.

[114] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "Investigation of language understanding impact for reinforcement learning based dialogue systems," *arXiv preprint arXiv:1703.07055*, 2017.

[115] A. Stent, M. Marge, and M. Singhai, "Evaluating evaluation methods for generation in the presence of variation," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2005, pp. 341–351.

[116] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," *arXiv preprint arXiv:1603.08023*, 2016.

[117] A. H. Oh and A. I. Rudnicky, "Stochastic natural language generation for spoken dialog systems," *Computer Speech & Language*, vol. 16, no. 3-4, pp. 387–407, 2002.

[118] T.-H. Wen, M. Gasic, D. Kim, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, "Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking," *arXiv preprint arXiv:1508.01755*, 2015.

[119] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned lstm-based natural language generation for spoken dialogue systems," *arXiv preprint arXiv:1508.01745*, 2015.

[120] O. Dušek and F. Jurčíček, "Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings," *arXiv preprint arXiv:1606.05491*, 2016.

[121] ——, "A context-aware natural language generator for dialogue systems," *arXiv preprint arXiv:1608.07076*, 2016.

[122] S.-Y. Su, K.-L. Lo, Y.-T. Yeh, and Y.-N. Chen, "Natural language generation by hierarchical decoding with linguistic patterns," *arXiv preprint arXiv:1808.02747*, 2018.

[123] D. Jurafsky and M. H. James, "Speech and laguage processing 3rd edition," 9 2018, stanford, https://web.stanford.edu/ jurafsky/slp3/ Draft version.

[124] I. V. Serban, R. Lowe, P. Henderson, L. Charlin, and J. Pineau, "A survey of available corpora for building data-driven dialogue systems: The journal version," *Dialogue & Discourse*, vol. 9, no. 1, pp. 1–49, 2018.

[125] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," *arXiv preprint arXiv:1605.07683*, 2016.

[126] A. Madotto, C.-S. Wu, and P. Fung, "Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems," *arXiv preprint arXiv:1804.08217*, 2018.

[127] V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goodine, D. Goddeau, and J. Glass, "Pegasus: A spoken dialogue interface for on-line air travel planning," *Speech Communication*, 1994.

[128] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi, "Let's go public! taking a spoken dialog system to the real world," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[129] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, "Gus, a frame-driven dialog system," *Artificial intelligence*, vol. 8, no. 2, pp. 155–173, 1977.

[130] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai, "A form-based dialogue manager for spoken language applications," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 2. IEEE, 1996, pp. 701–704.

[131] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 263–272.

[132] B. A. Shawar and E. Atwell, "Chatbots: are they really useful?" 2007.

[133] K. M. Colby, "Artificial paranoia: A computer simulation of paranoid processes," 1975.

[134] R. S. Wallace, "The anatomy of alice," in *Parsing the Turing Test*. Springer, 2009, pp. 181–210.

[135] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The design and implementation of xiaoice, an empathetic social chatbot," *arXiv preprint arXiv:1812.08989*, 2018.

[136] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364*, 2015.

[137] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[138] Z. Ji, Z. Lu, and H. Li, "An information retrieval approach to short text conversation," *arXiv preprint arXiv:1408.6988*, 2014.

[139] Z. Yan, N. Duan, J. Bao, P. Chen, M. Zhou, Z. Li, and J. Zhou, "Docchat: An information retrieval approach for chatbot engines using unstructured documents," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 516–525.

[140] R. T. Lowe, N. Pow, I. V. Serban, L. Charlin, C.-W. Liu, and J. Pineau, "Training end-to-end dialogue systems with the ubuntu dialogue corpus," *Dialogue & Discourse*, vol. 8, no. 1, pp. 31–65, 2017.

[141] Y. Wu, Z. Li, W. Wu, and M. Zhou, "Response selection with topic clues for retrieval-based chatbots," *Neurocomputing*, vol. 316, pp. 251–261, 2018.

[142] I. V. Serban, C. Sankar, M. Germain, S. Zhang, Z. Lin, S. Subramanian, T. Kim, M. Pieper, S. Chandar, N. R. Ke *et al.*, "A deep reinforcement learning chatbot," *arXiv preprint arXiv:1709.02349*, 2017.

[143] Z. Wei, Q. Liu, B. Peng, H. Tou, T. Chen, X. Huang, K.-F. Wong, and X. Dai, "Task-oriented dialogue system for automatic diagnosis," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 201–207.

[144] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "End-to-end task-completion neural dialogue systems," *arXiv preprint arXiv:1703.01008*, 2017.

[145] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[146] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[147] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[148] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[149] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[150] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *arXiv preprint arXiv:1610.10099*, 2016.

[151] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[152] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[153] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," *arXiv preprint arXiv:1506.06714*, 2015.

[154] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[155] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[156] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[157] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," *arXiv preprint arXiv:1703.10960*, 2017.

[158] S. Jiang and M. de Rijke, "Why are sequence-to-sequence models so dull? understanding the low-diversity problem of chatbots," *arXiv preprint arXiv:1809.01941*, 2018.

[159] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," *arXiv preprint arXiv:1510.03055*, 2015.

[160] S. J. Young, "Probabilistic methods in spoken–dialogue systems," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1769, pp. 1389–1402, 2000.

[161] E. Levin, R. Pieraccini, and W. Eckert, "Using markov decision process for learning dialogue strategies," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 1. IEEE, 1998, pp. 201–204.

[162] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[163] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[164] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[165] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[166] S. Singh, M. Kearns, D. J. Litman, M. A. Walker *et al.*, "Empirical evaluation of a reinforcement learning spoken dialogue system," 2000.

[167] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue management with reinforcement learning: Experiments with the njfun system," *Journal of Artificial Intelligence Research*, pp. 105–133, 2002.

[168] O. Pietquin, M. Geist, S. Chandramohan, and H. Frezza-Buet, "Sample-efficient batch reinforcement learning for dialogue management optimization," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 7, no. 3, p. 7, 2011.

[169] P.-H. Su, M. Gasic, N. Mrksic, L. Rojas-Barahona, S. Ultes, D. Vandyke, T.-H. Wen, and S. Young, "Continuously learning neural dialogue management," *arXiv preprint arXiv:1606.02689*, 2016.

[170] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *arXiv preprint arXiv:1606.01541*, 2016.

[171] B. Dhingra, L. Li, X. Li, J. Gao, Y.-N. Chen, F. Ahmed, and L. Deng, "Towards end-to-end reinforcement learning of dialogue agents for information access," *arXiv preprint arXiv:1609.00777*, 2016.

[172] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, "Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1437–1447.

[173] Z. Yu, A. W. Black, and A. I. Rudnicky, "Learning conversational systems that interleave task and non-task content," *arXiv preprint arXiv:1703.00099*, 2017.

[174] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *arXiv preprint arXiv:1701.06547*, 2017.

[175] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems." Association for Computational Linguistics, 2015.

[176] A. Ritter, C. Cherry, and B. Dolan, "Unsupervised modeling of twitter conversations," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 172–180.

[177] C. Danescu-Niculescu-Mizil and L. Lee, "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs," in *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*. Association for Computational Linguistics, 2011, pp. 76–87.

[178] T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," *arXiv preprint arXiv:1604.04562*, 2016.

[179] L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman, "Frames: A corpus for adding memory to goal-oriented dialogue systems," *arXiv preprint arXiv:1704.00057*, 2017.

[180] P. Shah, D. Hakkani-Tür, G. Tür, A. Rastogi, A. Bapna, N. Nayak, and L. Heck, "Building a conversational agent overnight with dialogue self-play," *arXiv preprint arXiv:1801.04871*, 2018.

[181] U. DESA, "United nations department of economic and social affairs,population division (2015): World population ageing 2015," 2015. [Online]. Available: http://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2015_Report.pdf

[182] R. Salakhutdinov, "Learning deep generative models," *Annual Review of Statistics and Its Application*, vol. 2, pp. 361–385, 2015.

[183] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5, pp. 555–559, 2003.

[184] T. L. Kasteren, G. Englebienne, and B. J. Kröse, "Human activity recognition from wireless sensor network data: Benchmark and software," *Activity recognition in pervasive intelligent environments*, pp. 165–186, 2011.

[185] H. Alemdar, H. Ertan, O. D. Incel, and C. Ersoy, "Aras human activity datasets in multiple homes with multiple residents," in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2013, pp. 232–235.

[186] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, 2015.

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

[188] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: Deep neural networks over encrypted data," *arXiv preprint arXiv:1711.05189*, 2017.

[189] *Health Insurance Portability and Accountability Act Of 1996*, https://www.gpo.gov/fdsys/pkg/PLAW-104publ191/html/PLAW-104publ191.htm.

[190] "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," *OJ*, vol. L 119, pp. 1–8, 2016-05-4.

[191] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

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

[193] J. Tiedemann, "News from opus-a collection of multilingual parallel corpora with tools and interfaces," in *Recent Advances in Natural Language Processing (RANLP)*, 2009.

[194] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics (ACL)*. Association for Computational Linguistics, 2002.

[195] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[196] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. Association for Computational Linguistics, 2005.

[197] R. Artstein, S. Gandhe, J. Gerten, A. Leuski, and D. Traum, "Semi-formal evaluation of conversational characters," in *Languages: From formal to natural*. Springer, 2009.

[198] P. Forchini, "Spontaneity reloaded: American face-to-face and movie conversation compared," in *Corpus Linguistics*, 2009.

[199] R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "On the evaluation of dialogue systems with next utterance classification," in *Proceedings of the SIGDIAL 2016 Conference*. Association for Computational Linguistics, 2016.

[200] R. E. Banchs, "Movie-dic: a movie dialogue corpus for research and development," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2012.

[201] J. Tiedemann, "Parallel data, tools and interfaces in opus." in *International Conference on Language Resources and Evaluation (LREC)*, 2012.

[202] X.-L. Li, B. Liu, and S.-K. Ng, "Negative training data can be harmful to text classification," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2010.

[203] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*. Association for Computational Linguistics, 2016.

[204] R. Lowe, M. Noseworthy, I. V. Serban, N. Angelard-Gontier, Y. Bengio, and J. Pineau, "Towards an automatic turing test: Learning to evaluate dialogue responses," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2017.

[205] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[206] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, 2005.

[207] A. B. Sai, M. D. Gupta, M. M. Khapra, and M. Srinivasan, "Re-evaluating adem: A deeper look at scoring dialogue responses," *AAAI*, 2019.

[208] S. Sharma, L. El Asri, H. Schulz, and J. Zumer, "Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation," in *CoRR*, 2017.

[209] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014.

[210] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[211] ——, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[212] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[213] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[214] S. Koelsch, E. Kasper, D. Sammler, K. Schulze, T. Gunter, and A. D. Friederici, "Music, language and meaning: brain signatures of semantic processing," *Nature neuroscience*, vol. 7, no. 3, p. 302, 2004.

[215] P. G. Simos, L. F. Basile, and A. C. Papanicolaou, "Source localization of the n400 response in a sentence-reading paradigm using evoked magnetic fields and magnetic resonance imaging," *Brain research*, vol. 762, no. 1-2, pp. 29–39, 1997.

[216] S. Cèbe and R. Goigoux, *Apprendre à lire à l'école: Tout ce qu'il faut savoir pour accompagner l'enfant*. Retz, 2011.

[217] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[218] A. Copestake, "Augmented and alternative nlp techniques for augmentative and alternative communication," *Natural Language Processing for Communication Aids*, 1997.

[219] Í. de Pontes Oliveira, J. L. P. Medeiros, and V. F. de Sousa, "A data augmentation methodology to improve age estimation using convolutional neural networks," in *Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference on*. IEEE, 2016, pp. 88–95.

[220] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[221] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 165–172.

[222] W. Huang, "Character-level convolutional network for text classification applied to chinese corpus," Ph.D. dissertation, University College London, 2016.

[223] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," *RecSys*, 2013.

[224] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *EMNLP*, 2014.

[225] A. Bordes and J. Weston, "Learning end-to-end goal-oriented dialog," *arXiv:1605.07683*, 2016.

[226] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *International Conference on Learning Representations (ICLR)*, 2016.

[227] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, "Sample efficient actor-critic with experience replay," *International Conference on Learning Representations (ICLR)*, 2017.

[228] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning (ICML)*, 2016.

[229] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning (ICML)*, 2015.

[230] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[231] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," in *Advances in neural information processing systems (NeurIPS)*, 2017.

[232] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International Conference on Machine Learning (ICML)*, 2017.

[233] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *International Conference on Machine Learning (ICML)*, 2018.

[234] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[235] V. Gabillon, A. Lazaric, M. Ghavamzadeh, and B. Scherrer, "Classification-based policy iteration with a critic," in *International Conference on Machine Learning (ICML)*, 2011.

[236] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2002.

[237] M. L. Puterman and M. C. Shin, "Modified policy iteration algorithms for discounted markov decision problems," *Management Science*, 1978.

[238] B. Scherrer, M. Ghavamzadeh, V. Gabillon, B. Lesner, and M. Geist, "Approximate modified policy iteration and its application to the game of tetris." *Journal of Machine Learning Research (JMLR)*, 2015.

[239] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare, "Safe and efficient off-policy reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[240] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[241] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[242] A. Abdolmaleki, J. T. Springenberg, J. Degrave, S. Bohez, Y. Tassa, D. Belov, N. Heess, and M. Riedmiller, "Relative entropy regularized policy iteration," *arXiv preprint arXiv:1812.02256*, 2018.

[243] T. De Bruin, J. Kober, K. Tuyls, and R. Babuška, "Experience selection in deep reinforcement learning for control," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 347–402, 2018.

Bordereau de thèse soutenue

Aide ambiante à la personne par apprentissage profond

L'aide ambiante à la personne (ambiant assisted living) a pour objectif d'accompagner le vieillissement de la population. Cela s'instancie notamment par les maisons intelligentes (smart homes), équipées de multiples capteurs connectés, dont un des objectifs est de prolonger le maintien à domicile des personnes âgées. Le manuscrit s'attache d'abord à introduire la problématique générale des maisons intelligentes, avant de présenter plus avant les trois sous-thématiques qui font plus particulièrement l'objet de la thèse, à savoir la reconnaissance d'activités, la confidentialité et les systèmes de dialogue.

La reconnaissance d'activités consiste à déterminer les activités courantes d'une personne ou d'un groupe de personnes, à partir des données (brutes) des capteurs dont est équipée la maison. On peut citer comme exemple la détection de la chute d'une personne. Une maison intelligent repose typiquement sur l'internet des objets (Internet of Things, ou IoT). De nombreuses données sont produites, pouvant contenir des informations privées ou sensibles. Une partie de ces données doit être partagée avec l'extérieur, ce qui peut poser des problèmes de confidentialité. Enfin, pour interragir avec la maison intelligente, un moyen naturel pour l'utilisateur est d'utiliser le dialogue, sujet traité par les systèmes de dialogue.

Ce travail de thèse propose des contributions sur ces trois versants, la plupart basées sur l'apprentissage profond.

Mots-clés en français (5 environ) : Aide ambiante à la personne, apprentissage profond, reconnaissance d'activités,systèmes de dialogue, maisons intelligentes

Ambient Assisted Living with Deep Learning

Ambient assisted living aims to support the aging population. This is particularly the case with smart homes, equipped with multiple connected sensors, which enables to extend home care for the elderly. The manuscript begins by introducing the general problem of smart homes, after presenting further the three sub-themes that are the subject of the thesis, namely the activity recognition, privacy and dialogue systems.

Activity recognition is the process of determining the day-to-day activities of a person or a group of people from the (raw) sensor data that the home is equipped with. An example of this is the detection of a person's fall. A smart home is typically based on the Internet of Things (IoT). Many data are produced, which may contain private or sensitive information. Some of this data must be shared externally, which may pose privacy issues. Finally, a natural way of communication for the user is to use the dialogue to interact with the smart home via dialogue manager.

This thesis proposes contributions on these three sides, most of them based on deep learning.

Mots-clés traduits en anglais (5 environ) :

Ambient assisted living, deep learning, Activity recognition, dialogue manager, smart home