Unsupervised Transfer Learning for Human Behavior Classification

著者（英）	Batsergelen Myagmar
year	2020
その他のタイトル	人の行動分類のための教師なし転移学習
学位授与大学	筑波大学 (University of Tsukuba)
学位授与年度	2019
報告番号	12102甲第9402号
URL	http://doi.org/10.15068/00161548

Unsupervised Transfer Learning for Human Behavior Classification

March　２０２０

Myagmar Batsergelen

Unsupervised Transfer Learning for Human Behavior Classification

Graduate School of Systems and Information Engineering
University of Tsukuba

March　２０２０

Myagmar Batsergelen

Unsupervised Transfer Learning for Human Behavior Classification

Myagmar Batsergelen

Abstract

Machine-learning technology are becoming prevalent in modern society. Most of the impressive performances of recent machine learning approaches come mostly via the supervised deep learning process where massive amounts of labeled data are used to create high-performing models. However, in real world applications, there are many scenarios where it is difficult to collect sufficiently large data. Transfer learning techniques try to transfer the knowledge from some source task or source domain with abundant labeled data to help improve the prediction performance in the target task or target domain with little or no labeled data.

In this thesis, we address the challenging problem of transfer learning for heterogeneous domains through three different levels of common feature representations in two human behavior classification scenarios of physical behavior classification in smart-home activities of daily living (ADL) recognition and verbal behavior classification with sentiment analysis. For human behavior classification, we propose novel low-level heuristic mappings between heterogeneous sensor features from different smart-home datasets and adapt Long Short-Term Memory (LSTM) networks for cross-domain activities of daily living (ADL) classification and show their effectiveness in real-life smart-home datasets. For verbal behavior classification, we explore the usage of Transformer-based bidirectional language models for cross-domain sentiment classification. Then, we present a cross-lingual sentiment classification framework based on BERT and a novel non-task specific English-Japanese parallel sentiment corpus.

Acknowledgements

First, I would like to express my sincere gratitude to my PhD supervisors Dr. Shigetomo Kimura and Dr. Jie Li for giving me the opportunity to do research with freedom, while steadily providing me with great advice and support to pursue my research interest.

I would like to thank Dr. Hirotake Abe, Dr. Kazuhiro Fukui, Dr. Yongbing Zhang for constructive feedbacks on drafts of this thesis. I am very grateful to Dr. Keisuke Kameyama for giving me crucial guidance and encouragement during the preparation of this thesis.

Last but not least, I am deeply thankful to my family for their support throughout my PhD study. Special gratitude goes to my wonderful wife Myagmarsuren for her inspirational kindness, patience, and encouragement that made it all possible.

I dedicate this dissertation to my wife and my bright son Erkhem, who was born when I first started my PhD study and from whom I learn everyday about the wonders and the mysteries of human behavior.

Contents

Chapter 1. Introduction 1

1.1. Motivation and problem statement 1

1.2. Problem formulation 6

1.3. Contributions 7

1.4. Thesis outline 8

Chapter 2. Background 9

2.1. Physical behavior classification from embedded sensors 9

2.1.1. Sensor modalities and features 11

2.1.2. ADL feature extraction 15

2.1.3. ADL recognition methods 16

2.2. Verbal behavior classification from text documents 19

2.2.1. Traditional lexicon-based sentiment classification 20

2.2.2. Deep learning sentiment classification 21

2.2.3. Transfer learning sentiment classification 23

Chapter 3. Low-level common feature extraction for cross-domain ADL recognition 26

3.1. Introduction 26

3.2. Related works 27

3.3. Heuristic common sensor feature extraction for cross-domain ADL recognition 29

3.3.1. Sensor location mapping 30

3.3.2. Sensor type mapping 31

3.3.3. Sensor value mapping 31

3.3.4. Daily activity partition hour mapping 31

3.3.5. Contextual time association within sliding window 33

3.3.6. ADL classification with LSTM networks 34

3.4. Experiments 36

3.4.1. Dataset 38

3.4.2. Implementation details 42

3.4.3. Baselines 42

3.4.4. Experimentation scenarios 43

3.4.5. Results 44

3.5. Conclusion 47

Chapter 4. Mid-level common feature representations with pre-trained language models 52

4.1. Introduction 52

4.2. Related works 54

4.3. Bidirectional pre-trained transformer language models 56

4.3.1. Transformer 56

4.3.2. BERT 59

4.3.3. XLNet 60

4.3.4. Fine-tuning for cross-domain sentiment classification 62

4.4. Experiments 63

4.4.1. Dataset 63

4.4.2. Pre-training 64

4.4.3. Implementation details 64

4.4.4. Baselines 65

4.4.5. Results 67

4.5. Conclusion 72

Chapter 5. High-level common feature representation with parallel corpus for cross-lingual sentiment classification 73

5.1. Introduction 73

5.2. Related works 76

5.3. Bert-based unsupervised cross-lingual (BUCL) sentiment classification framework ... 77

5.4. Experiments 79

5.4.1. Baselines 79

5.4.2. Implementation details 80

5.4.3. Results 81

5.5. Conclusion 84

Chapter 6. Summary and future work 86

Bibliography 88

List of figures

Figure 1.1. Deep learning structure.. 3

Figure 1.2. Multi-level transfer learning framework. 5

Figure 2.1. Data collection categories for HAR 11

Figure 3.1. ADL data collection from embedded sensors. 26

Figure 3.2. Proposed cross-domain ADL recognition with common low-level feature representation... 30

Figure 3.3. hh101 house sensor locations 36

Figure 3.4. hh103 house sensor locations 37

Figure 3.5. hh109 house sensor locations 37

Figure 3.6. hh123 house sensor locations 38

Figure 3.7. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.. 45

Figure 3.8. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.. 45

Figure 3.9. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.. 46

Figure 3.10. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.. 46

Figure 3.11. Single-source, single-target classification accuracy... 48

Figure 3.12. Single-source, single-target classification accuracy... 48

Figure 3.13. Single-source, single-target F1 score. 49

Figure 3.14. Single-source, single-target F1 score. 49

Figure 3.15. Confusion matrix when evaluated with 'Unknown' activity included...................... 50

Figure 3.16. Confusion matrix when evaluated without 'Unknown' class samples...................... 51

Figure 4.1. Transformer encoder architecture... 57

Figure 4.2. Mid-level common feature representation with pre-trained language models for CDSC. 63

Figure 4.3. CDSC accuracy rates over different source domain labeled training data size. Horizontal axis is the training data size. 69

Figure 4.4. Shows how three XLNet models trained with different number of source domain labeled data (60, 600, 6000) performs over increasing number of training steps (300, 1000, 3000, 9000, 30000). 70

Figure 4.5. Shows how three BERT models trained with different number of source domain labeled data (60, 600, 6000) performs over increasing number of training steps (300, 1000, 3000, 9000, 30000). 71

Figure 5.1. Bert-based Unsupervised Cross-Lingual (BUCL) transfer learning framework........ 78

Figure 5.2. Cross-lingual sentiment classification accuracy on Japanese Amazon sentiment dataset. 83

Figure 5.3. Cross-lingual sentiment classification accuracy on Japanese Rakuten sentiment dataset. 84

List of tables

Table 2.1. List of ADLs 10

Table 3.1. Sensor location tags and the corresponding physical areas 30

Table 3.2. Sensor types 31

Table 3.3. Daily activity partitions and the ADLs occuring with the partition 32

Table 3.4. Example of a partitioned 24 hour period 33

Table 3.5. Activity class instance percentages for each dataset. ‘Unknown’ class excluded....... 40

Table 3.6. Activity class instance percentages for each dataset. ‘Unknown’ class included. 41

Table 4.1. Cross-domain sentiment classification on Amazon sentiment dataset. 66

Table 4.2. CDSC accuracy rates on different source domain training data size........................... 68

Table 4.3. Training run time over different training step numbers.. 70

Table 4.4. Test run times over different test data sizes (20, ..., 6000) 70

Chapter 1. Introduction

"The conception of a free, responsible individual is embedded in our language and pervades our practices, codes, and beliefs. Given an example of human behavior, most people can describe it immediately in terms of such a conception. The practice is so natural that it is seldom examined. A scientific formulation, on the other hand, is new and strange. Very few people have any notion of the extent to which a science of human behavior is indeed possible. In what way can the behavior of the individual or of groups of individuals be predicted and controlled? What are laws of behavior like? What overall conception of the human organism as a behaving system emerges? It is only when we have answered these questions, at least in a preliminary fashion, that we may consider the implications of a science of human behavior with respect to either a theory of human nature or the management of human affairs."

Burrhus Frederic Skinner, *Science and human behavior,* 1965.

1.1. Motivation and problem statement

Making machines that can correctly perceive and interact with humans have long captured our imagination ever since the notion of automation was first conceived. Currently we are still only at the first stage of trying to make machines that can understand us accurately. *Artificial Intelligence (AI)* is an acutely thriving research field that attempts to build "intelligent agents", as defined in Russel *et al.* [2009]. Over the last couple of decades, the consensus has become that AI systems need the ability to acquire their own knowledge by extracting patterns from raw data and Goodfellow *et al.* [2016] defines this capability is known as *machine learning (ML)*. The input to a machine learning algorithm is training data and an optimization objective, and the output is some expertise acquired through training that can be utilized to perform prediction tasks that are too complex to directly program.

Machine-learning technology are becoming prevalent in modern society, from web searches to content filtering on social networks to recommendations on e-commerce websites, and in consumer products such as cameras and smartphones. Machine learning systems are used to identify objects in images, provide recommendations in accordance with users' online activities, transcribe speech into text, and select relevant results of search. There are still many not fully realized areas of applications for machine learning systems and we discuss two such examples that works toward recognizing human behavior in an automated manner: **physical behavior classification from embedded sensors, i.e. activities of daily living recognition,** and **verbal behavior classification from text documents, i.e. sentiment classification**.

Since the global increase in the ratio of the elderly population is already prominent, the aging at home gained substantial importance. Most older adults prefer to stay in the comfort of their own homes, and given the costs of nursing home care, it is important to develop technologies that help older adults to age at home. Human Activity Recognition (HAR) is one of the most promising research topics under the rapid development of ubiquitous technologies. The goal of HAR is to identify user's activities based on context information collected by sensors. In the literature, many activity recognition approaches have been proposed and most of them deal with data collected from video cameras, or wearable sensors. However, the problem with video cameras and wearable sensors are their intrusive nature of data collection methods, in addition to the privacy issues. Ambient sensors, on the other hand, are used to capture the interaction between humans and the environment in a nonintrusive way. The sensors are embedded in users' smart environment and **activities of daily living (ADL)** is detected through changes in the environment. In comparison to the video and wearable sensor-based approaches, much fewer methods have been proposed in recognizing ADL using ambient non-invasive sensors embedded in smart homes. An ADLs recognition algorithm takes as input the pre-processed sensor events, extracts features from a window of the time series, learn a classification model based on the features for inferring the activity and produces as output the most likely performed activities.

With the user sentiment and opinion expressions becoming widespread throughout social and e-commerce platforms, correctly understanding these opinions and views becomes important in facilitating various service-based applications. **Sentiment classification**, an important task of Natural Language Processing (NLP), aims to identify the polarity of people's opinions towards entities such as products, services, organizations, individuals, issues, etc. Usually this polarity is binary (positive or negative) or ternary (positive, negative, or neutral). Most datasets belong to domains that contain a large number of emotive texts such as movie and product reviews or tweets. With the abundance of available raw data in various social and e-commerce platforms, correctly identifying users' needs and tendencies can facilitate in creating more suited communication between people that enhances common understandings, which results in better service for the said platforms.

Most of the impressive performances of recent machine learning approaches come mostly via the ***supervised deep learning*** process where massive amounts of labeled data are used to create high-performing models. Figure 1.1 illustrates the general deep learning structure, where first some low-level feature mappings are extracted from the raw data, represented within some input feature space, either using pre-processing methods or shallow neural networks. Examples of such low-level mappings are features representing edges and corners in images or word embeddings in text documents. Afterwards, the low-level features are passed to deeper neural networks for extracting higher-level abstract features and contexts. Finally, the high-level features are mapped to expected outputs, i.e. labels, via some fully connected layer or softmax function layer to learn the discernable patterns within the input data.

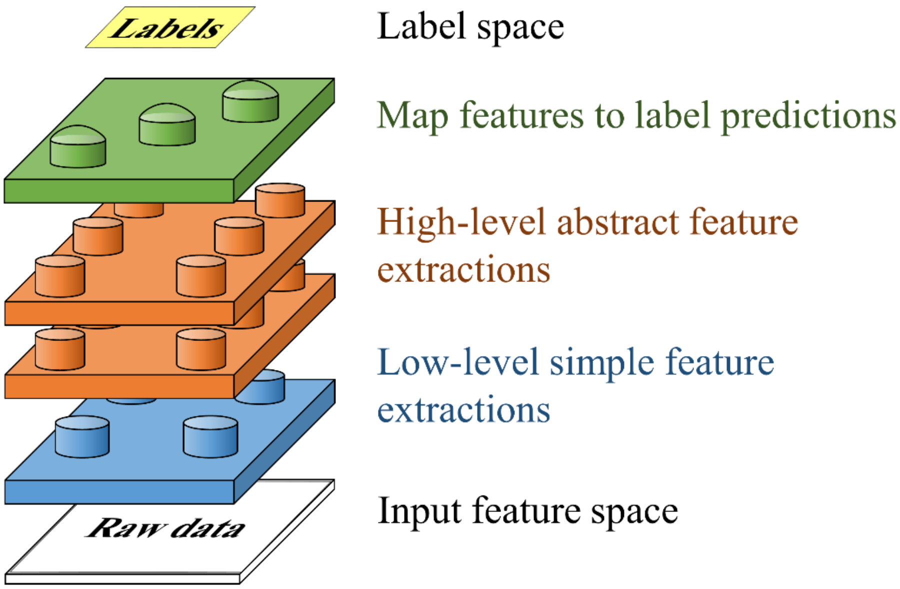

Labels
Label space
Map features to label predictions
High-level abstract feature extractions
Low-level simple feature extractions
Raw data
Input feature space

Figure 1.1. Deep learning structure

However, in real world applications, there are many scenarios where it is difficult to collect sufficiently large data for high-performing supervised learning model of a specific task due to factors of scarcity of readily available data or the high expense of data collection. In addition, statistical learning methods hold the i.i.d. assumption that both the training and test data come from a common underlying distribution, but due to the high variability of human behaviors and data collection methods, oftentimes there is distribution differences in the real-world data and the specialized training data. Recent studies by Jia *et al.* [2017], and Belinkov *et al.* [2018], show that current machine learning algorithms do not generalize beyond the data they have seen during training. They conform to the characteristics of the data they have been trained on and are not able to adapt when conditions change.

These sort of common problems demonstrate the need for machine learning algorithms that can learn efficiently from a small amount of labeled training data by leveraging knowledge from related unlabeled or noisy labeled data or differently distributed data. The research direction that deals with these kind of problems is called ***transfer learning (TL)***. The study of transfer learning is motivated by the fact that humans apply previously learned knowledge to solve new problems efficiently. Compared to traditional supervised learning techniques which try to learn each task from scratch, transfer learning techniques try to transfer the knowledge from some source task or source domain with abundant labeled data to help improve the prediction performance in the target task or target domain with little or no labeled data.

For less researched ML application areas, it has become common practice to adapt the transfer learning methods from well researched fields such as computer vision (CV) and natural language processing (NLP) because in ML classification tasks where some target label data are available, similar transfer learning methods can be applied cross-task, even if the tasks are substantially different. However, when no labeled data are available in target domain, transfer learning methods must rely on common low-level to high-level feature representations between source and target domains, and hence it becomes difficult adapting TL methods cross-tasks. In CV and NLP tasks, there are well established such common feature extraction methods, such as Convolutional Neural Networks (CNN) models pre-trained on labeled Imagenet dataset in CV or word-embeddings pre-trained on unlabeled large corpus in NLP, that can provide adequate mapping between source and target domains, even if they have heterogeneous features and no labeled data in the target domain. However, in less researched application areas such as ADL recognition, there are no such established low-level mapping methods available. Also, even though there has been great progress in TL methods for NLP tasks, most of the methods have been based on English language datasets.

In our thesis we focus on ***unsupervised transfer learning (UTL)***, which we define as the TL scenario of directly using a classification model trained on the source domain labeled data to predict the class labels of target domain without using any labeled or unlabeled data in the target domain.

In this thesis, we argue that UTL can only be performed if there exist common feature representations across both source and target domains, and that the common features can be represented in different abstraction levels. We shall refer to such transfer learning scenario as **multi-level transfer learning**. In our works, we propose and explore following three abstraction levels of multi-level transfer learning:

1. Low-level common features with coarse-grained heuristic sensor feature mapping for ADL recognition across heterogeneous sensor spaces in Chapter 3,
2. Mid-level common features with pre-trained contextualized language models for cross-domain sentiment classification in Chapter 4,
3. High-level common features with bilingual parallel corpus for cross-lingual sentiment classification in Chapter 5.

Figure 1.2 illustrates the proposed multi-level transfer learning framework.

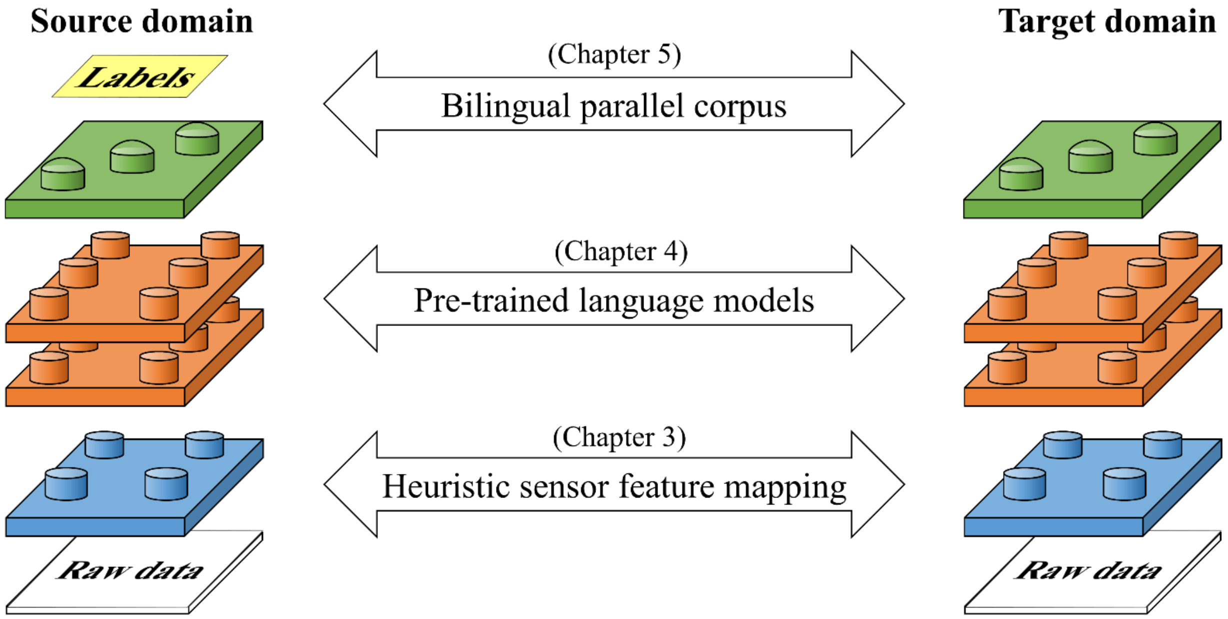

Source domain
Target domain
Labels
(Chapter 5)
Bilingual parallel corpus
(Chapter 4)
Pre-trained language models
(Chapter 3)
Heuristic sensor feature mapping
Raw data
Raw data

Figure 1.2. Multi-level transfer learning framework.

1.2. Problem formulation

Transfer learning's objective is to leverage and transfer knowledge from a different but related source domain to train a prediction model for a target domain. We define transfer learning following the notations in the works of Pan and Yang, [2010, and Ruder [2019], with both ADL recognition and sentiment classification as running examples.

The main concepts in transfer learning are the domain and the task. A domain $\mathcal{D}$ is composed of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\} \in \mathcal{X}$. In the case of ADL recognition, $\mathcal{X}$ represents the space of binary values for all sensors, where x_i is the i-th sensor event expressed as a vector of sensor values. For sentiment classification, we shall use a binary bag-of words representation of an input text document and the feature space $\mathcal{X}$ would be the set of all possible binary term vectors of all the words in pre-determined vocabulary, x_i is the i-th word or token of input text. For both scenarios, X is the learning sample selected from the input training data.

Given a specific domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a learning task $\mathcal{T} = \{\mathcal{Y} | P(Y|X)\}$ consists of two components: a label space $\mathcal{Y}$ and a conditional probability distribution $P(Y|X)$ that is not observed but learned from the training data pairs of $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. For multi-labeled ADL recognition task, $\mathcal{Y}$ is the set of all activity labels such as 'Eat_Lunch', 'Bathe', 'Cook_Dinner', etc., with y_i having a value of $\{1, \dots, \mathcal{C}\}$ where $\mathcal{C}$ is the number of activity labels. In the context of binary sentiment classification task, $\mathcal{Y}$ is the set of two sentiment labels representing the positive and negative sentiments, with y_i having a value of either 1 or 0. For both scenarios, Y is the random variable associated with the input sample's label.

Given a source domain $\mathcal{D}_S$ with its task $\mathcal{T}_S$ and a target domain $\mathcal{D}_T$ with its task $\mathcal{T}_T$, the objective of *transfer learning* is to learn the conditional probability distribution $P_T(Y_T|X_T)$ in $\mathcal{D}_T$ by utilizing the knowledge learned from $\mathcal{D}_S$ and $\mathcal{D}_T$, where either $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ and sufficient labeled training data are available in $\mathcal{D}_S$.

The differences between source and target can be generally divided into the following four categories:

1) The conditional probability distributions are different, i.e. $P_S(Y_S|X_S) \neq P_T(Y_T|X_T)$. Here the label classes of source and target samples are unbalanced.

2) The label spaces are different, i.e. $\mathcal{Y}_S \neq \mathcal{Y}_T$. In this case, the source and target tasks assign different labels to the samples. This issue is commonly faced in *multi-task learning* problems.
3) The marginal probability distributions are different, i.e. $P_S(X_S) \neq P_T(X_T)$. For ADL, this means activities are performed in different ways by the subjects. For sentiment analysis, the documents are discussing different topics. This case is often referred to as *domain adaptation*.
4) The feature spaces are different, i.e. $X_S \neq X_T$. For ADL, the sensor locations and/or physical environments are different. For sentiment analysis, the sample texts are in different languages. This case is often referred to as *heterogeneous transfer learning.*

For this thesis, we will address transfer learning application scenarios where all of the above-mentioned transfer learning issues are present except the case of different label spaces. For ADL recognition task, the time-series sensor events of source and target domains are obtained from separate smart homes with different occupants and the multi-label classes of activities are the same. For sentiment analysis, the documents of source and target domains will be in different languages and the binary label class is the same. For both application scenarios, we will not have any labeled training data in the target domain.

Therefore, the objective of the both of our proposed multi-level transfer learning methods is to learn a robust classifier $P_S(Y_S|X_S)$ trained on labeled data in the source domain to predict the unlabeled examples from the target domain using the learned classifier $P_S(Y_T|X_T)$, where both domains have the same label space $Y_T = Y_S$.

1.3. Contributions

The contributions in this thesis are as follows:

- we propose novel low-level heuristic mappings between heterogeneous sensor features from different smart-home datasets.
- we adapt LSTM networks for low-level UTL in cross-domain ADL classification and show their effectiveness in real-life smart-home datasets.
- we explore the usage of Transformer-based bidirectional language models for mid-level UTL in cross-domain sentiment classification (CDSC).

- we comprehensively analyze the performance of the two highest performing Transformer language models of XLNet and BERT in the context of CDSC and achieve new state-of-the-arts results with significant improvements over the previous approaches.
- we present a high-level UTL method for cross-lingual sentiment classification based on BERT and a novel non-task specific English-Japanese parallel sentiment corpus.
- we verify the effectiveness of the framework and the parallel corpus in comparison with state-of-the-arts in cross-lingual sentiment classification and other non-task specific Japanese-English parallel corpora.

1.4. Thesis outline

In Chapter 2, we provide an overview of background information in ADL recognition and sentiment analysis.

In Chapter 3, we present our work on cross-domain ADL recognition. We propose a novel low-level heuristic mapping method between heterogeneous sensor features from different smart-home datasets, based on their location, type, value, activity hour and normalized sensor event times in sliding windows. We adapt multilayer bidirectional LSTM network as a classifier and evaluate the performances in multiple experimentation scenarios.

In Chapter 4, we provide extensive analysis of Transformer-based bidirectional language models in the context of cross-domain sentiment analysis. We fine-tune and evaluate BERT and XLNet language models on Amazon sentiment datasets and compare the results with the state-of-the-arts.

In Chapter 5, we propose cross-lingual sentiment classification teacher-student framework based on BERT with multilanguage support and a novel English-Japanese sentiment corpus. We experiment on Japanese Amazon sentiment dataset and compare the approach with state-of-the-arts in cross-lingual sentiment classification and other non-task specific Japanese-English parallel corpora (JESC, Kyoto). Additionally, we experiment on Japanese Rakuten sentiment dataset to evaluate how our framework and the parallel corpus perform on different datasets.

In Chapter 6, we summarize our findings and provide potential directions of future works.

Chapter 2. Background

There are myriad of definitions for human behavior. According to the Encyclopedia Britannica, it is the potential and expressed capacity for physical, mental and social activity during the phases of human life[1]. The Nature journal generalizes it as the way humans act and interact based on and influenced by factors such as genetic make-up, culture and individual values and attitudes[2]. Wikipedia defines it as the array of every physical action and observable emotion associated with individuals, as well as the human race[3].

The most direct way for intelligent agent to record and classify human behavior is through direct observations from data collection tools such as camera, microphone, wearable sensors, , embedded sensors, and text processor that convert human behavior into machine consumable formats of video files, audio files, sensor event logs, and text documents.

For our thesis, we will explore human behavior classification techniques via two general directions: *physical behavior classification from embedded sensors* and *verbal behavior classification from text documents*.

2.1. Physical behavior classification from embedded sensors

We can distinguish human physical behaviors based on their complexities, from a physical *state* at a given time, e.g. pose and posture, to single ambulatory *action* composed of multiple states lasting short temporal duration, e.g. sitting and waving, and further to complex *activity* that consists of sequence of actions that take longer duration of time, e.g. cooking and playing a sport. In our work, we are mainly interested in *activities of daily living* (ADL), referring to daily routine self-care activities such as cooking, taking a bathe, dressing, cleaning, etc.

The Index of ADL was first introduced by a team of health professionals Katz *et al.* [1963] at the Benjamin Rose Hospital in Cleveland, Ohio as the standardized measure of an individual's cognitive and physical functional capabilities in studies of treatment and prognosis in the elderly and chronically ill. Cook and Krishnan, [2015] provides following examples of classes of ADL activities:

[1] https://www.britannica.com/topic/human-behavior
[2] https://www.nature.com/subjects/human-behaviour
[3] https://en.wikipedia.org/wiki/Human_behavior

Table 2.1. List of ADLs

Clean house	○ Dust, vacuum, sweep, mop ○ Make bet, change sheets ○ Scrub floor, toilet, surface, windows, ceiling fans ○ Clear table, wash dishes, dry dishes ○ Garden, wed, water plants ○ Gather trash, take out trash ○ Organizing items ○ Wash clothes, sort clothes, fold clothes, iron clothes
Meals	○ Prepare breakfast, lunch, dinner, snack ○ Set table ○ Eat breakfast, lunch, dinner, snack ○ Drink
Personal hygiene	○ Bathe, shower ○ Brush teeth, floss ○ Comb hair ○ Select outfit, dress ○ Groom ○ Shave, wash face, wash hands ○ Toilet ○ Trim nails, trim hair
Health maintenance	○ Take medicine, fill medicine dispenser, apply medicine
Sleep	○ Nighttime sleep ○ Sleep out of bed
Leisure	○ Play musical instrument ○ Read ○ Sew ○ Watch television, video, play video games
Social	○ Make phone call, talk on phone ○ Send text, read text, send email, read email ○ Write letters, cards ○ Entertain guests ○ Leave home, enter home
Work	○ Work at computer, work at desk, work at table

Human activity recognition (HAR) refers to the capacity of detecting human activity based on the data received from various sensors. HAR plays an important role in building human-centric intelligent agents that can correctly perceive human physical behaviors. With the advancement of modern sensor technologies, an activity recognition (AR) system can receive data about an individual's physical actions/activities and his/her environmental surroundings from varying degree of sensors such as wearable sensors, video cameras, smart tags, and sensors embedded in the environment.

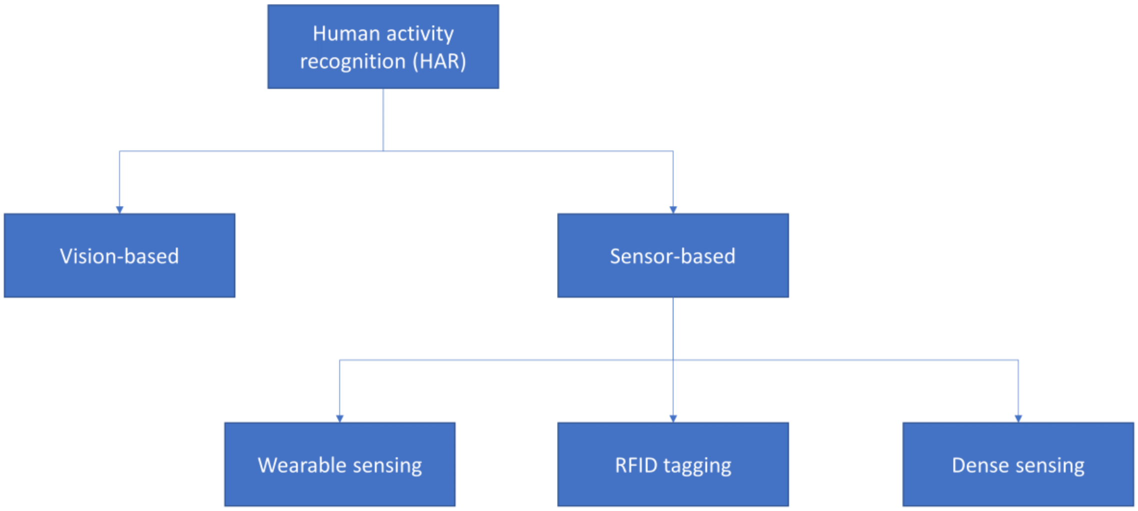


Figure 2.1. Data collection categories for HAR

Per Chen et al. [2012a], we can separate HAR methods into two general categories: vision-based and sensor-based. Along with the other fast advancing works of computer vision that are at the forefront of machine learning research, vision-based AR has made significant progress in terms of HAR. However, due to its need to continuously monitor a person's activity in order to have good performance, the issue of protecting individual's privacy becomes a main challenge of vision-based methods when applied on recognizing daily activities of individuals at home. Our work is aimed towards human behavior classification while preserving the comfort and privacy of an individual. Therefore, we shall not include vision-based methods in our discussion.

2.1.1. Sensor modalities and features

With the low cost of modern sensors, it has become viable to deploy comprehensive HAR systems only using sensors. Per Wang *et al.* [2019], we can classify sensor modalities of ADL recognition into two categories: portable sensors implanted in the objects that are carried or worn by the activity performer or stationary ambient sensors embedded in the environment. The portable sensors such as accelerometer, magnetometer, and gyroscope are the most common modality among the activity recognition works due to their wide availability on smart phones, bands, watches, etc. Among the proposed methods, as in Alsheikh *et al.* [2016], Lee *et al.* [2017], Chen and Xue, [2015], Khan *et al.* [2018], the accelerometer is the most widely utilized, with magnetometer and gyroscope are also used in combination with the accelerometer. Other portable sensors are used to detect human movements are ones attached a specific object that has some identifiable tag, e.g. RFID, attached to it. Object sensors are usually deployed in together with other types of sensors to detect complex activities, as in Yao *et al.* [2017], Ha and Choi, [2016].

Commonly used portable sensors include:

a) *Accelerometer* – detects acceleration changes in velocity over time along three-dimensional axis.
b) *Gyroscope* – detects change over time in angular position and usually deployed in tandem with accelerometer to provide more fine-grained action data representation.
c) *Magnetometer* – measures the strength of the magnetic field along three-dimensional axis. Used to detect the individual's orientation and proximity in relation to some magnetic objects.

Common stationary ambient sensors embedded in environments are:

a) *Passive Infrared (PIR) motion sensor* – detects the infrared light radiating from objects in its field of view through multiple PIR sensor slots. The motion sensor sends a movement message to some event logger when the difference in the detected radiation between the multiple slots is higher than some predefined threshold. PIR sensor will detect movement from any organic and non-organic object that emits infrared radiation, i.e. heat.
b) *Magnetic door sensor* – sends event message when its state changes. Change of state occurs when the electric circuit is either becomes complete or gets cut off with the reed switch connecting to and disconnecting from the magnet. Due to this feature, it is commonly used to detect opening and closing of doors, cabinets, drawers and windows.
c) *Light/temperature/humidity sensor* – sends either periodic environmental measurements or significant change of measurement value in time. Commonly bundled with other sensor types such as motion sensors or magnetic sensors.

d) *Vibration sensor* – detects vibration and tilt occurring to the object it was attached to. Used for detecting interaction with the object, but susceptible to unintentional actions and environmental changes.

e) *Pressure sensor* – measures the pressure imposed on the sensor. They placed in chairs, under floors and mats to observe the locations and changes in weight distribution of a person in the monitored space.

f) *Global position system (GPS) sensor* – reports the present location information in terms of latitude, longitude and height. The location can be triangulated either based on communications with GPS satellites or in the case of smart phones, based on time delay and signal strength when communicating with cell towers. Localization can also be calculated indoors using Bluetooth and WiFi signals.

g) *Radio-frequency identification (RFID) sensor* – detects proximity of a RFID tagged object to a RFID reader. The RFID tag does not require any power to operate and acts as a passive identifier when in close proximity of a reader. With multiple readers, movement of a tagged object or a person can be monitored.

For our ADL recognition task, we will focus on data collected from stationary ambient sensors that are embedded in smart home environment, specifically the PIR motion sensors, magnetic door sensors and light sensors.

In human activity recognition (HAR), data is first collected from the sensors embedded in the smart environment through a centralized event logger, and followed by data analytics stages such as data pre-processing, segmentation, feature extraction, and finally with classification of the activities with trained models. Pre-processing stage generates the representation of the raw sensor data. The segmentation stage divides the generated data representation into separate fixed-size or dynamic sized windows in order to extract informative features. Afterwards, from the segmented data set of low-level features are represented as vectors in order to be processed by the chosen machine learning algorithms with the objective of minimizing some classification errors.

Of all the different phases of human activity recognition framework, feature extraction is the most important stage due to the correlation between performances of activity recognition system and extraction of relevant and discriminative feature vectors, as shown in Nweke *et al.* [2018]. Due to the lack of generally accepted procedures for selecting appropriate features for any given dataset, many activity recognition works resort to heuristic feature extraction schemes. Depending on the expert knowledge, there are manual and automatic feature extraction techniques.

Because of the different modalities, locations and mobilities of the data collecting sensors, ADL recognition models are highly dependent on how well the low-level feature extraction represents the sensor data. We can categorize the low-level sensor data features into the following categories:

1) *Sequence features* – human activity sensor events are recorded as a sequence of time stamped data. Therefore, time becomes an essential feature, with different levels of granularity from nanoseconds to minutes, hours or day of the week and month. Since activities are performed over some duration of time, it is also important to represent a sensor event with respect to the time distance to the sensor events within some sliding window of predefined size.
2) *Discrete features* – many of the ambient sensors, e.g. PIR sensors and magnetic door sensors, have discrete values such as ‘ON’, ‘OFF’, ‘OPEN’, ‘CLOSE’ and we can use *bag-of-sensors* feature to represent these data. It is similar to the bag-of-words feature used in Natural Language Processing task where a document is represented as frequency of words that appear in the text, based on some predefined vocabulary. In activity recognition case, we count the frequency of sensor events that appear within a window of number of events or time duration.
3) *Statistical features* – besides the discrete value sensors, there are sensors such as accelerometers or light sensors that send its numeric value at each pre-determined time interval. There are various methods that extract statistical features from such time series numerical values with varying degree of coarseness, e.g. min-max, standard deviation, kurtosis, skewness, correlation, signal energy etc.
4) *Spectral features* – alternative to the above feature representations for sensor data, spectral view converts time series data into its frequency spectra using the Fourier Transform.
5) *Activity context features* – besides considering the current window events, previous windows can also be considered with varying degree of coarseness, from weighted concatenation of all of the previous window data with the current window to only retrieving the dominant event information.

2.1.2. ADL feature extraction

Human natural activities are usually performed in a continuous fluid process and without clear discernable gaps between different activities.

In order to detect human activities from time series sensor data, it is crucial to create partitions and subsequences that is discernable by a trained classification model. Otherwise, it is very difficult for a classifier to assign an activity label to a sensor event without knowing the context of the sensor activation. In the literature, there are two general directions of sensor data segmentation: event segmentation and sliding window.

In event segmentation, input sequence is partitioned into non-overlapping subsequences that represent a single activity. Example is shown as the explicit segmentation process which creates subsequences that can be associated with a single activity label. However, it is very challenging to correctly segment data into subsequences that correctly align with each activity's begin and end time. One approach to event segmentation is with supervised learning techniques where activity boundaries are learned from the provided pair of sensor data and the corresponding activity label using supervised machine learning algorithms.

Alternative approach to event segmentation is rule-based partitioning that holds an underlying assumption that each time an activity is performed, it will be centered around some set of sensor events and/or value range. The mapping between sets of sensor events and value ranges to activity labels can be either predefined using domain knowledge or learned from training data.

Alternative to learning or predefining the single activity subsequence, sliding window partitions the input sequence into a separate window of overlapping subsequences and the object is to assign label to the last event in each window. A sliding window can be partitioned in following ways:

1) Time-based: divides the input sequence into uniform time intervals based on time distance from the last event in the window. This approach is most appropriate for recognizing fine-grained activity or action using sensors that send its state numeric value at constant time rate, e.g. accelerometers, gyroscopes. The appropriate length of time can be predefined or learned from the training data.
2) Size-based: partitions the input sequence into overlapping uniform number of event windows. This approach is more suited for segmenting data from sensors that send event

messages only when triggered by change of state. As with the time-based sliding window, the number of events can also be predefined by domain knowledge or learned from the training data. Both in time-based and size-based approaches the window is represented as bag-of-sensors.

3) Weight-based: introduces time-based weights to the sensor events within the size-based sliding window. This way the sensor events that occurred much earlier shall have less influence on classifying the current sensor event. Here the window is represented as weighted sum of occurrences of sensor events, i.e. weighted bag-of-sensors.
4) Dynamic: the size or the time interval of the window is heuristically or probabilistically derived for each sensor event from a set of potential time durations or event numbers.

2.1.3. ADL recognition methods

In the literature, there are two main directions in recognizing ADLs: knowledge-driven and data-driven approaches, as categorized in Bakar *et al.* [2016].

In knowledge-driven approaches, sensor events and activity labels are modeled either using structured logics or ontologies using prior domain knowledge. The domain knowledge is first acquired in order to define the activities. Afterwards, logical representations of the activities are formalized based on the gathered domain knowledge. Logic-based approaches convert ADL features into formal logical structures that are processed using some knowledge-based inferences to identify activities. Since logical approaches do not need training data, they can re-used in multiple different datasets and scenarios as proposed by Ferilli and Esposito [2013], and Rafferty *et al.* [2017].

Utilization of ontologies improves on logical methods with more flexible models by representing sensor events and activities with interdependent properties. In the work proposed by [Chen et al. 2012a], knowledge-driven approach based on ontological modeling and semantic reasoning is proposed. The notable component of the approach is its unified ontological modelling and representation for both sensor data and activities, which allows reusability of semantic reasoning for activity recognition in different datasets. The proposed method was demonstrated on data collected from three participants individually performing eight activities with different permutations of sequence and the data is collected from contact sensors, motion sensors, tilt sensors and pressure sensors. Helaoui *et al.* [2013] propose probabilistic description logic method for multi-level activity recognition framework that hierarchically decompose complex activities into their simplest atomic components using OWL 2 ontological language. Soulas *et al.* [2015] propose an Extended Episode Discovery algorithm to search for regular activity patterns, highlighting the periodicity and variability of each discovered activity pattern. However, their experimental data is quite limited to fully validate their approach. Riboni *et al.* [2016] propose an unsupervised ADL recognition method based on ontological reasoning, where they derive ontologies to describe the smart home environment and the semantics of interleaved activities. The ontologies formally define the semantic conditions of the sensor events during the execution of a specific activity in the given environment. They identify activity instances using Markov Logic Network (MLN) probabilistic reasoning that predicts the start and end time of occurred activities from extracted semantic correlations between triggered sensor events. They experiment on CASAS dataset covering interleaved ADLs of multiple subjects performing eight simple activities (*fill medication dispenser*, *watch DVD*, *water plants*, *answer the phone*, *prepare birthday card*, *prepare soup*, *clean*, and *choose outfit*) in a smart home laboratory. Compared to supervised learning methods, this approach does not need acquisition labeled training data and identify activity instances using its ontology model. However, it requires manually modeling the semantic sensor events that must occur for each type of activity. In Gayathri *et al.* [2017], ontology-based activity recognition is augmented with probabilistic reasoning through Markov Logic Network (MLN) applies weights to the first order rules. Experiment is also conducted on CASAS dataset with eight simple activities.

Advantages of knowledge-driven methods are the semantic clarity, reusability, and the cold-start capability, i.e. immediate use with no training data. However, due to their logical formalism, they have static nature and have difficulties when encountering noisy sensor data and temporal information.

Data-driven approaches, on the other hand, model the human activities directly from the provided training datasets using machine learning or data mining techniques and much more suited in dealing with sensor data noise, uncertainty and temporal parameters. We can further divide the data-driven approaches into generative and discriminative models.

Generative data-driven approaches model the underlying data distribution of each activity class data by learning joint probability distribution. Kabir *et al.* [2016] propose a two-layer HMM to represent the mapping between low-level sensor data and high-level activity based on the binary sensor data. The first layer uses location data to predict coarse-grain activity class and the second layer uses sequence data to further narrow down the activity label. The show the effectiveness of their method on Van Kasteren *et al.* [2008] datasets in comparison to NB, CRF, and HMM models. In Oukrich *et al.* [2018], DBN algorithm uses ontological features to recognize three categories of multiple resident activities within a smart home: single resident performing an activity individually, multiple residents performing an activity together and multiple residents performing separate activities. They experiment on CASAS multiple resident dataset and their model outperforms SVM and ANN models. The research work in Donaj and Maučec, [2019] presents a HMM-based model extended to a second-order Markov chain model of activity sequences to recognize long-term dependency in the model. They also introduce an activity transition cost to negate the propensity of HMM model to make many transitions. They experiment on CASAS dataset and show that the combination of activity transition cost and Markov chain models improves the performance when compared to regular HMM and NB models.

Discriminative data-driven approaches model the decision boundary between the different activity class data by learning conditional probability distribution. Singh *et al.* [2017] perform activity recognition using 1D convolutional neural model and compare it with activity recognition methods using LSTM recurrent neural network, Hidden Markov Model (HMM), Hidden Semi-Markov Model (HSMM), Naïve Bayes, and Conditional Random Fields (CRF). Their experimental results on Van Kasteren *et al.* [2008] datasets show that deep learning models of 1D-CNN and LSTM have much better prediction performance than probabilistic models such as HMM, HSM, Naïve Bayes and CRF. Between the deep models, LSTM seems to perform slightly better than CNN but slower in terms of training time.

Wan *et al.* [2018] propose a cumulative overlapped fixed-size sliding windowing approach for real-time activity recognition. It looks at each given sample with multiple different size windows, e.g. {10, 30, 60, 120} seconds windows, and the classification is performed on each of these different window views. Additionally, the activities are divided into instantaneous, where activities are identified based on pre-defined close coupling with individual sensors, and durational, where logistic regression is utilized to learn and predict the activities. Experimentation is done on CASAS Aruba dataset where two types of instantaneous activities are predicted (Leave_Home and

Enter_Home), and six general types of durational activities are predicted (Meal preparation, Relax, Sleeping, Work, Housekeeping).

Sukor *et al.* [2019] propose hybrid approach that combines knowledge-driven and data-driven methods. Initially, an activity model is created with knowledge-driven reasoning. The model is then further trained using data-driven method to produce a dynamic activity model that accommodates to users' individual actions. This approach has been evaluated using a publicly available Kasteren dataset and the experimental results show the learned activity model yields significantly higher recognition rates compared to the initial knowledge-driven activity model.

2.2. Verbal behavior classification from text documents

Natural Language Processing (NLP) is one of the leading machine learning research areas that deals with teaching machines to understand natural human language. It comprises of varied subfields including language modelling, speech recognition, named entity recognition, part-of-speech tagging, and many others. In our work, we will focus on sentiment analysis, which is a subfield of NLP. The term sentiment analysis has been applied to varying machine learning tasks, such as film or product review opinion extraction, determining the polarity of news document, or identify people attitude towards political topics, etc.

The main objective in **sentiment analysis**, or *opinion mining*, is to identify how sentiments are expressed in texts and whether the expression indicate positive or negative opinions towards subject [Nasukawa and Yi, 2003].

The most widely researched task among the sentiment analysis is the document sentiment classification task. It simplifies the sentiment analysis problem into a single opinion holder's sentiment expression towards a single entity [Liu, 2010]. We can view a sentiment text document as a tuple of:

- *Entity*: a product, event, person, organization, etc. towards which a sentiment or an opinion is expressed. An entity can be further decomposed to set of aspects that comprise the entity, which itself consists of components and attributes. For example, a computer entity's aspects include components of cpu, memory, screen, battery, etc. and their corresponding attributes of speed, capacity, size, longevity, etc.

- *Opinion holder*: the person expressing the sentiments. Opinion holders are either explicitly indicated as in news documents or implicitly assumed to be the text document author, as in user reviews and social media posts.
- *Sentiment*: opinion holder's verbal expression of personal attitude towards the entity. We can view verbal expressions as a part of person's verbal behavior. Based on the task, sentiments can have nominal values like negative, positive, and neutral, or numeric values that indicate the intensity of the attitude as in strongly negative or slightly positive, etc.

As described by Alessia *et al.* [2015], in general, the sentiment analysis is comprised of the following four steps:

- *data collection*: sentiment text document is collected from user generated contents in social networks, blog posts, and online reviews. Due to the sheer size of contents, it is highly costly to manually analyze each of these data.
- *text preparation*: since the raw text data might have non-relevant contents, some pre-processing is performed to clean the text document.
- *sentiment detection*: the pre-processed text document is examined for subjective expressions that can be utilized for sentiment classification.
- *sentiment classification*: the retained subjective expressions are classified according to the given sentiment classes.

Based on the different principles of prediction algorithms, Liu *et al.* [2019] categorize sentiment classification methods into three main techniques, namely traditional lexicon-based, deep learning, and transfer learning.

2.2.1. Traditional lexicon-based sentiment classification

Lexicon-based approaches in Hu and Liu, [2004], Ding *et al.* [2008], Taboada *et al.* [2011] predict the sentiment polarity of text document based on some external evidences such as dictionaries of words annotated with their sentiment orientation and/or linguistic conventions of natural language expressions. Work by Thelwall *et al.* [2010] propose SentiStrength sentiment classification algorithm that uses a dictionary of sentiment words with associated strength measures and a range of recognizable textual patterns of expressing sentiment. Saif *et al.* [2016] propose a lexicon-based method, SentiCircles, that update the semantics of words with pre-assigned sentiment polarities and strengths based on their co-occurrences in different contexts. Another method in Bravo-Marquez *et al.* [2016] expands existing pre-assigned sentiment lexicons with information from automatically annotated tweets in a supervised manner. The expanded lexicon is comprised of part-of-speech (POS) entries with a probability distribution for each sentiment polarity class. The biggest advantage of lexicon-based methods is they do not require training data, but limited by the range of words in the lexicon and the fixed semantic scores assigned to the words.

2.2.2. Deep learning sentiment classification

In recent years, machine learning methods have become the widely used approach in sentiment classification. As with other machine learning classification tasks, supervised learning has been the most researched direction in document sentiment classification. Some of the methods employ emoticons as labels for sentiment text. In Go *et al.* [2009], binary sentiment classification is performed on twitter messages by training a classifier using distant supervision. Linear-kernel SVM based approach is proposed by Kiritchenko *et al.* [2014] that detects the overall sentiment of short informal textual messages and the sentiment of a word or a phrase within the messages. The sentiment features are derived from lexicons that were automatically generated from tweets with hashtags and emoticons.

Ensemble systems have been employed to create better performing models by combining multiple classifiers. Lin and Kolcz, [2012] present an integration of machine learning tools into Twitter's Hadoop-based, Pig-centric analytics platform that uses ensemble of machine learning algorithms to provide predictive sentiment analytics. Also, Da Silva *et al.* [2014] propose an ensemble-based approach that use combination of lexicons, bag-of-words, emoticons, and feature hashing strategies to represent the sentiment features of tweets and the classifiers are an ensemble of Random Forest, SVM, and logistic regression.

In most of the latest NLP works, including sentiment classification, word embeddings have become the standard way to derive low-level feature representations of text document. Word embeddings are low-dimensional representations that maps words with similar meaning closer to another. Word2Vec in Mikolov *et al.* [2013] and GloVe in Pennington *et al.* [2014] are the most widely used pre-trained word embedding models. The simplest way to use word-embeddings for sentiment classification is to calculate the average of the word vectors in the given document, and use it as the feature representation to train a classification model, as in Castellucci *et al.* [2015].

With the advancement of deep learning in CV and NLP, most of the recent sentiment classification approaches have been based on application of deep learning techniques, specifically Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

Kim *et al.* [2014] use CNN in combination with word embeddings to classify sentiment at sentence level and show that a simple CNN model combined with pre-trained word vectors performs well in various benchmarks by fine-tuning only few hyperparameters.

Work in Jonhnson and Zhang [2015] exploits word order of text data to predict sentiment classes by directly learning embedding of small text regions from the high-dimensional text data using CNN. Additionally, the combination of multiple convolutional layers for better prediction performance is also explored in the work.

Conneau *et al.* [2017] propose a very deep CNN models, up to 29 convolutional layers, that combines VGGNet and the ResNet artificial networks. Their proposed model for text processing operates directly at the character level of the text document using only few convolutions and pooling operations.

Due to its inherent recurrent sequential data processing nature that can theoretically process any length of sequence, RNN based methods are more widely utilized for NLP tasks than CNN based approaches. Socher *et al.* [2013] use RNN model to learn the sentiment for different fragments of a document, from a word to phrases, and up to sentences. The RNN model uses sentiment annotated treebank[4] of parsed sentences to learn feature representations of words and phrases. The proposed model is limited by its use of parsed sentences to be applied to other datasets that include phrases not included in the treebank.

Tang *et al.* [2015] introduce a RNN model that learns sentence representation from word representations using CNN or LSTM. Afterwards, gated recurrent units (GRU) neural network encodes the semantics of sentences and their relationships in document representations that are used as features for classifying the sentiment label.

Furthering their work in Jonhnson and Zhang, [2015] that deals with using CNN for text region embedding, Jonhnson and Zhang, [2016] explore text region embeddings with LSTM due to its

[4] http://nlp.stanford.edu/sentiment/treebank.html

ability to embed regions of variable size, as oppose to CNN that needs fixed region size. They show that region embedding using LSTM and convolutional layers trained on unlabeled data produce the best results.

There are also works that explore combination of methods and techniques other than CNN or RNN. [Akhtar et al. 2016] propose hybrid deep learning architecture for sentiment classification in resource-poor languages, such as Hindi. Learned embedded vectors output by the CNN are augmented to a set of optimized features selected through a multi-objective optimization framework. SVM is trained on the final the augmented output for sentiment classification.

[Yang et al. 2016] propose a hierarchical attention network that has separate attention mechanism for word and sentence-level context. It constructs a document representation by building aggregating representations of sentences. The two-level attention mechanisms help the model to pay varying degree of attention to individual words and sentences when constructing the document representation.

2.2.3. Transfer learning sentiment classification

In accordance with categorization of [Pan and Yang, 2010], the transfer learning methods for sentiment analysis can be divided based on what is being transferred, namely transfer learning of instance, feature and parameter.

Instance transfer

In instance transfer learning methods, some of the source domain data are reused, either directly or re-weighted, together with few labeled data in the target domain.

[Dai et al. 2007] propose a TrAdaBoost algorithm, an extension of AdaBoost algorithm, that assumes both source and target domain data have same features and labels, but the distribution is different. It iteratively re-weights the source domain data to increase the effect of useful data that can contribute to the target domain and decrease the effect of ‘bad’ data that might not be useful in training classifier for target domain.

[Chakraborty et al. 2012] propose Conditional Probability based Multi-source Domain Adaptation (CP-MDA) framework that labels target domain unlabeled data using a weighting scheme based on similarity measurement in conditional probabilities between source and target domains. They also propose a second transfer learning framework, Two Stage Weighting Framework for Multi-source Domain Adaptation (2SW-MDA) that computes the weights for multi-source data samples to reduce both marginal and conditional probability differences between the domains. Marginal distribution difference is computed using Maximum Mean Discrepancy (MMD).

Gui *et al.* [2015] propose transfer learning method for detecting negative transfers in cross-lingual sentiment classification. It iteratively detects and removes bad samples by identifying high quality samples in the target domain unlabeled data.

Feature transfer

Feature transfer learning methods tries to derive good common feature representations for both source and target domains that can reduce domain divergence and classification loss.

Blitzer *et al.* [2006] propose Structural Correspondence Learning (SCL) to automatically induce correspondences among features from different domains by modeling their correlations with pivot and non-pivot features. The approach is limited by its dependence on the suitability of the derived latent space and the number of auxiliary learning tasks.

Pan *et al.* [2010] introduce Spectral Feature Alignment (SFA) algorithm to align domain-specific words from different domains into unified clusters using pivot domain-independent words and model the co-occurrence patterns between domain-specific and domain-independent words. Their key idea is that domain-specific words will be aligned together with greater probability if they are connected to more common domain-independent words.

Xia *et al.* [2013] propose feature ensemble plus ample selection (SS-FE) joint domain adaptation method for both labeling adaptation and instance adaptation. A labeling function is learned in an ensemble feature re-weighting scheme. Additionally, a PCA-based sample selection method helps the feature ensemble for instance adaptation.

Zhou *et al.* [2014] propose a multi-class heterogeneous domain adaptation method to reconstruct a sparse and class-invariant feature transformation matrix in order to map the weight vector of source domain binary classifiers to the target domain. They show that the sparse feature mapping can be learned if a sufficient number of classifiers are provided.

Parameter transfer

Parameter transfer learning methods hold assumption that individual models can share parameters or some hyperparameter distributions due to their closely related tasks.

Glorot *et al.* [2011] propose to use a Stacked Denoising Autoencoder (SDA) to extract high-level features in unsupervised fashion from the text documents of all the available domains. In SDA, auto-encoders are trained to minimize a reconstruction loss on the input. Once an auto-encoder has been trained, another auto-encoder can be stacked on top of the trained one by training the second auto-encoder using output of the trained one. Afterwards, a linear SVM is trained on the transformed labeled data for sentiment classification.

Work in Chen *et al.* [2012] introduces marginalized SDA that aims improve the SDA method by addressing the high computational cost and the lack of scalability to high-dimensional features. In contrast to SDAs, the marginalized SDA does not require stochastic gradient descent or other optimization algorithms to learn parameters, but rather computed in closed-form.

Dai and Le [2015] first introduce two pre-training algorithms to be used for a later supervised sequence learning task. The first algorithm is predicting a next token of a given sequence and the second is to reconstruct the given sequence using autoencoder. They show that pre-training algorithms help stabilize the learning in recurrent networks such as LSTM.

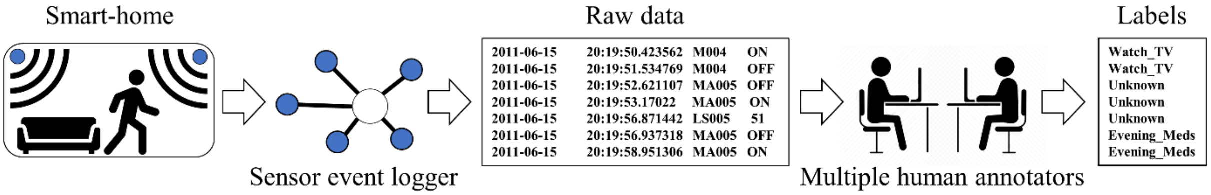
Peters *et al.* [2018] propose a new type of deep contextualized word representations that are learned functions of the internal states of a pre-trained deep bidirectional language model on the entire input sentence. It uses bidirectional LSTM to create the language models by training on an English corpus with about 30 million sentences. This pre-trained language model provides a common low-level representation of texts that can be passed to additional neural network, e.g. RNN or CNN, for sentiment classification.

Howard and Ruder [2018] introduce Universal Language Model Fine-Tuning (ULMFiT) transfer learning method for NLP tasks. They propose 3-layer LSTM architecture, with same hyperparameters, that can perform better than task-specific engineered models on various NLP tasks using novel techniques for retaining prior knowledge and avoiding catastrophic forgetting during fine-tuning for downstream tasks.

Chapter 3. Low-level common feature extraction for cross-domain ADL recognition

3.1. Introduction

Most current ADL recognition approaches adopt supervised learning methods because the way a person performs daily routines at home differs greatly depending on the person’s moods, habits and the layout of the furniture and appliances. However, acquiring annotated dataset for a single home from raw ambient sensor events is very expensive due to the need to have an annotator who are closely familiar the layout and the daily routine of the occupant in order to annotate the raw time-series sensor event data. Additionally, we need to have multiple annotators examining single dataset to reduce potential human errors, which multiplies the label acquisition cost. An alternative is to have either video cameras or in-house observer continuously monitoring the occupant’s daily home routines, which heavily infringes on the occupant’s privacy and comfort. Figure 3.1 shows the general process for ADL data collection from embedded sensors.



```
2011-06-15    20:19:50.423562  M004    ON
2011-06-15    20:19:51.534769  M004    OFF
2011-06-15    20:19:52.621107  MA005   OFF
2011-06-15    20:19:53.17022   MA005   ON
2011-06-15    20:19:56.871442  LS005   51
2011-06-15    20:19:56.937318  MA005   OFF
2011-06-15    20:19:58.951306  MA005   ON
```

```
Watch_TV
Watch_TV
Unknown
Unknown
Unknown
Evening_Meds
Evening_Meds
```

Figure 3.1. ADL data collection from embedded sensors.

In the literature, many activity recognition approaches have been proposed and most of them deal with data collected from video cameras, as in Lan *et al.* [2017], Chakraborty *et al.* [2017], Zhang *et al.* [2019] or wearable sensors, as in Sztyler *et al.* [2017], Lee *et al.* [2017]. However, the problem with video cameras and wearable sensors are the intrusive nature of data collection methods, in addition to the privacy issues. Ambient sensors, on the other hand, are used to capture the interaction between humans and the environment in a non-intrusive way. The sensors are embedded in users' smart environment and activity is detected through changes in the environment. In comparison to the video and wearable sensor based approaches, such as De *et al.* [2015], Liu *et al.* [2016], ali Hamad *et al.* [2019], Tahir *et al.* [2019], much fewer methods have been proposed in recognizing ADL using ambient sensors. However, these are still supervised learning techniques and the exploration of applying unsupervised transfer learning in ADL recognition with ambient sensor data have been very limited.

In this chapter we propose a novel multi-level transfer learning method for cross-domain ADL recognition that utilizes heuristic mappings between the heterogeneous features. Our contributions are as follows:

- we propose novel low-level heuristic mappings between heterogeneous sensor features from different smart-home datasets, based on their location, type, value, activity hour and normalized sensor event times in sliding windows.
- We propose a cross-domain ADL classification method using multilayer bidirectional LSTM networks.
- we evaluate the effectiveness of our proposed method via multiple experiment scenarios on CASAS single resident real-life smart-home datasets and compare the results with other ADL transfer learning methods.

3.2. Related works

Feuz and Cook [2014] propose three heterogeneous transfer learning methods of Feature-Space Remapping (FSR), Genetic Algorithm for Feature-Space Remapping (GAFSR), and Greedy Search for Feature-Space Remapping (GrFSR) that transforms target domain data into source domain feature space. All of the tree proposed approaches use some labelled target data to infer relations to the source domain.

Also, Feuz and Cook [2017] propose a multi-view supervised transfer learning algorithms is introduced, which transfers knowledge between heterogeneous activity learning domains. The domains differ in their sensor modalities, i.e. one is smart home sensor based (source view) and the other is smart phone sensor based (target view). In their proposed Personalized ECOSystem, the source view initially provides labels for a few samples that both source and target views observe with their different sensors. Afterwards, the system adopts iterative co-training method to the benefit of both views. They experiment on three activity recognition datasets with each containing activity data of multiple participants acquired from multiple heterogeneous sensor types. The results show the high dependence on good selection of source view.

Khan and Roy [2018] propose UnTran activity recognition model that utilizes source domains' pre-trained autoencoder enabled activity model that transfers two layers of this network to generate a common feature space for both source and target domain activities. Their key novelty is it leverages the performance of existing source and target domain activity recognition models by learning the variability of the activity patterns using few labeled target domain data. They evaluate their approach on three datasets with wearable accelerometer sensor data and compare the results with transfer learning methods of Transfer Component Analysis (TCA) by[Pan *et al.* [2011] and Joint Distribution Adaptation (JDA) in Long *et al.* [2013].

Wemlinger and Holder [2018] propose Semantic Cross-Environment Activity Recognition (SCEAR) system that projects raw sensor activities from different smart environments into common semantic feature space between different domains and transfers data-driven models across environments. SCEAR is composed of an ontology component and a set of sensor reasoners. The ontology defines the sensors' physical location using the authors' previously proposed CASAS Ontology for Smart Environments (COSE) in Wemlinger and Holder [2011], for example, the class *Toilet* is a subclass of the concept *Bathroom Objects*. Each sensor reasoner accepts as input a filtered subset of events and outputs estimates regarding the likelihood that a resident is interacting with an area or object. They experiment on 20 smart home datasets from Alemdar *et al.* [2013] and Cook *et al.* [2012]. Individual models are trained for each unique triple of (*source*, *target*, *activity*) using machine learning algorithms of Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine.

Shang *et al.* [2019] proposes an unsupervised behavior identification algorithm that uses unsupervised clustering method to identify behaviors of single elderly people. The activities are clustered based on statistically observed Event Shift and Histogram Shape Similarity Properties of the data. The clusters are assigned labels by measuring similarity with predefined activity patterns. The evaluation is performed on WMNL2016 dataset collected from 20 state-change sensors and smart devices deployed at an apartment with 5 regions, including bathroom, foyer, living room, bedroom, kitchen and dining room. A volunteer lived in the apartment for more than 5 months, while providing ground truth labels via a smartphone microphone for 9 activity classes of "GoToToilet", "TakeShower", "Outdoor", "Watch_TV", "TakeMedicine", "MeasureBloodPressure", "GetUp", "Sleep", and "CookEat".

3.3. Heuristic common sensor feature extraction for cross-domain ADL recognition

To solve our heterogeneous ADL transfer learning problem, first we need create a point-of-reference between the domains. In computer vision (CV), the general concept of an image provides some low-level contextual mapping between different domains. Provided that the image's pixel locations and values are not displaced or disarranged, we can have some assumptions about the continuation of pixels to represent an object(s) and the orientation of the image. With the vanguard advancement of computer vision in machine learning research, there exist well established convolutional models that are pre-trained on various large scale image datasets, such as ImageNet, that are capable of providing comparable low-level feature representations to images without any labels, which can be thought of as low-level contextual mapping. Similarly in NLP tasks, there are empirically proven word-embeddings that also act as low-level mapping of input text document to a common latent space.

However, in ADL recognition tasks, there are currently no established or empirically proven low-level contextual mappings available. Due to the variations in the architectural structure of homes, data collection configurations, sensor types, sensor deployment locations, unclear ordering of sensor features, lack of large scale ADL datasets and the wide variations in how individuals perform their daily activities, it becomes very challenging to apply unsupervised transfer learning and domain adaptation methods on heterogeneous ADL datasets without any feature mapping. Therefore, it becomes necessary to create some low-level feature mappings between heterogeneous sensor feature spaces using a heuristic approach. We propose to project source and target domain sensor data into a coarse-grained common feature space with contextual mappings based on sensor location, value range, type, daily activity hour partition and contextual time association. With exception of location mapping, all the contextual mappings strategies are done so automatically.

In Figure 3.2, we illustrate the proposed method for cross-domain ADL recognition with common low-level feature extraction. Using the derived low-level sensor feature representations, we extract high-level abstract features with LSTM networks and map the extracted feature to labels for the source domain data.

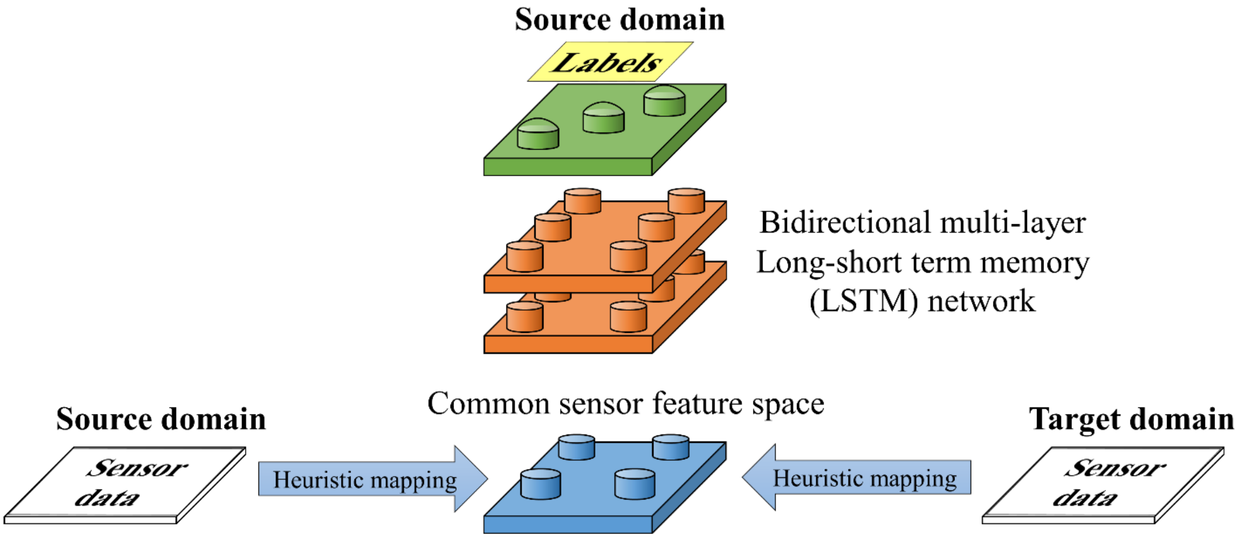


Figure 3.2. Proposed cross-domain ADL recognition with common low-level feature representation.

3.3.1. Sensor location mapping

We propose a manually mapping of sensors with locations tags based on the deployed physical location of the sensor with respect to the house outline. For better transferability between datasets, we select single bedroom apartment with one bedroom, a living room, a kitchen area, a bathroom and front door area as our general location template. We propose following tags for location mapping based on the sensor's physical location:

Table 3.1. Sensor location tags and the corresponding physical areas

Location tag	*Represented physical area*
"BB"	within the bedroom area
"BF"	between bedroom and front door area
"FF"	within the front door area
"KK"	within the kitchen area
"KL"	between kitchen and living room area
"LL"	within the living room area
"LF"	between the living room and front door area
"RF"	between restroom (bathroom) and front door area
"RR"	within the restroom (bathroom) area

Sensor location tags are manually extracted from the sensors' location map provided with the dataset.

3.3.2. Sensor type mapping

Since our work focuses on ADL recognition using ambient sensors, we consider following sensor type tags for contextual mapping between different dataset sensors:

Table 3.2. Sensor types

Sensor type tag	*Description*
“M0”	Binary PIR motion detection sensor
“MA”	wide area PIR motion detection sensor with binary values
“LL”	light sensor with periodic numeric measurements
“L0”	light sensor with binary values
“D0”	door sensor with binary values

The sensor type tags are automatically extracted from the raw sensor data.

3.3.3. Sensor value mapping

Most of the ambient sensors have binary values which are straight forward and easily represented as bag of sensors. However other environmental sensors, e.g. “LL” light sensors, have numeric measurement values and their value fluctuations might not be directly comparable. For example, a relatively high measurement value for a light sensor located in a dimly lit area might seem as a low value for a sensor installed in a brightly lit area. Therefore, we propose more coarse-grained sensor mapping strategy where each non-binary sensor value is normalized to “High”, “Mid”, “Low” value ranges that are determined individually per sensor based on their respective minimum and maximum value ranges detected in the given dataset.

3.3.4. Daily activity partition hour mapping

Representing time correctly plays a crucial role in classifying ADLs because an individual’s daily activities, knowingly or unconsciously, often follow some time-based routines. Since everybody’s daily routines vary greatly, we propose to represent the sensor event time with following two features: *the day of the week* and *the activity hour partition* of the day. The day of the week simply represents one of the days from Monday through Sunday.

However, to represent the hour of the day, rather just assigning values between 0 and 23, we propose to separate the hours into partitions based on the activities performed. Specifically, we divide the day into four general partitions (Night, Morning, Day, Evening) and another four intermediary partitions that act as transition phase between the four general partitions. We then convert the standard hour of day (between 0 and 23) feature of a sensor event into occupant-specific activity partition hour value.

Table 3.3. Daily activity partitions and the ADLs occuring with the partition

Daily activity partition	*Common ADLs within the partition*
Night (‘NN’)	Sleep associated activities (‘Sleeping’).
Night-to-Morning (‘NM’)	Transition activity partition between sleeping and morning activities.
Morning (‘MM’)	Morning activities (‘Cook_Breakfast’, ‘Eat_Breakfast’, ‘Wash_Breakfast_Dishes’, ‘Morning_Meds’).
Morning-to-Day (‘MD’)	Transition time period between morning activities to afternoon activities.
Day (‘DD’)	Activities performed during the noon and afternoon (‘Cook_Lunch’, Eat_Lunch’, ‘Wash_Lunch_Dishes’).
Day-to-Evening (‘DE’)	Transition activity partition between afternoon and evening
Evening (‘EE’)	Evening activities (‘Cook_Dinner’, ‘Eat_Dinner’, ‘Wash_Dinner_Dishes’, ‘Evening_Meds’) and (‘Watch_TV’, ‘Relax’).
Evening-to-Night (‘EN’)	Transition time period between evening activities to sleeping.

Since individuals have different daily routines, the daily activity partitions for each single resident smart home is determined separately based on the occupant’s activity patterns.

To determine the activity partitions for a smart home dataset, we first extract sensor events that have the same activity labels as our activity partition classes. We map the activity labels into one of the four partition labels (‘NN’, MM’, ‘DD’, ‘EE’) based on which partition the activity label belongs to. Then, given a sensor event dataset containing only the standard hour of the day feature $\mathcal{X} \in \{0, \dots, 23\}$ and the corresponding activity partition labels $\mathcal{Y} \in \{1, \dots, 4\}$, activity partitions hours are determined by training a Multinomial Naïve Bayes classifier to estimate the fraction of times activity partition $y_j \in \mathcal{Y}$ appears in the hour $x_i \in \mathcal{X}$:

$$P(y_j|x_i) = \frac{count(y_j, x_i)}{\sum_{y \in \mathcal{Y}} count(y, x_i)}$$

Once the model is trained, we map the hour of day feature into activity partition hour as follows:

$$x_i = \begin{cases} y_j, if\ P\left(y_j \middle| x_i\right) > \ 0.95. \\ y_{jk}, otherwise. \end{cases}$$

where y_{jk} is the transition activity partition with $P\left(y_j \middle| x_i\right) > \ 0.05$ and $P(y_k|x_i) > \ 0.05$, i.e. for given a hour of day, if there are multiple activity partitions of y_j and y_k that have frequency probability of over 5%, then the given hour of day shall be assigned the transition activity partition, such as 'NM', 'MD', etc.

Once all the sensor times are mapped to activity partitions, we can observe that multiple hours will have the same partition labels, e.g. hours 12, 13, 14 mapped to the same 'DD' partition. To introduce more granularity, we assign sequential numbering to the activity partition hours, e.g. hours 12, 13, 14 mapped to the 'DD1', 'DD2', 'DD3'. Henceforth, all of the smart-home datasets' hour of the day feature is mapped to activity partition hour. For example, one dataset's day can be partitioned as follows with the initial hour value in brackets:

Table 3.4. Example of a partitioned 24 hour period

NN1 (00)	NN2 (01)	NN3 (02)	NN4 (03)	NN5 (04)	NM1 (05)	NM2 (06)	MM1 (07)
MM2 (08)	MM3 (09)	MM4 (10)	MD1 (11)	DD1 (12)	DD2 (13)	DD3 (14)	DD4 (15)
DE1 (16)	EE1 (17)	EE2 (18)	EE3 (19)	EE4 (20)	EE5 (21)	EN1 (22)	EN2 (23)

For our unsupervised transfer learning scenario, we can map the source domains' time into the activity partition hours using the above strategy via the available labeled data. However, since we will not have any labels for the target domain, we will utilize simple majority voting mechanism among the source domains to determine the target domain's activity partitions.

3.3.5. Contextual time association within sliding window

Because any sensor can be a part of different daily activity sequences, a single sensor event is not sufficient to determine the performing activity's class label. Thus, window segmentation is commonly utilized approach for time series sensor data to perform classification tasks. We propose a window segmentation of fixed event size, but with normalized time value for each sensor within the window.

Since our proposed ADL classification method is in offline mode, we can look ahead of the current event to create the window. Based on this assumption, we propose a sliding window comprising of fixed number of events preceding and succeeding the current sensor event. However, the contextual association between the current sensor event and all the other sensor events within the window can differ depending on how far or how close time-wise those events are to the current one.

In addition to the above mentioned *day of the week* and *activity partition hour* features, for each sensor event we have the event time represented in Unix timestamp, i.e. seconds elapsed since January 1st of 1970. To represent the time-dependent contextual association between the current sensor event and the other events within the sliding window, we will normalize the Unix timestamps of the sensor events to be a continuous value between 0 and 1, where 1 indicating the closest timestamp to the current event and 0 indicating the furthest timestamp. Specifically, for each preceding timestamp $t_i^{\leftarrow}$ within the window, we normalize it to be between the timestamps of the window's first sensor event $t^{\leftarrow}$ and the current sensor event t_c:

$$t_i^{\leftarrow} = \frac{(t_i^{\leftarrow} - t^{\leftarrow})}{(t_c - t^{\leftarrow})}$$

We also similarly normalize each succeeding timestamp $t_i^{\rightarrow}$ to be between the current sensor event timestamp t_c and the window's last sensor event timestamp $t^{\rightarrow}$:

$$t_i^{\rightarrow} = \frac{(t^{\rightarrow} - t_i^{\rightarrow})}{(t^{\rightarrow} - t_c)}$$

3.3.6. ADL classification with LSTM networks

We use Long-Short Term Memory (LSTM) networks, a variation of recurrent neural networks, as our proposed ADL recognition classifier. Recurrent neural networks (RNN) are a type of artificial neural networks that are most suitable for processing sequential data such as text, speech, time-based sensors activation sequence, speech etc. With the activation state saved in a hidden state, RNN recursively processes the input samples one by one, with the hidden state for the previous sample is used for computing the hidden state for the current sequence sample and the current hidden state used for the next sample and so on. The hidden state for the previous input is summed with the current input sample after both multiplied with the weight matrices. Even though RNN can theoretically process any length of sequence data, Bengio *et al.* [1994] empirically shows it is unable to capture long-term dependencies and faces exploding/vanishing gradient problem with the increasing sequence length.

To correct these RNN issues, Long-Short Term Memory (LSTM) network with explicit long-term memory and easy gradient flow mechanism is proposed by Hochreiter and Schmidhuber [1997]. LSTM has a self-loop mechanism for long-term memory that decides what to “remember” and what to “forget” using its multiple gates (forget gate f_t, input gate i_t, output gate o_t) and a cell state c_t. The self-loop mechanism also allows gradients to flow freely between hidden states in long input sequences. LSTM computes the current hidden state h_t by element-wise multiplying the current cell state c_t with the output gate o_t, which are formulated as follows:

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$

where h_{t-1} is the previous hidden state, c_{t-1} is the previous cell state, σ_g is a sigmoid activation function, σ_c and σ_h both denote tanh activation functions, and $\circ$ represents the element-wise multiplication. W_f, W_i, W_o, W_c and b_f, b_i, b_o, b_c are weight matrices and bias terms that are learned during the training process. LeCun *et al.* [2015] shows that LSTM networks have proven to be more effective than conventional RNNs for modeling sequential data, especially when they have several layers.

One disadvantage of LSTM networks is that they only look at preceding text for contextual understanding. Since we can look up ahead of the current sensor event in our offline ADL recognition scenario, it makes sense to have LSTM be able look at the input text in bidirectional manner where it process the data in both forward and backward directions using two separate hidden layers as first proposed for RNN by Schuster and Paliwal [1997]. Such LSTM model is referred to as bidirectional LSTM and it computes separate forward and backward hidden sequences from the input sequence. If there are multiple layers, then bidirectional LSTM will have multiple layers for each direction, with the higher layers taking the lower layer's outputs as their inputs. For our proposed method, we use bidirectional LSTM with 3 layers.

At the end of the bidirectional LSTM networks, we concatenate the outputs from the two directions and pass it through a softmax layer to produce activity label probabilities:

$$y = softmax(Wz + b)$$

where z is the concatenated output from the bidirectional LSTM and W and b are the $softmax$ layer parameters.

3.4. Experiments

For our experiments, we use smart home datasets collected by the CASAS (Center for Advanced Studies in Adaptive Systems) at Washington State University in Cook *et al.* [2012]. The dataset consists of event logs containing a date, time, sensor identifier and the value sent by the sensor.

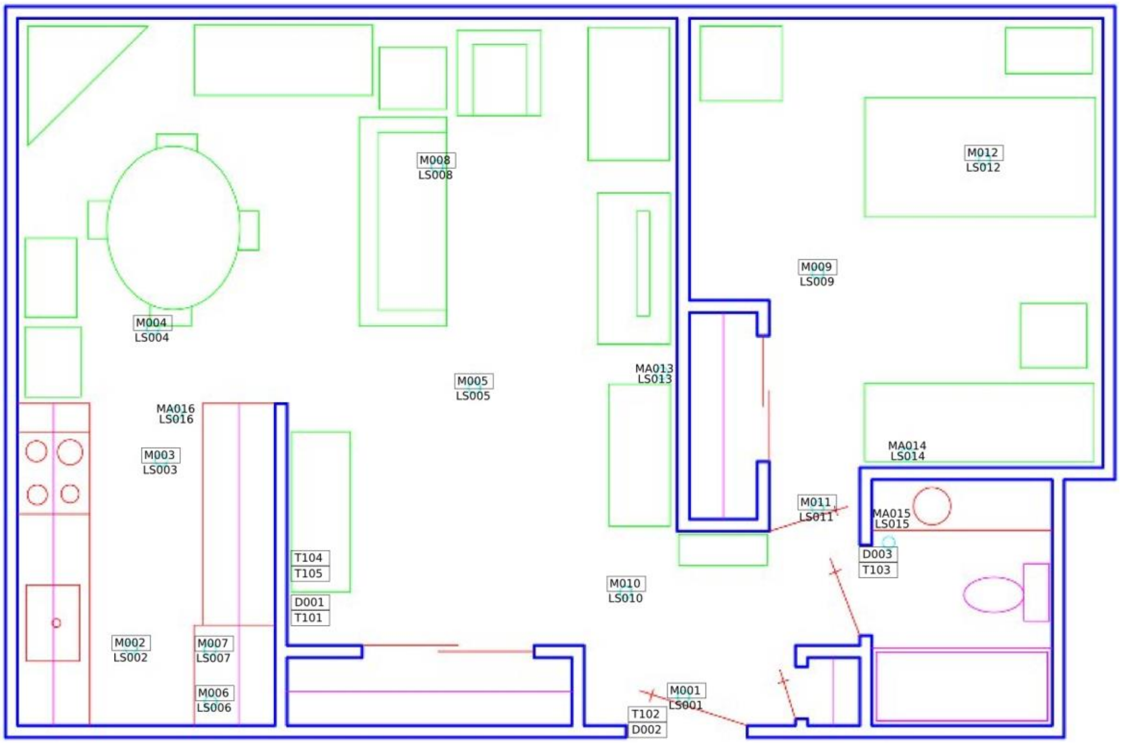


Figure 3.3. hh101 house sensor locations

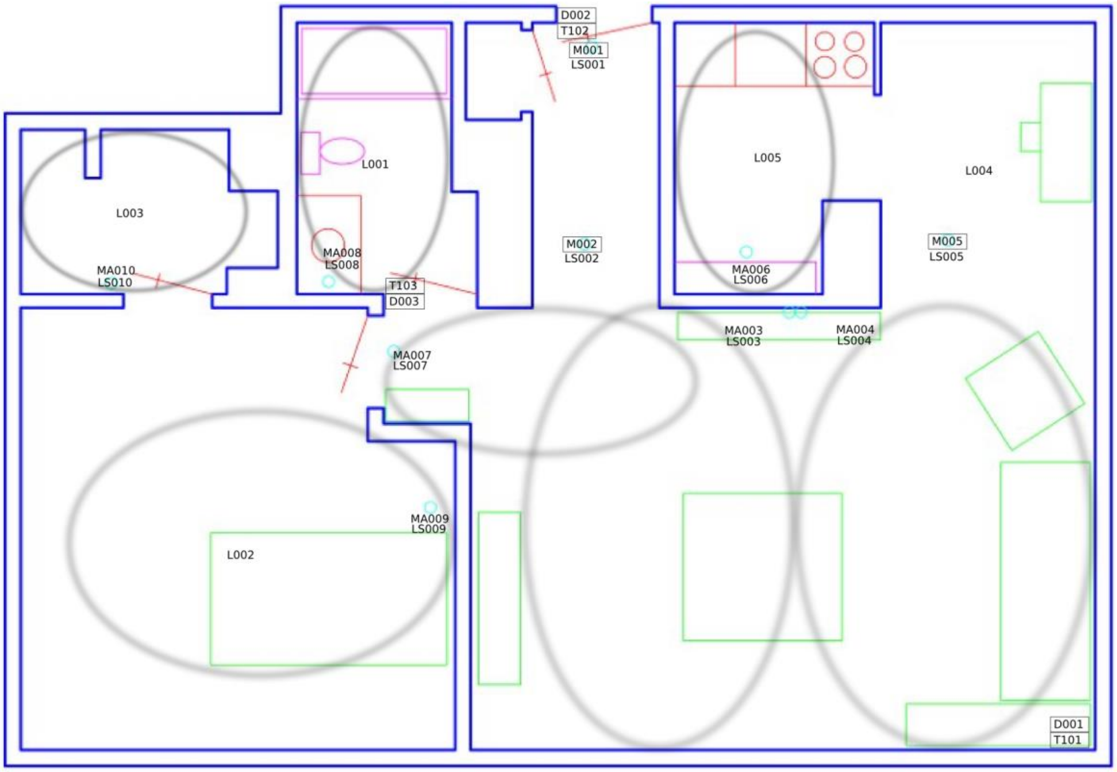


Figure 3.4. hh103 house sensor locations

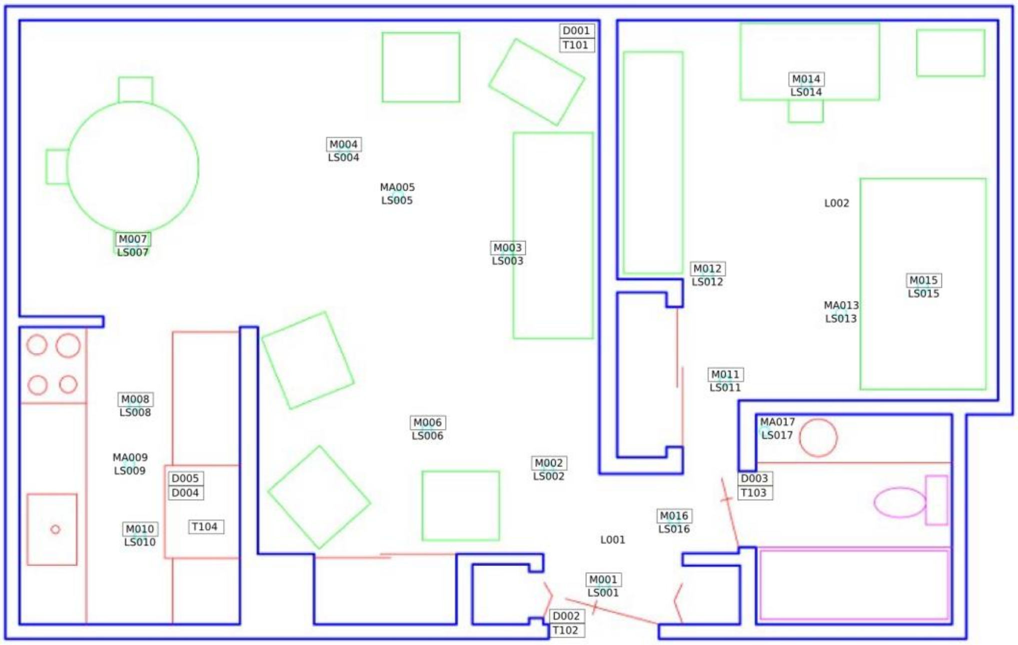


Figure 3.5. hh109 house sensor locations

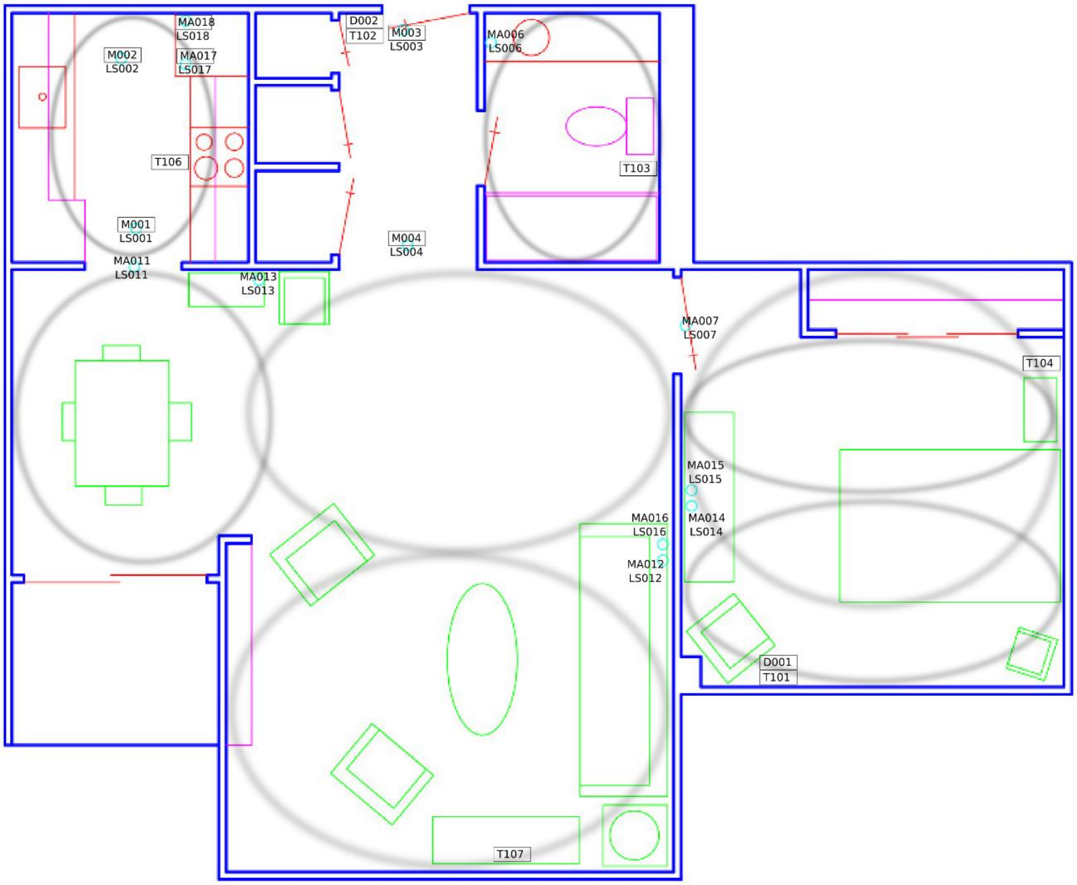


Figure 3.6. hh123 house sensor locations

3.4.1. Dataset

Amongst the smart homes, we have selected the apartments that have the most similar architectural outlines, with each having one bedroom, one living room, a kitchen area adjacent to the living room, a bathroom and front door area. Each apartment has embedded wireless motion, light, and door sensors and occupied by a single older adult resident who performs routine daily activities. The passive infrared motion sensors are mounted on the walls and ceilings and they send ‘ON’ message to the event logger when a motion is detected and ‘OFF’ message when the movement stops, as described in Aminikhanghahi and Cook [2019]. The motion sensor sends ‘OFF’ message if no movement is detected for 1.25 seconds. If the continuous movement is detected, then motion sensor will not send ‘OFF’ message until the movement stops, provided the ‘ON’ message was already sent. The door sensors send ‘OPEN’ or ‘CLOSE’ message when its magnetic switch is triggered. Besides the external doors, the door sensors are also sometimes mounted on either a room door or on a cabinet that holds medicines. There are two types of light sensors. First is the binary light sensor that is activated when a light is turned on/off. Second is the light sensor that sends detected light measurement value.

The smart home datasets are partially labeled and the activity labels are tagged by multiple human annotators who are given the house floor plan, the positions of the sensors, a resident-completed form describing the typical routine of the occupant's daily activities, in addition to the time-series sensor event logs. The inter-annotator agreement is set to 80%. For our experimentation, we have selected following four smart home datasets: hh101[5], hh103[6], hh109[7], and hh123[8]. The reasons for selecting these datasets are as follows:

a) All are single bedroom apartments with similar floorplans, occupied by a single elderly resident.
b) Have the greatest number of common activity classes (28).
c) All of the data are collected from embedded ambient sensors.

Apartment floorplans and the sensor locations are illustrated in Figure 3.3-Figure 3.6.

[5] http://ailab.wsu.edu/casas/hh/hh101/profile/page-1.html
[6] http://ailab.wsu.edu/casas/hh/hh103/profile/page-1.html
[7] http://ailab.wsu.edu/casas/hh/hh109/profile/page-1.html
[8] http://ailab.wsu.edu/casas/hh/hh123/profile/page-1.html

Table 3.5. Activity class instance percentages for each dataset. 'Unknown' class excluded.

ACTIVITY CLASSES	**DATASET**			
	hh101	**hh103**	**hh109**	**hh123**
Bathe	8.01%	3.02%	2.30%	2.12%
Bed_Toilet_Transition	0.33%	3.20%	0.60%	1.09%
Cook	0.89%	0.36%	2.31%	0.56%
Cook_Breakfast	7.64%	10.39%	7.32%	6.46%
Cook_Dinner	2.58%	17.62%	15.73%	15.60%
Cook_Lunch	1.43%	10.03%	10.11%	2.04%
Dress	4.26%	2.36%	6.57%	6.94%
Eat	0.19%	0.01%	0.33%	0.05%
Eat_Breakfast	1.61%	1.30%	1.01%	2.63%
Eat_Dinner	0.52%	1.83%	2.71%	1.52%
Eat_Lunch	0.36%	1.11%	1.95%	0.28%
Enter_Home	1.43%	1.31%	1.17%	1.60%
Evening_Meds	1.18%	0.56%	0.62%	1.02%
Leave_Home	1.65%	1.25%	1.28%	1.54%
Morning_Meds	1.44%	0.83%	0.98%	0.77%
Personal_Hygiene	6.15%	10.02%	5.25%	4.14%
Phone	0.43%	0.51%	0.18%	0.95%
Read	2.14%	0.80%	0.77%	0.61%
Relax	3.72%	1.16%	1.61%	1.89%
Sleep	5.13%	9.26%	3.14%	7.93%
Sleep_Out_Of_Bed	8.04%	0.81%	3.39%	0.04%
Toilet	6.21%	9.88%	5.35%	3.95%
Wash_Breakfast_Dishes	1.96%	1.08%	2.83%	1.79%
Wash_Dinner_Dishes	1.15%	2.20%	6.58%	7.83%
Wash_Dishes	1.82%	0.11%	2.33%	8.89%
Wash_Lunch_Dishes	0.44%	2.70%	2.67%	0.82%
Watch_TV	29.21%	4.89%	1.82%	16.93%
Work_At_Table	0.08%	1.40%	0.09087	0.02%
	212356	**103357**	**279454**	**95242**
	TOTAL SAMPLES			

Table 3.6. Activity class instance percentages for each dataset. 'Unknown' class included.

ACTIVITY CLASSES	**DATASET**			
	hh101	**hh103**	**hh109**	**hh123**
Bathe	5.45%	1.91%	1.18%	1.38%
Bed_Toilet_Transition	0.23%	2.02%	0.31%	0.71%
Cook	0.60%	0.23%	1.19%	0.36%
Cook_Breakfast	5.20%	6.57%	3.76%	4.21%
Cook_Dinner	1.75%	11.14%	8.08%	10.16%
Cook_Lunch	0.98%	6.34%	5.19%	1.33%
Dress	2.90%	1.49%	3.37%	4.52%
Eat	0.13%	0.01%	0.17%	0.03%
Eat_Breakfast	1.09%	0.82%	0.52%	1.71%
Eat_Dinner	0.35%	1.16%	1.39%	0.99%
Eat_Lunch	0.25%	0.70%	1.00%	0.18%
Enter_Home	0.97%	0.83%	0.60%	1.04%
Evening_Meds	0.80%	0.35%	0.32%	0.66%
Leave_Home	1.12%	0.79%	0.66%	1.00%
Morning_Meds	0.98%	0.52%	0.51%	0.50%
Personal_Hygiene	4.19%	6.34%	2.70%	2.69%
Phone	0.29%	0.32%	0.09%	0.62%
Read	1.46%	0.51%	0.40%	0.40%
Relax	2.53%	0.73%	0.83%	1.23%
Sleep	3.49%	5.85%	1.61%	5.16%
Sleep_Out_Of_Bed	5.48%	0.51%	1.74%	0.03%
Toilet	4.23%	6.25%	2.75%	2.57%
Unknown	31.90%	36.77%	48.65%	34.89%
Wash_Breakfast_Dishes	1.34%	0.68%	1.45%	1.17%
Wash_Dinner_Dishes	0.79%	1.39%	3.38%	5.10%
Wash_Dishes	1.24%	0.07%	1.20%	5.79%
Wash_Lunch_Dishes	0.30%	1.71%	1.37%	0.54%
Watch_TV	19.89%	3.09%	0.94%	11.02%
Work_At_Table	0.05%	0.88%	4.67%	0.01%
	311810	**163469**	**544196**	**146288**
	TOTAL SAMPLES			

3.4.2. Implementation details

The hidden dimension of the 3-layer bidirectional LSTM is set to 256. After concatenating the outputs from the LSTM, we apply dropout with probability of 0.1. For optimization, we use AdamW adaptive optimization algorithm proposed by Kingma and Ba [2014] with initial learning rate of 1e-3. Training batch size is set to 1000 sliding windows. We implement our proposed method using PyTorch library. We report the best results from the first 10 epochs.

3.4.3. Baselines

Feuz and Cook [2014] propose three heterogeneous transfer learning methods of Feature-Space Remapping (FSR), Genetic Algorithm for Feature-Space Remapping (GAFSR), and Greedy Search for Feature-Space Remapping (GrFSR) that transforms target domain data into source domain feature space. All of the three proposed approaches use some labelled target data to infer relations to the source domain. They experiment on 18 smart home datasets [Cook et al. 2012] to recognize 37 activity classes *('Bathe', 'Bed Toilet Transition', 'Cook Breakfast', 'Cook Dinner', 'Cook Lunch', 'Cook', 'Dress', 'Eat Breakfast', 'Eat Dinner', 'Eat Lunch', 'Eat', 'Entertain Guests', 'Evening Meds', 'Exercise', 'Groom', 'Housekeeping', 'Leave Home', 'Morning Meds', 'Other Activity', 'Personal Hygiene', 'Phone', 'Read', 'Relax', 'Sleep Out of Bed', 'Sleep', 'Take Medicine', 'Toilet', 'Wash Breakfast Dishes', 'Wash Dinner Dishes', 'Wash Dishes', 'Wash Lunch Dishes', 'Watch TV', 'Work at Desk', 'Work at Table', 'Work on Computer', 'Work', 'Enter Home'*). For multi-source transfer learning, they propose Ensemble Learning via Feature Space Remapping (ELFSR) that trains Naïve Bayes classifier for each source dataset and test on the target domain based on two ensemble voting schemes. We take two results from their work:

1) **IFSR-Maj**: best performance of ELFSR in a majority voting ensemble,
2) **IFSR-Sum**: best performance of ELFSR in a summed probability ensemble.

Wemlinger and Holder [2018] propose Semantic Cross-Environment Activity Recognition (SCEAR) system that projects raw sensor activities from different smart environments into common semantic feature space between different domains and transfers data-driven models across environments. SCEAR is composed of an ontology component and a set of sensor reasoners. Only feature present in both source and target environments are used to train the model to recognize 28 activity classes (*'Bathing', 'Bed_To_Toilet', 'Breakfast', 'Changing_Clothes', 'Cooking', 'Dinner', 'Eating', 'Enter_Home', 'Exercise', 'Having_Guest', 'Housekeeping',* *‘Leave_Home’, ‘Lunch’, ‘Napping’, ‘Out_of_Home’, ‘Personal_Hygiene’, ‘Read’, ‘Relax’, ‘Sleeping’, ‘Study’, ‘Take_medicine’, ‘Talking_On_The_Phone’, ‘Toileting’, ‘Wake’, ‘Wandering’, ‘Wash_Dishes’, ‘Watch_TV’, ‘Work’*). We take two results from their work:

1) **SCEAR-hh123**: best single source SCEAR performance on hh123 as target dataset,
2) **SCEAR-Best**: best SCEAR performance on any dataset.

3.4.4. Experimentation scenarios

The common ADL class labels in the selected datasets are: {*'Bathe', 'Bed_Toilet_Transition', 'Cook', 'Cook_Breakfast', 'Cook_Dinner', 'Cook_Lunch', 'Dress', 'Eat', 'Eat_Breakfast', 'Eat_Dinner', 'Eat_Lunch', 'Enter_Home', 'Evening_Meds', 'Leave_Home', 'Morning_Meds', 'Personal_Hygiene', 'Phone', 'Read', 'Relax', 'Sleep', 'Sleep_Out_Of_Bed', 'Toilet', 'Wash_Breakfast_Dishes', 'Wash_Dinner_Dishes', 'Wash_Dishes', 'Wash_Lunch_Dishes', 'Watch_TV', 'Work_At_Table'*}. The datasets are only partially labelled, with the rest of the unannotated data labeled as ‘Unknown’.

In Table 3.5 and Table 3.6 show the percentage breakdown of activity class samples for each dataset with the ‘Unknown’ activities both included and excluded.

In the literature, some approaches use the unannotated data as ‘Other’ or ‘Unknown’ activity type and evaluate their performances including it, and others do not include ‘Unknown’ when evaluating their performances, as can be seen in our baselines. In order to fairly compare results with the baselines, we evaluate our proposed method with the following experimentation scenarios:

1) Select one dataset as a target domain and train the model on the rest of the datasets (multi-source). Multiple source domains’ data are combined without any weighting scheme. ‘Unknown’ activity samples are included when evaluation performance.
2) Same as the above with single target domain and multiple source domains, but the ‘Unknown’ activity samples are excluded from evaluation.
3) Single source domain and a single target domain with the ‘Unknown’ activity samples included during evaluation.
4) Single source domain and a single target domain, but without the ‘Unknown’ activity samples during evaluation.

Most of the ADL recognition literature report only the accuracy rates. For completeness and fair assessment, we report both the accuracy and the F1 score for each experiment scenarios. The accuracy is calculated as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

The F1 score is calculated as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

In our multi-label classification case, we report the macro F1 score, which is unweighted average of the F1 score for each class. F1 score emphasizes the misclassification of labels with few samples, as opposed to accuracy that highlights the classification performance on classes with larger number of samples.

3.4.5. Results

Figure 3.7 shows the accuracy rates of our proposed method in multi-source and single-target scenario with the ‘Unknown’ class included, for target domains of hh101, hh103, hh109, and hh109. Our approach outperforms the methods of IFSR-Maj and IFSR-Sum, even though these baselines use labeled target data. Figure 3.8 displays the F1 scores for this experiment scenario. We can already observe discrepancy between accuracy and F1 score due to the imbalanced activity classes. Unfortunately, IFSR-Maj and IFRS-Sum do not report the F1 scores.

Figure 3.9 shows the performances of our multi-source and single-target scenario without the ‘Unknown’ class. Our worst performing model with target domain hh123 is competitive with the best result from the SCEAR method. We can also directly compare our method’s performance on target domain hh123 with the SCEAR’s performance on the same target domain. Our approach significantly outperforms the SCEAR method. Also in Figure 3.10, the F1 score results show the effectiveness of method where our worst performing model has the same score as the best SCEAR model.

In general, the accuracy drops compared to the results that included ‘Unknown’ because of the large number of samples that are labeled as ‘Unknown’ creates a strong bias towards the class and causes the classifier to predict large mass of sensor events as ‘Unknown’. We can see in Figure 3.10 that once the ‘Unknown’ is excluded, there are slight increases in F1 scores.

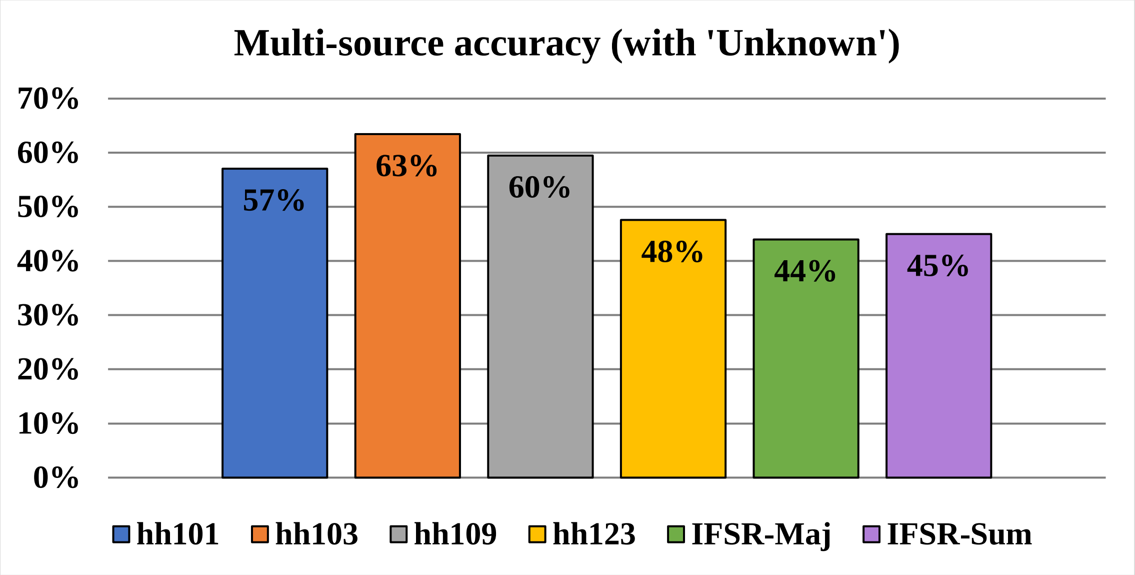


Figure 3.7. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.

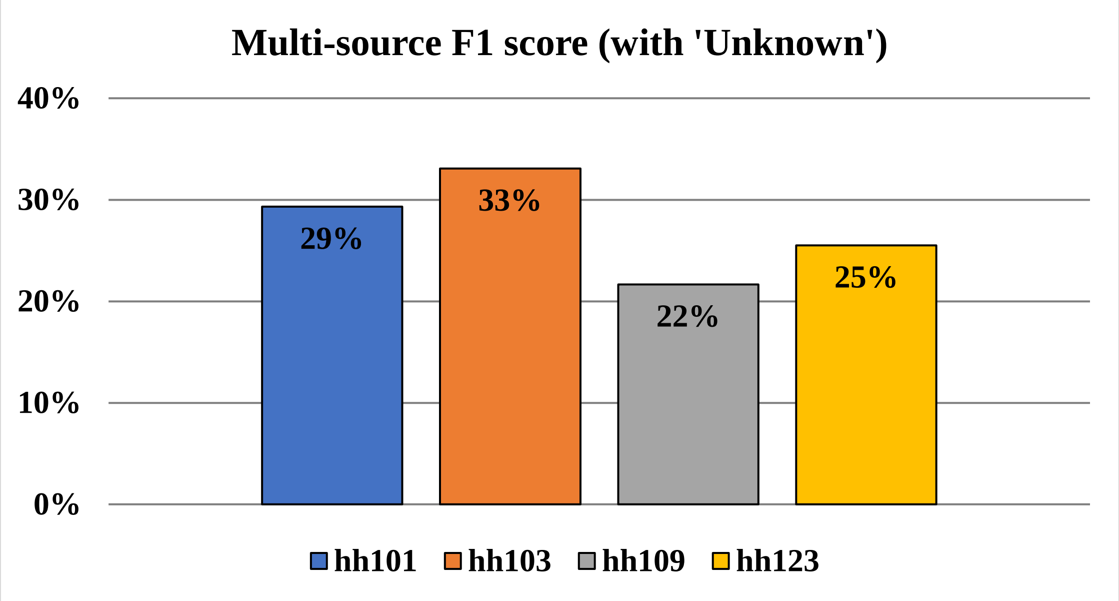


Figure 3.8. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.

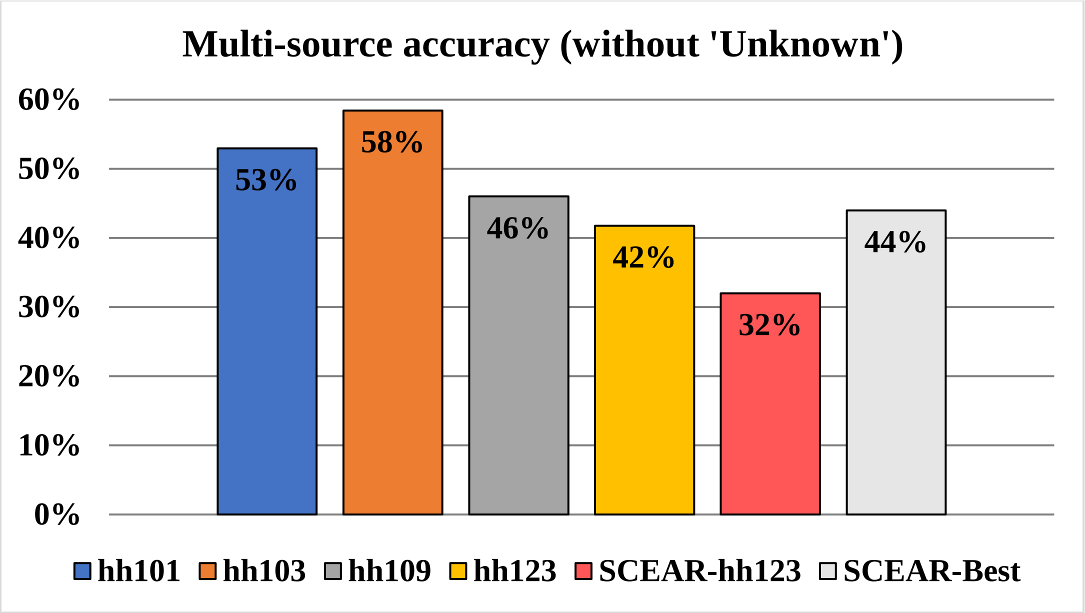


Figure 3.9. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.

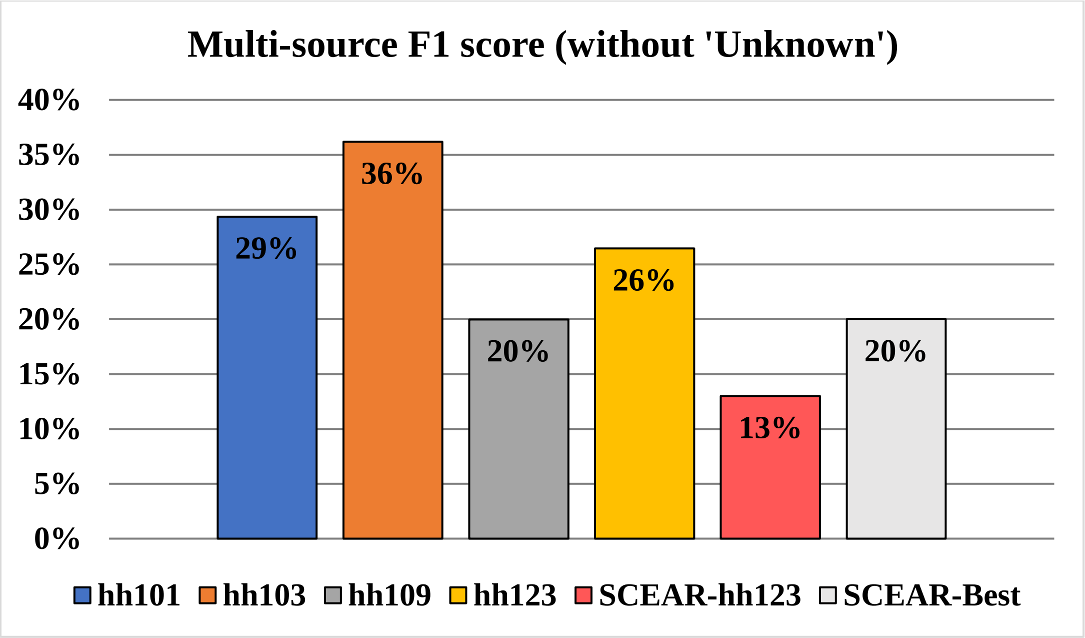


Figure 3.10. Performance of multiple source models on target domains hh101, hh103, hh109, and hh123.

Figure 3.11 - Figure 3.14 show the accuracy rates and F1 scores for single-source and single-target experiments. It becomes apparent that including the ‘Unknown’ class for evaluating an ADL recognition models gives false impression of high accuracy rate, where in fact observing the F1 score gives more accurate depiction of models’ performance.

The scale of confusion ‘Unknown’ class introduces to the classifier can be seen from the confusion matrix in Figure 3.15. The main reason for such high concentration of confusion is that the ‘Unknown’ class itself do not have any discernable patterns or structure, and it is just label assigned events that did not belong to known activity classes.

In Figure 3.16, the classifier confusion is more spread out and most of the confusion comes from activities performed in the same location areas.

3.5. Conclusion

In this chapter we presented a novel heuristic approach for multiple low-level feature mappings between heterogeneous sensor data when there is a lack of available recognized embedding methods to process the low-level features, as there are in computer vision and NLP tasks. The experiment results show that the proposed feature mappings method shows promising results when combined with bidirectional LSTM for unsupervised cross-domain ADL recognition. With exception of the sensor location extraction, our heuristic mapping method is performed automatically without any manual mapping between sensors of different datasets. Since there are very few works done in unsupervised transfer learning of smart-home ADL recognition, our work advances the field with respect to providing updated baseline of low-level feature extraction for such transfer learning problem.

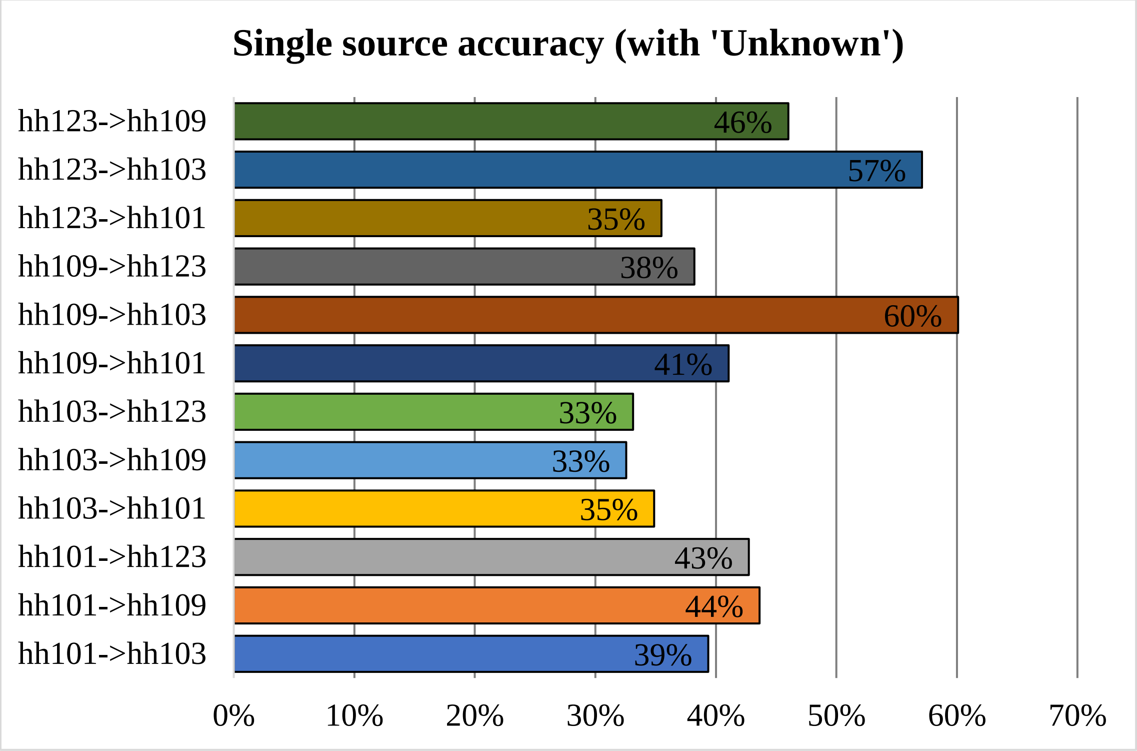


Figure 3.11. Single-source, single-target classification accuracy.

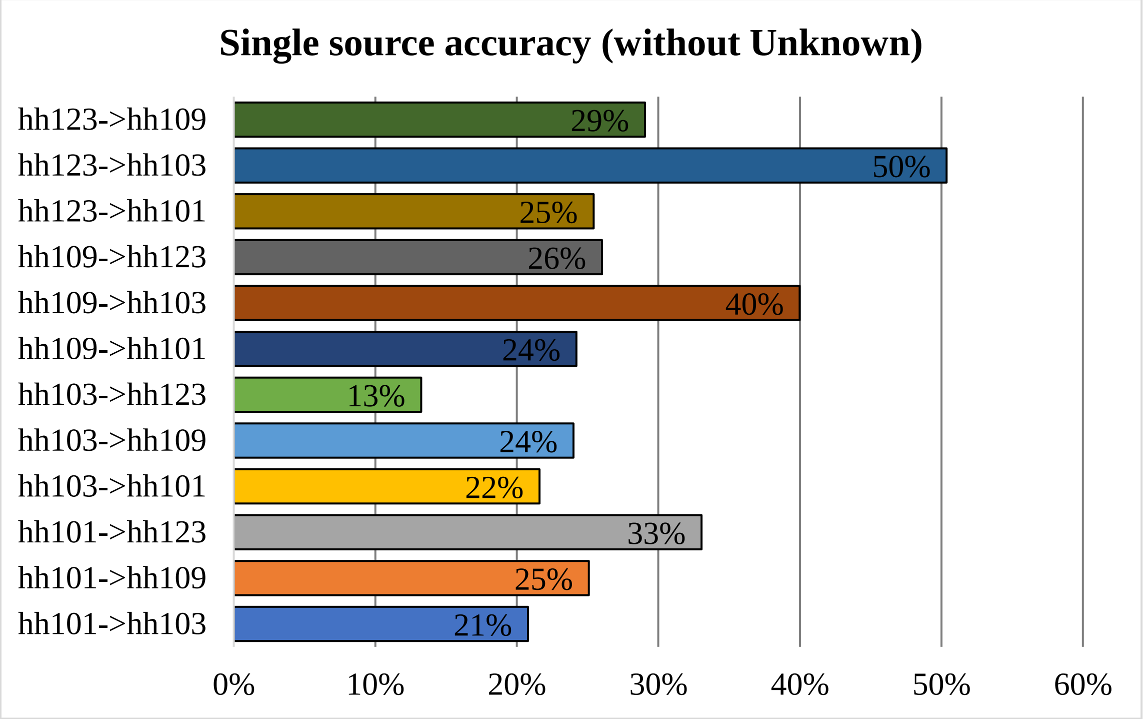


Figure 3.12. Single-source, single-target classification accuracy.

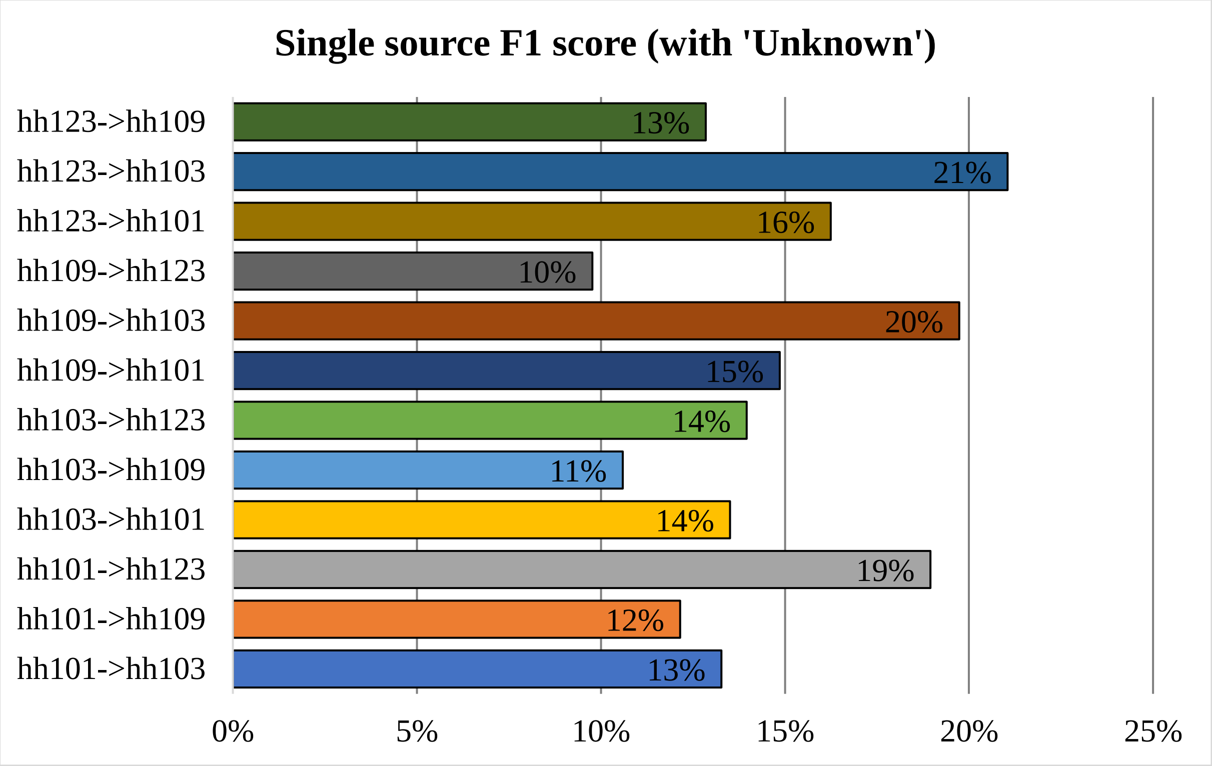


Figure 3.13. Single-source, single-target F1 score.

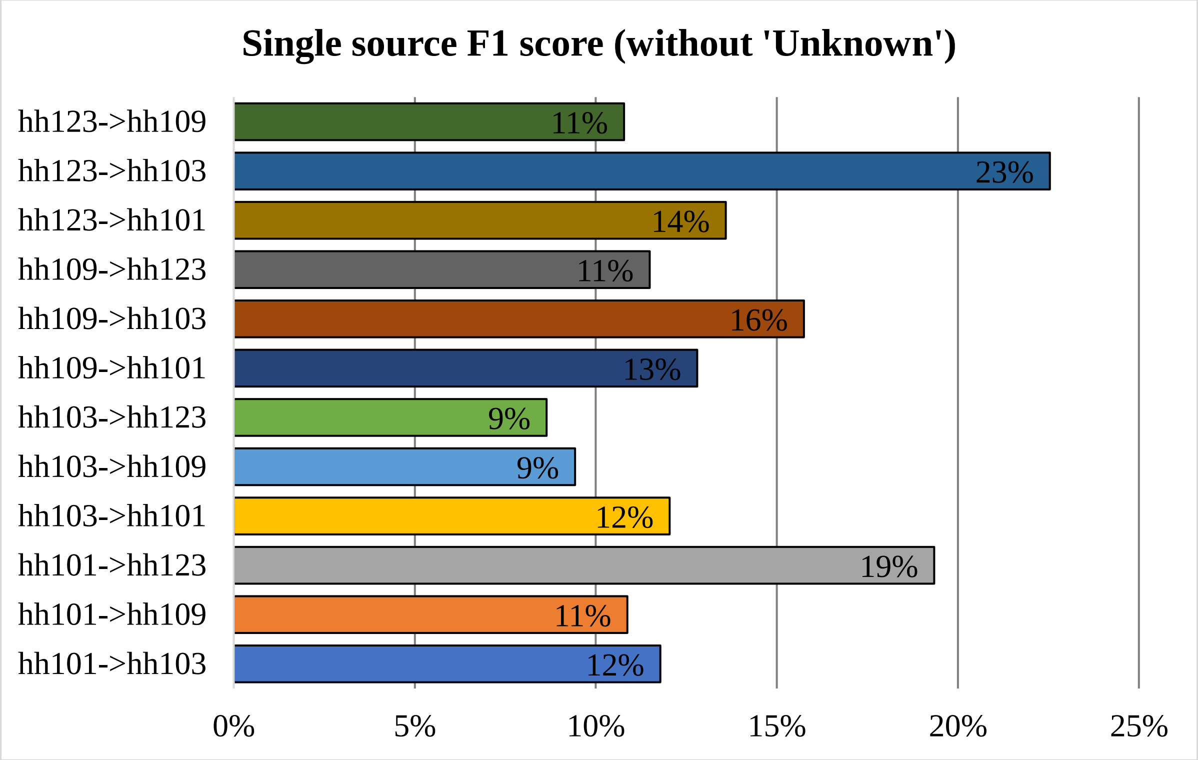
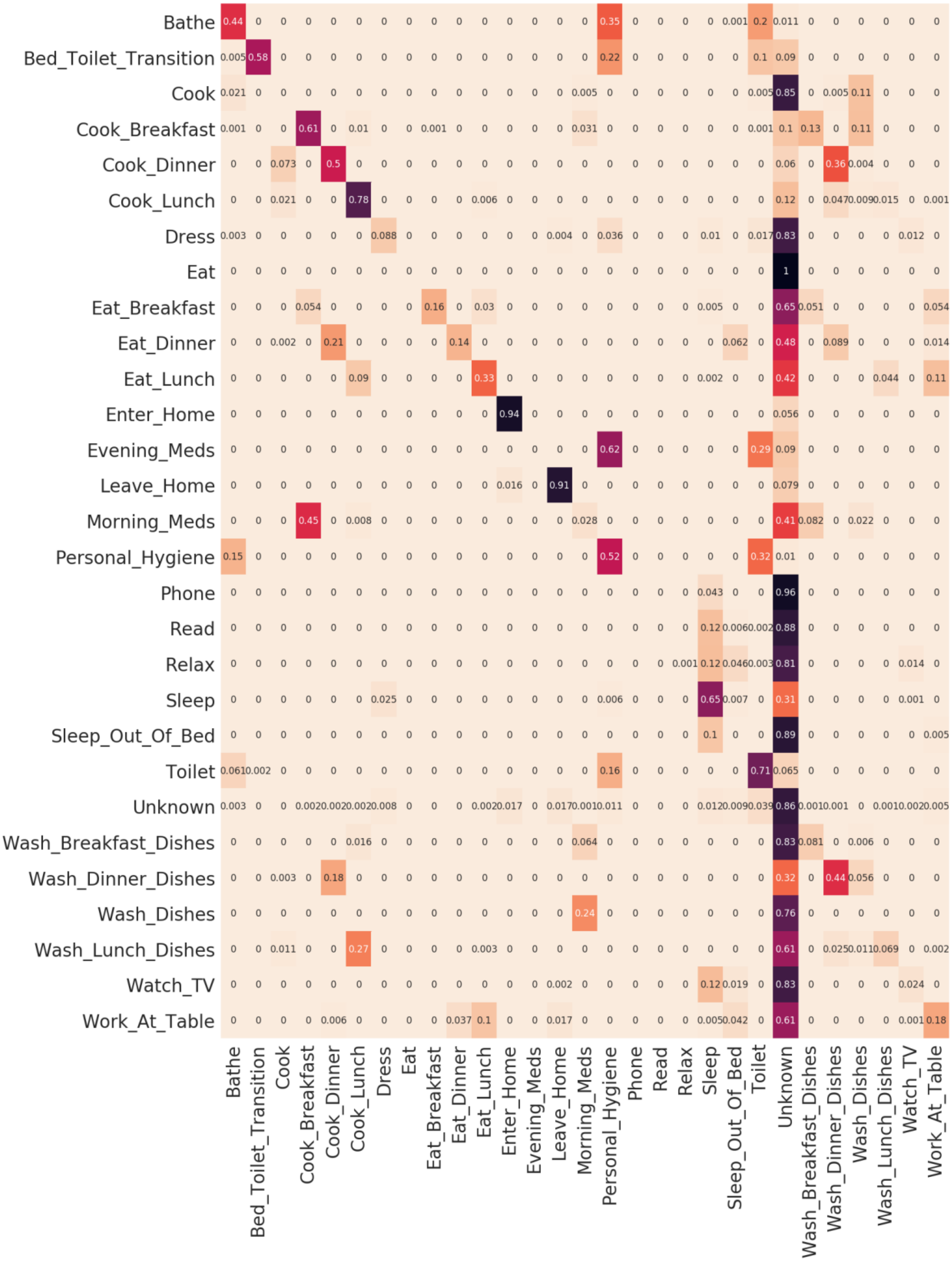
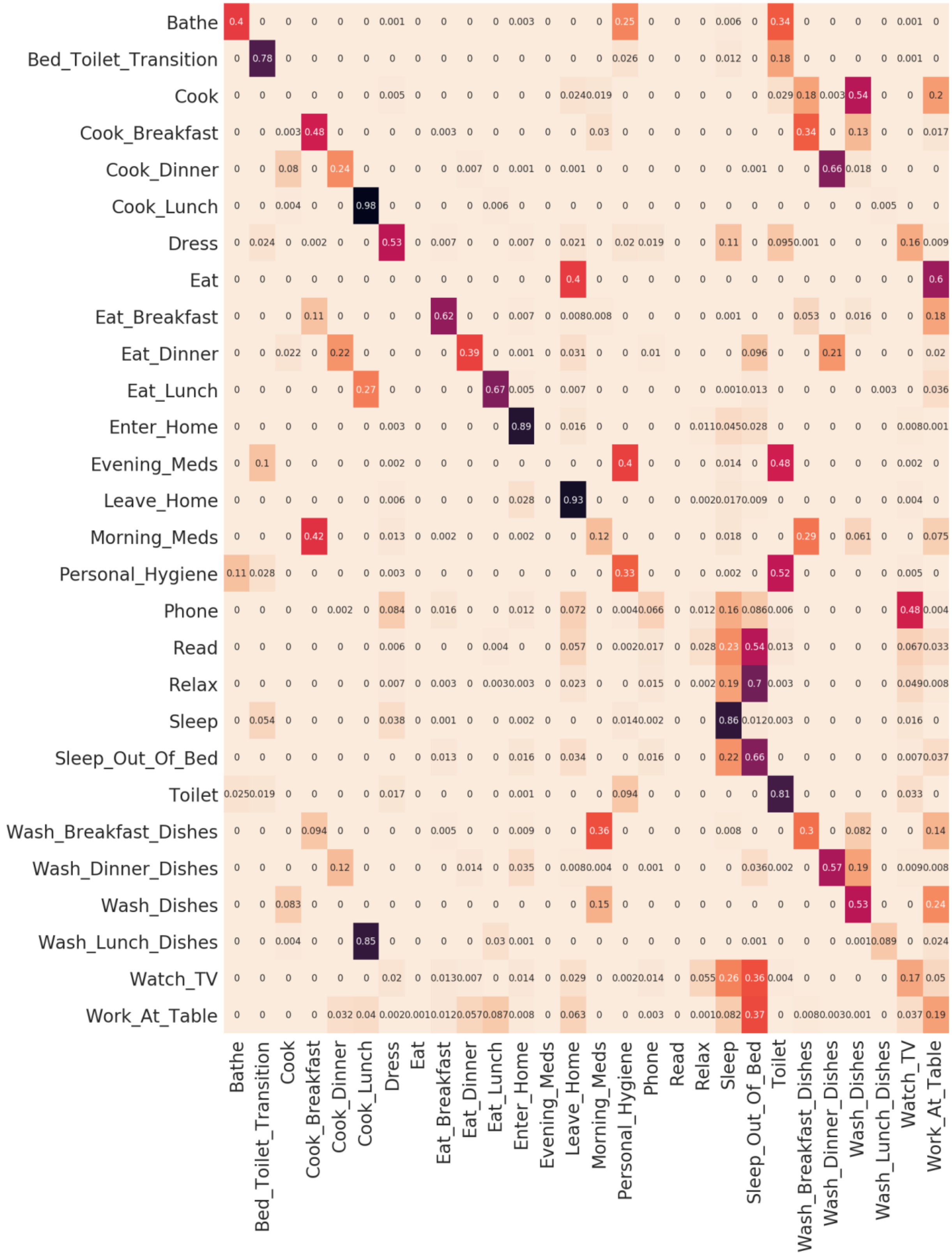


Figure 3.14. Single-source, single-target F1 score.



	Bathe	Bed_Toilet_Transition	Cook	Cook_Breakfast	Cook_Dinner	Cook_Lunch	Dress	Eat	Eat_Breakfast	Eat_Dinner	Eat_Lunch	Enter_Home	Evening_Meds	Leave_Home	Morning_Meds	Personal_Hygiene	Phone	Read	Relax	Sleep	Sleep_Out_Of_Bed	Toilet	Unknown	Wash_Breakfast_Dishes	Wash_Dinner_Dishes	Wash_Dishes	Wash_Lunch_Dishes	Watch_TV	Work_At_Table
Bathe	0.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.35	0	0	0	0	0.001	0.2	0.011	0	0	0	0	0	0
Bed_Toilet_Transition	0.005	0.58	0	0	0	0	0	0	0	0	0	0	0	0	0	0.22	0	0	0	0	0	0.1	0.09	0	0	0	0	0	0
Cook	0.021	0	0	0	0	0	0	0	0	0	0	0	0	0	0.005	0	0	0	0	0	0	0.005	0.85	0	0.005	0.11	0	0	0
Cook_Breakfast	0.001	0	0	0.61	0	0.01	0	0	0.001	0	0	0	0	0	0.031	0	0	0	0	0	0	0.001	0.1	0.13	0	0.11	0	0	0
Cook_Dinner	0	0	0.073	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0	0.36	0.004	0	0	0
Cook_Lunch	0	0	0.021	0	0	0.78	0	0	0	0	0.006	0	0	0	0	0	0	0	0	0	0	0	0.12	0	0.047	0.009	0.015	0	0.001
Dress	0.003	0	0	0	0	0	0.088	0	0	0	0	0	0	0.004	0	0.036	0	0	0	0.01	0	0.017	0.83	0	0	0	0	0.012	0
Eat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Eat_Breakfast	0	0	0	0.054	0	0	0	0	0.16	0	0.03	0	0	0	0	0	0	0	0	0.005	0	0	0.65	0.051	0	0	0	0	0.054
Eat_Dinner	0	0	0.002	0	0.21	0	0	0	0	0.14	0	0	0	0	0	0	0	0	0	0	0.062	0	0.48	0	0.089	0	0	0	0.014
Eat_Lunch	0	0	0	0	0	0.09	0	0	0	0	0.33	0	0	0	0	0	0	0	0	0.002	0	0	0.42	0	0	0	0.044	0	0.11
Enter_Home	0	0	0	0	0	0	0	0	0	0	0	0.94	0	0	0	0	0	0	0	0	0	0	0.056	0	0	0	0	0	0
Evening_Meds	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.62	0	0	0	0	0	0.29	0.09	0	0	0	0	0	0
Leave_Home	0	0	0	0	0	0	0	0	0	0	0	0.016	0	0.91	0	0	0	0	0	0	0	0	0.079	0	0	0	0	0	0
Morning_Meds	0	0	0	0.45	0	0.008	0	0	0	0	0	0	0	0	0.028	0	0	0	0	0	0	0	0.41	0.082	0	0.022	0	0	0
Personal_Hygiene	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.52	0	0	0	0	0	0.32	0.01	0	0	0	0	0	0
Phone	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.043	0	0	0.96	0	0	0	0	0	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.12	0.006	0.002	0.88	0	0	0	0	0	0
Relax	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.001	0.12	0.046	0.003	0.81	0	0	0	0	0.014	0
Sleep	0	0	0	0	0	0	0.025	0	0	0	0	0	0	0	0	0.006	0	0	0	0.65	0.007	0	0.31	0	0	0	0	0.001	0
Sleep_Out_Of_Bed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0.89	0	0	0	0	0	0.005
Toilet	0.061	0.002	0	0	0	0	0	0	0	0	0	0	0	0	0	0.16	0	0	0	0	0	0.71	0.065	0	0	0	0	0	0
Unknown	0.003	0	0	0.002	0.002	0.002	0.008	0	0	0	0.002	0.017	0	0.017	0.001	0.011	0	0	0	0.012	0.009	0.039	0.86	0.001	0.001	0	0.001	0.002	0.005
Wash_Breakfast_Dishes	0	0	0	0	0	0.016	0	0	0	0	0	0	0	0	0.064	0	0	0	0	0	0	0	0.83	0.081	0	0.006	0	0	0
Wash_Dinner_Dishes	0	0	0.003	0	0.18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.32	0	0.44	0.056	0	0	0
Wash_Dishes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.24	0	0	0	0	0	0	0	0.76	0	0	0	0	0	0
Wash_Lunch_Dishes	0	0	0.011	0	0	0.27	0	0	0	0	0.003	0	0	0	0	0	0	0	0	0	0	0	0.61	0	0.025	0.011	0.069	0	0.002
Watch_TV	0	0	0	0	0	0	0	0	0	0	0	0	0	0.002	0	0	0	0	0	0.12	0.019	0	0.83	0	0	0	0	0.024	0
Work_At_Table	0	0	0	0	0.006	0	0	0	0	0.037	0.1	0	0	0.017	0	0	0	0	0	0.005	0.042	0	0.61	0	0	0	0	0.001	0.18

Figure 3.15. Confusion matrix when evaluated with 'Unknown' activity included.



	Bathe	Bed_Toilet_Transition	Cook	Cook_Breakfast	Cook_Dinner	Cook_Lunch	Dress	Eat	Eat_Breakfast	Eat_Dinner	Eat_Lunch	Enter_Home	Evening_Meds	Leave_Home	Morning_Meds	Personal_Hygiene	Phone	Read	Relax	Sleep	Sleep_Out_Of_Bed	Toilet	Wash_Breakfast_Dishes	Wash_Dinner_Dishes	Wash_Dishes	Wash_Lunch_Dishes	Watch_TV	Work_At_Table
Bathe	0.4	0	0	0	0	0	0.001	0	0	0	0	0.003	0	0	0	0.25	0	0	0	0.006	0	0.34	0	0	0	0	0.001	0
Bed_Toilet_Transition	0	0.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0.026	0	0	0	0.012	0	0.18	0	0	0	0	0.001	0
Cook	0	0	0	0	0	0	0.005	0	0	0	0	0	0	0.024	0.019	0	0	0	0	0	0	0.029	0.18	0.003	0.54	0	0	0.2
Cook_Breakfast	0	0	0.003	0.48	0	0	0	0	0.003	0	0	0	0	0	0.03	0	0	0	0	0	0	0	0.34	0	0.13	0	0	0.017
Cook_Dinner	0	0	0.08	0	0.24	0	0	0	0	0.007	0	0.001	0	0.001	0	0	0	0	0	0	0.001	0	0	0.66	0.018	0	0	0
Cook_Lunch	0	0	0.004	0	0	0.98	0	0	0	0	0.006	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.005	0	0
Dress	0	0.024	0	0.002	0	0	0.53	0	0.007	0	0	0.007	0	0.021	0	0.02	0.019	0	0	0.11	0	0.095	0.001	0	0	0	0.16	0.009
Eat	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6
Eat_Breakfast	0	0	0	0.11	0	0	0	0	0.62	0	0	0.007	0	0.008	0.008	0	0	0	0	0.001	0	0	0.053	0	0.016	0	0	0.18
Eat_Dinner	0	0	0.022	0	0.22	0	0	0	0	0.39	0	0.001	0	0.031	0	0	0.01	0	0	0	0.096	0	0	0.21	0	0	0	0.02
Eat_Lunch	0	0	0	0	0	0.27	0	0	0	0	0.67	0.005	0	0.007	0	0	0	0	0	0.001	0.013	0	0	0	0	0.003	0	0.036
Enter_Home	0	0	0	0	0	0	0.003	0	0	0	0	0.89	0	0.016	0	0	0	0	0.011	0.045	0.028	0	0	0	0	0	0.008	0.001
Evening_Meds	0	0.1	0	0	0	0	0.002	0	0	0	0	0	0	0	0	0.4	0	0	0	0.014	0	0.48	0	0	0	0	0.002	0
Leave_Home	0	0	0	0	0	0	0.006	0	0	0	0	0.028	0	0.93	0	0	0	0	0.002	0.017	0.009	0	0	0	0	0	0.004	0
Morning_Meds	0	0	0	0.42	0	0	0.013	0	0.002	0	0	0.002	0	0	0.12	0	0	0	0	0.018	0	0	0.29	0	0.061	0	0	0.075
Personal_Hygiene	0.11	0.028	0	0	0	0	0.003	0	0	0	0	0	0	0	0	0.33	0	0	0	0.002	0	0.52	0	0	0	0	0.005	0
Phone	0	0	0	0	0.002	0	0.084	0	0.016	0	0	0.012	0	0.072	0	0.004	0.066	0	0.012	0.16	0.086	0.006	0	0	0	0	0.48	0.004
Read	0	0	0	0	0	0	0.006	0	0	0	0.004	0	0	0.057	0	0.002	0.017	0	0.028	0.23	0.54	0.013	0	0	0	0	0.067	0.033
Relax	0	0	0	0	0	0	0.007	0	0.003	0	0.003	0.003	0	0.023	0	0	0.015	0	0.002	0.19	0.7	0.003	0	0	0	0	0.049	0.008
Sleep	0	0.054	0	0	0	0	0.038	0	0.001	0	0	0.002	0	0	0	0.014	0.002	0	0	0.86	0.012	0.003	0	0	0	0	0.016	0
Sleep_Out_Of_Bed	0	0	0	0	0	0	0	0	0.013	0	0	0.016	0	0.034	0	0	0.016	0	0	0.22	0.66	0	0	0	0	0	0.007	0.037
Toilet	0.025	0.019	0	0	0	0	0.017	0	0	0	0	0.001	0	0	0	0.094	0	0	0	0	0	0.81	0	0	0	0	0.033	0
Wash_Breakfast_Dishes	0	0	0	0.094	0	0	0	0	0.005	0	0	0.009	0	0	0.36	0	0	0	0	0.008	0	0	0.3	0	0.082	0	0	0.14
Wash_Dinner_Dishes	0	0	0	0	0.12	0	0	0	0	0.014	0	0.035	0	0.008	0.004	0	0.001	0	0	0	0.036	0.002	0	0.57	0.19	0	0.009	0.008
Wash_Dishes	0	0	0.083	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0.53	0	0	0.24
Wash_Lunch_Dishes	0	0	0.004	0	0	0.85	0	0	0	0	0.03	0.001	0	0	0	0	0	0	0	0	0.001	0	0	0	0.001	0.089	0	0.024
Watch_TV	0	0	0	0	0	0	0.02	0	0.013	0.007	0	0.014	0	0.029	0	0.002	0.014	0	0.055	0.26	0.36	0.004	0	0	0	0	0.17	0.05
Work_At_Table	0	0	0	0	0.032	0.04	0.002	0.001	0.012	0.057	0.087	0.008	0	0.063	0	0	0.003	0	0.001	0.082	0.37	0	0.008	0.003	0.001	0	0.037	0.19

Figure 3.16. Confusion matrix when evaluated without 'Unknown' class samples.

Chapter 4. Mid-level common feature representations with pre-trained language models

4.1. Introduction

With the ever-increasing integration of social media and e-commerce into people's daily lives and with the greater availability of user opinion and sentiment data, the research in sentiment analysis have garnered great interest both in academy and the industry. Sentiment analysis or sentiment classification is a Natural Language Processing (NLP) task that deals with classifying the polarity of the input text document towards a particular target.

Deep neural networks have been successfully applied for diverse machine learning problems, including various NLP tasks, with greatly improved prediction performance metrics. The standard model training for a NLP task had focused on initializing the first layer of a neural network with pretrained word vectors such as word2vec by Mikolov *et al.* [2013] and GloVe in Pennington *et al*. [2014], and the rest of the network is trained on the task-specific data with convolutional and/or recurrent neural networks. Krizhevsky *et al.* [2012] shows CNN are able to learn the local response from the temporal or spatial data but lack the ability to learn sequential correlations. Recurrent Neural Networks (RNN) are used by Socher *et al.* [2013] because of their sequence modelling capabilities and dealing with short-term dependencies in a sequence of data but it is shown that RNNs have trouble when dealing with long-term dependencies. Long Short-Term Memory networks (LSTM), first proposed by Hochreiter and Schmidhuber [1997], which is a variation of RNN architecture, aims to solve the long-term dependency problem by introducing a memory into the network. RNN-based deep learning architectures has been the standard for various NLP tasks, including sentiment classification. However, these approaches still processed context in one direction only, i.e., create dependencies only on the left or right side of the current word. Therefore, they cannot capture contexts in both directions at the same time, i.e., consider words on both sides of the current word when capturing dependencies.

Most of these performance improvements in NLP with deep neural networks come only via supervised learning with massive amounts of labeled data. However, in real world applications, there are many scenarios where it is difficult to collect sufficient data for high-performing supervised learning model of a specific task due to factors of scarcity of readily available data or the high expense of data collection. In addition, statistical classifiers assume that both the training and test data come from a common underlying distribution, as in Li [2012], but due to the high variability and sparsity of natural language, oftentimes there is distribution differences in the real world data and the specialized training data, as described by Goldbert [2017].

Transfer learning allows us to deal with this scenario by borrowing information from a relevant source domain with abundant labeled data to help improve the prediction performance in the target domain [Wan, 2009]. *Cross-domain sentiment classification* (CDSC) aims at leveraging knowledge obtained from a source domain to train a high-performance learner for sentiment classification on a target domain, e.g., book product review, to help classification in the target domain, e.g., electronics product review, with few or no labeled data. In the literature, transfer learning techniques have been applied to CDSC. Traditional pivot-based CDSC schemes in Blitzer *et al.* [2007], Yu and Jiang [2016] attempt to infer the correlation between pivot words, i.e., the domain-shared sentiment words, and non-pivot words, i.e., the domain-specific sentiment words, by utilizing multiple pivot prediction tasks. However, these schemes share a major limitation that manual selection of pivots is required.

All of the above discussed schemes need to train a dedicated NLP model from scratch for every new task with its own specialized training data, which could take days and weeks to converge to a stable, high-performance model. Alternatively, substantial work has shown that unsupervised pre-trained language models on large text corpus are beneficial for text classification and other NLP tasks, which can avoid training a new model from scratch. Various approaches are proposed for training general purpose language representation models using an enormous amount of unannotated text, such as ELMo in Peters *et al.* [2018] and GPT in Radford *et al.* [2018]. Pre-trained models can be fine-tuned on NLP tasks without requiring huge amount of labeled data and have achieved significant improvement over training on task-specific annotated data. More recently, a pre-training technique, Bidirectional Encoder Representations from Transformers (BERT) is proposed in Devlin *et al.* [2019] and has created state-of-the-art models for a wide variety of NLP tasks, including question answering (SQuAD v1.1), natural language inference, text classification and others. The latest of such pre-trained language models is XLNet, introduced by Yang *et al.* [2019], a generalized autoregressive pretraining method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and overcomes the limitations of BERT thanks to its autoregressive formulation. Furthermore, XLNet integrates ideas from Transformer-XL model by Dai *et al.* [2019] into pretraining.

In this chapter, we fine-tune BERT and XLNet for CDSC and compare them with the current state-of-the-art methods. We also closely study their performances in comparison to each other with various experimental settings. Our main contributions are summarized as follows:

- This is the first work to explore the usage of Transformer-based bidirectional contextualized language models for CDSC.
- Compare and comprehensively analyze the performance of the two highest performing Transformer language models of XLNet and BERT in the context of CDSC.
- Achieves new state-of-the-arts results with significant improvements over the previous approaches.

4.2. Related works

Over the last decade, many methods have been proposed for cross-domain sentiment classification. Structural Correspondence Learning (SCL) method is proposed by Blitzer *et al.* [2007] to learn a joint low-dimensional feature representation for the source and target domains. Similarly, Pan *et al.* 2010 propose a Spectral Feature Alignment (SFA) method to align the pivots with the non-pivots to build a bridge between the source and target domains. However, these methods need to manually select the pivots based on criterions such as the frequency in both domains, the mutual information between features and labels on the source domain data, and the mutual information between features and domains. Domain-Adversarial training of Neural Networks (DANN) is proposed by Ganin *et al.* [2016] for domain adaptation using a gradient reversal layer to reverse the gradient direction in order to produce representations such that a domain classifier cannot predict the domain of the encoded representation, and at the same time, a sentiment classifier is built on the representation shared by domains to reduce the domain discrepancy and achieves better performance for cross-domain sentiment classification. Proposed approaches by Sun *et al.* [2016], and Zellinger *et al.* [2017] focus on learning domain invariant features whose distribution is similar in source and target domain. They attempt to minimize the discrepancy between domain-specific latent feature representations. However, all the domain alignment approaches can only reduce, but not remove, the domain discrepancy. Therefore, the target samples distributed near the edge of the clusters, or far from their corresponding class centers are most likely to be misclassified by the hyperplane learned from the source domain, as in Chen *et al.* [2019a].

Transfer learning has been successfully applied in computer vision where lower network layers are trained on high-resource supervised datasets like ImageNet to learn generic features, as in Krizhevsky *et al.* [2012], and are then fine-tuned on target tasks, leading to impressive results for image classification and object detection, as shown by Donahue *et al.* [2014], Sharif *et al.* [2014]. Following the successful practice of pre-trained models for computer vision tasks, high-level contextualized language models pre-trained on unlabeled large text corpus and fine-tuned for a given specific task have recently been proposed in NLP with great results. Howard and Ruder [2018] propose ULMFiT, the first to propose fine-tuning with pre-trained language model, showcasing the effectiveness of discriminative fine-tuning, and gradual unfreezing for retaining prior knowledge and circumventing catastrophic forgetting during fine-tuning. There are two existing strategies for applying pre-trained language representations to downstream tasks: feature-based and fine-tuning. The feature-based approach, such as ELMo proposed by Peters *et al.* [2018], uses tasks-specific architectures that include the pre-trained representations as additional features. Many fine-tuning approaches, such as the Generative Pre-trained Transformer (OpenAI GPT) proposed by Radford *et al.* [2018] and the Bidirectional Encoder Representations from Transformers (BERT) proposed by Devlin *et al.* [2019] introduce minimal task-specific parameters, and are trained on the downstream tasks by simply fine-tuning the pre-trained parameters.

Among the unsupervised pre-training methods for language models in the literature, the two most successful pretraining objectives are autoregressive (AR) language modeling that seeks to estimate the probability distribution of a text corpus with an autoregressive model, as in Peters *et al.* [2018], and Radford *et al.* [2018], and autoencoding (AE) language modeling that aims to reconstruct the original data from corrupted input, as in Devlin *et al.* [2019]. Yang *et al.* [2019] proposes the XLNet, a combination of AR and AE language modeling where it can capture dependencies beyond the input sequence limit and process bidirectional contexts at the same time.

4.3. Bidirectional pre-trained transformer language models

4.3.1. Transformer

Before the introduction of Transformers, previous state-of-the-art sequence modelling approaches in NLP relied mostly on recurrent neural networks (RNN), such as Long Short-Term Memory (LSTM) and gated RNN. However, the recurrent models' inherent sequential nature stymies parallelization during training and limits its ability to contextualize longer input sequences. Kim *et al.* [2017] shows that attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences.

The Transformer is first introduced by Vaswani *et al.* [2017] to improve the speed of training models for neural machine translations using the attention mechanism. Its architecture reduces sequential computation with multiple self-attention heads. In order to compute a representation of an input sequence, self-attention mechanism associates different positions of the sequence. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. The original Transformer has encoder-decoder structure, with the encoder mapping an input sequence to a sequence of continuous representations, which is used by the decoder to generate an output sequence one element at a time. Each of the encoder and the decoder consists of 6 identical layers, with each containing two sub-layers of 8 parallel self-attention heads and a fully connected feed-forward neural network.

The input representation to the first encoder layer is a concatenation of WordPiece embeddings, as in Wu *et al.* [2016], and positional embeddings generated from the input sequence. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Specifically, given an embedded vector x for an input sequence, we create a Query, Key, and Value vector for each input embedding token by multiplying the embedding by three learned matrices W^Q, W^K, W^V respectively. For parallel computation, we stack the Query, Key and Value vectors into matrices Q, K, V. Then the self-attention function is given by:

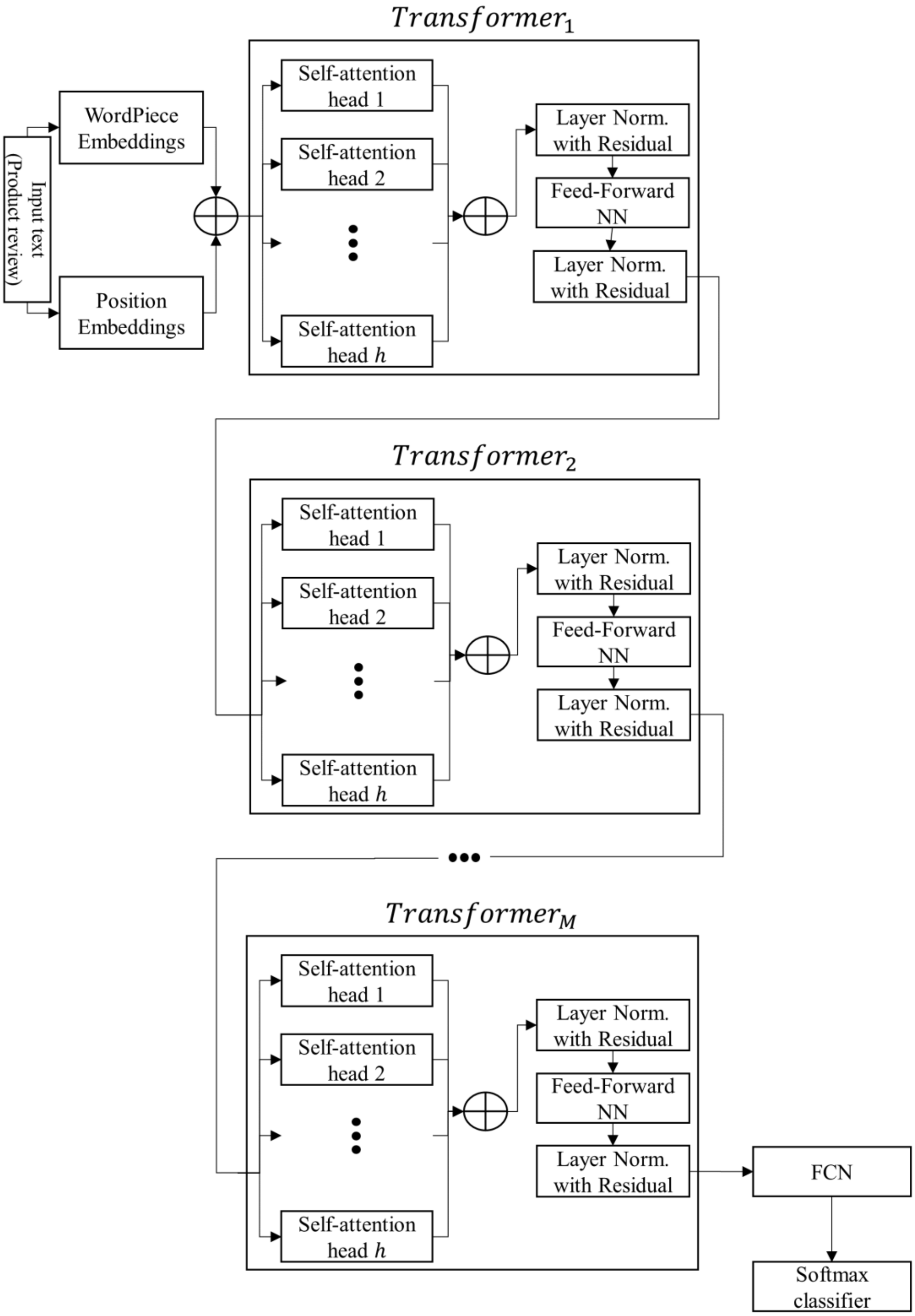


Figure 4.1. Transformer encoder architecture.

$$Attention(x) = Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where d_k is the dimension of queries and keys. The Transformer performs such self-attention function in parallel with multiple attention heads by projecting the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. Attention function is performed in parallel on each of these projected versions of queries, keys and values, resulting d_v-dimensional output values.

$$MultiHead(x) = MultHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O,$$

where $head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$, $Concat$ is the concatenation function, the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ with $d_{model} = d_k h$.

Each Transformer layer consists of two sub-layers. The first sub-layer is the multi-head attention and its normalized output is fed to the second sub-layer of fully connected feed forward network. The activation function for the feed forward networks is ReLU. Formally, the hidden states of Transformer with M number of Transformer layers are calculated as follows:

$$Tr_m(x) = norm\left(Att(x) + FFN\left(att(x)\right)\right),$$

where

$$Att(x) = norm\left(x + MultiHead(x)\right)$$

$$FFN(x) = \max(0, xW_1 + b_1)\, W_2 + b_2,$$

with $norm$ as the normalization function with linear connection following [Ba et al. 2016], FFN a fully connected feed forward network, W_1 and W_2 are the weights of the first and second fully connected networks with b_1, b_2 as bias values, and $m \in M$. These fully connected networks have separate weight parameters for each encoder layer. Each encoder layer passes its output as an input to the next encoder layer, with the final encoder layer producing the final encoded representation for fine-tuning. Figure 4.1 shows the architecture of the Transformer's layers. In the original Transformer in Vaswani *et al.* [2017], the layer size M is 6 and the multi-head h is 8.

4.3.2. BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is built upon recent works in pre-training contextual representations such as ELMo and ULMFiT, but these models are either unidirectional or shallowly bidirectional, meaning contextualized representation of a word only considers the words to its left or to its right. BERT, on the other hand, has deeply bidirectional contextualization that combines the representations of both left-context and right-context models. Its model architecture is a multi-layer bidirectional Transformer encoder based on the original Transformer model proposed in Vaswani *et al.* [2017]. The BERT model retains only the encoder part of the original model, without any decoder. It has 12 identical encoder layers, with each having two sub-layers of 12 parallel attention head and also a fully connected feed-forward network.

For pre-training, unlike ELMo and OpenAI GPT that use left-to-right or right-to-left language models, BERT uses two unsupervised prediction tasks. First is next sentence prediction task, where two sentences (*A*, *B*) are selected from the text corpus and a classifier is trained to predict whether *B* actually follows *A*. 50% of the time *B* is the actual next sentence that follows *A*, and 50% of the time it is a random sentence from the corpus. The second task is the Masked Language Model task, where they mask some percentage of the input tokens at random, and then predict only those masked tokens. Specifically, given a text sequence $x = [x_1, \ldots, x_T]$, BERT first constructs a corrupted version $\hat{x}$ by randomly setting a 15% of tokens in x to a special symbol '*MASK'*. If denote the original masked token as $\bar{x}$, then the training objective is to reconstruct $\bar{x}$ from $\hat{x}$:

$$\max_{\theta} \log p_\theta\,(\bar{x}|\hat{x}) \backslash\approx \sum_{t=1}^{T} m_t \log p_\theta(x_t|\hat{x}) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_\theta(\hat{x})_t^\top e(x_t)\right)}{\sum_{x'} \exp\left(H_\theta(\hat{x})_t^\top e(x')\right)},$$

where $m_t = 1$ indicates token x_t is masked, $e(x)$ denotes the embedding of x and H_θ is a Transformer that maps a text sequence x of length$T$ into a sequence of hidden vectors $H_\theta(x) = [H_\theta(x)_1, H_\theta(x)_2, \ldots, H_\theta(x)_T]$. Note that the $\approx$ sign indicates that when calculating $p_\theta(x_t|\hat{x})$, BERT makes an independence assumption that all masked tokens $\bar{x}$ are separately constructed. Devlin *et al.* [2019] shows that the biggest advantage of this training objective is it allows the model simultaneous access to the contextual information on both sides of a token. BERT is the first fine-tuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming many systems with task-specific architectures and advances the state-of-the-art for eleven NLP tasks.

4.3.3. XLNet

BERT has achieved strong performances across multiple tasks, but had the following major flaws:

- The original Transformer architecture can capture context within the specified maximum input sequence length. If a document is longer than the specified length, it would be divided into segments with each of them being processed by the model independently from scratch without any connection between them.
- BERT is trained to predict tokens replaced with the '*MASK'* symbol. However, this '*MASK'* token never appears in downstream tasks, which creates a discrepancy between pre-training and fine-tuning.
- BERT makes predictions for the masked tokens with assumption that there is no dependencies between these masked tokens, which is bit over-simplification and can cause reduced number of dependencies that BERT can learn at once.

XLNet solves BERT's first flaw of input length context constraint with the architecture of Transformer-XL proposed by Dai *et al.* [2019], which itself is a modification upon the original Transformer framework. Transformer-XL introduces *Recurrence Mechanism* and *Relative Positional Encoding* to the Transformer architecture to capture long-term dependencies for documents that are longer than the maximum allowed input length. With *Recurrence Mechanism*, the hidden state sequence computed for the previous segment is fixed and cached to be reused as an extended context when the model processes the next new segment. Although the gradient still remains within a segment, this additional input allows the network to exploit information in the history, leading to an ability of modeling longer-term dependency and avoiding context fragmentation. *Relative Positional Encoding* encodes position of a context in relative distance from the current token at each attention module, as opposed to encoding position statically only at the beginning like in BERT. This is done so to accommodate the Recurrence Mechanism and avoid having tokens from different segments having the same positional encoding.

Despite its ability to capture long-term dependencies, Transformer-XL still only holds unidirectional context, i.e., predicts the current token based on the given sequential context on its left or its right side only. XLNet solves the issue of unidirectional context, without using '*MASK'* symbol as in BERT, by introducing a language modeling objective called *Permutation language modeling* that predicts a current token based on the given preceding context just like traditional language model. However, instead of predicting tokens in sequential order, tokens are predicted following a random permutation order. One problem with this objective is the computational high expense and slow convergence if we to go through every permutation. Hence to reduce the optimization difficulty, only the last tokens in a factorization order is chosen for training. Formally, let Z_T be the set of all possible permutations of the length T index sequence $[1,2,\ldots,T]$ with z_t and $z_{<t}$ denoting the t-th element and the first $t-1$ elements of a permutation $z \in Z_T$. To choose the tokens in a factorization order, z is split into a non-target subsequence $z_{\leq c}$ and a target subsequence $z_{>c}$, where c is the cutting point. Then the permutation language modeling objective is to maximize the log-likelihood of the target subsequence conditioned on the non-target subsequence as follows:

$$\max_{\theta} \mathbb{E}_{z\sim Z_t}[\log p_\theta(x_{z_{<c}}|x_{z_{\leq c}})] = \mathbb{E}_{z\sim Z_t}\left[\sum_{t=c+1}^{|z|} \log p_\theta\left(x_{z_t}\middle|x_{z_{<t}}\right)\right] =$$

$$\mathbb{E}_{z\sim Z_t} \log \frac{\exp\left(e(x)^\top g_\theta(x_{z_{<t}}, z_t)\right)}{\sum_{x'} \exp\left(e(x')^\top g_\theta(x_{z_{<t}}, z_t)\right)}$$

where $e(x)$ denotes the embedding of x input sequence, $g_\theta(x_{z_{<t}}, z_t)$ denotes a new type of representations which additionally take the target position z_t as input. To compute $g_\theta(x_{z_{<t}}, z_t)$, XLNet introduces a scheme called *Two-Stream Self-Attention* that uses two sets of hidden representations:

- The *content stream* $h_\theta(x_{z_{\leq t}})$, or h_{z_t} for short, is same as the hidden states in the original Transformer. This representation encodes both the context and x_{z_t}.
- The *query stream* $g_\theta\left(x_{z_{<t}}, z_t\right)$, or g_{z_t} for short, only has the contextual information x_{z_t} and the position z_t, without any knowledge of the content x_{z_t}.

The language model is trained to predict each token in the sentence using only the query stream. The content stream is used as input to the query stream. During fine-tuning, the query stream is thrown away and the input data is represented with the content stream. Formally, for each self-attention layer $m = 1,2,\dots,M$, the two streams of representations are updated with shared set of parameters as follows:

$$g_{z_t}^{m} \leftarrow Attention\big(Q = g_{z_t}^{m-1}, KV = h_{z_t}^{m-1}; \theta\big),$$

$$h_{z_t}^{m} \leftarrow Attention\big(Q = h_{z_t}^{m-1}, KV = h_{z_{\leq t}}^{m-1}; \theta\big),$$

where Q, K, V denote the query, key, value in an attention operation. The update rule of the content stream is same as the original Transformer self-attention.

4.3.4. Fine-tuning for cross-domain sentiment classification

We shall fine-tune the pre-trained Transformer models, BERT and XLNet, with a labeled sentiment data from a selected source domain and measure its performance in predicting the sentiment polarity of other domain's sentiment data. To measure and compare the effectiveness of BERT and XLNet for cross-domain sentiment classification, on top of the pre-trained models we will only add one fully connected feed-forward network that consists of two linear transformations with GELU [Hendrycks and Gimpel, 2016] activation in between. Given source domain labeled data X_S, we calculate the probability distributions of input sequences using a softmax activation function.

$$f(x_i) = GELU(Tr_M(x_i)W_1 + b_1)W_2 + b_2$$

$$p(y_i|x_i) = \frac{e^{f(x_i)}}{\sum_j e^{f(x_j)}},$$

where $Tr_M(x_i)$ is the output from the last Transformer layer M of either BERT or XLNet for the input sequence $x_i \in X_S$. W_1 and W_2 are the weights of the first and second linear transformations with b_1, b_2 as bias values. The cost function to minimize is the cross-entropy loss as follows:

$$\mathcal{L} = -\sum_{t=1}^{T} (y_i \log p(y_i|x_i) + (1 - y_i)\log\big(1 - p(y_i|x_i)\big),$$

where N is the total number of samples in the current batch, y_i is the given label of the input sequence (1 for positive review and 0 for negative review) and $p(y_i|x_i)$ is the probability of the input sequence being positive.

After fine-tune training, we apply the learned models on the target domain and predict the sentiment binary values using softmax function $p(y_i|x_i)$ with the trained parameters where $x_i \in X_T$. Figure 4.2 illustrates the framework of using pre-trained language models for cross-domain sentiment classification.

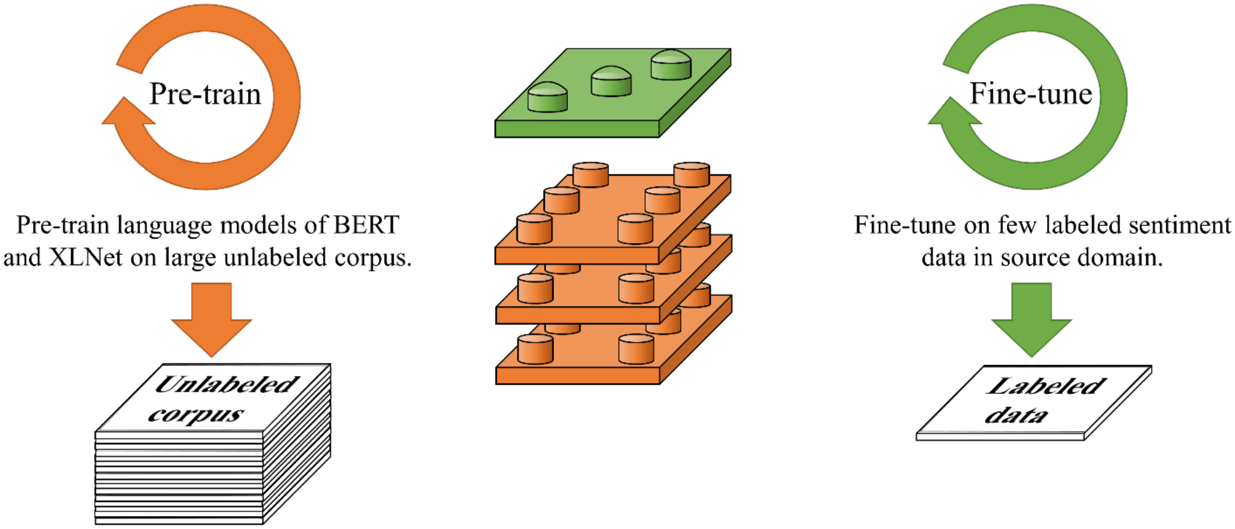

Pre-train
Pre-train language models of BERT and XLNet on large unlabeled corpus.
Unlabeled corpus
Fine-tune
Fine-tune on few labeled sentiment data in source domain.
Labeled data

Figure 4.2. Mid-level common feature representation with pre-trained language models for CDSC.

4.4. Experiments

4.4.1. Dataset

Our experiments are conducted on the Amazon reviews dataset from Blitzer *et al.* [2007] that has been widely used in the literature for cross-domain sentiment classification. The dataset contains reviews from five product types (i.e. domains): Books, DVD, Electronics, Kitchen and Video. There are 6000 labeled review data for each domain with 3000 positive reviews (higher than 3 stars) and 3000 negative reviews (lower than 3 stars). Following the convention in Pan *et al.* [2010], we construct 20 cross-domain sentiment classification tasks. We fine-tune on the pre-trained BERT-Large[9] and XLNet-Large[10] language models with differing number of labeled data from the selected source domain and test the trained models on the other domain data.

[9] https://github.com/google-research/bert

[10] https://github.com/zihangdai/xlnet

4.4.2. Pre-training

For our experiment we use the latest pre-trained cased BERT-Large model, referred to simply as BERT henceforth, with new pre-processing technique called Whole Word Masking where all of the tokens corresponding to a word are masked at once, instead of masking those tokens belonging to a word individually. It has 24 Transformer layers with 4096 hidden dimensions, 16 attention heads and a total of 340M parameters. For the pre-training, BERT uses the concatenation of BookCorpus (800M words) [Zhu *et al.* 2015] and English Wikipedia (2,500M words) as pre-training data. BERT is pre-trained with batch size of 256 sequences with each sequence containing maximum of 512 tokens for 1,000,000 steps, which is approximately 40 epochs over the 3.3 billion word corpus.

For pre-training data, in addition to the BookCorpus and English Wikipedia datasets, cased XLNet-Large model, referred to simply as XLNet henceforth, uses Giga5 (16GB text) [Parker *et al.* 2011], ClueWeb 2012-B [Callan *et al.* 2009] and Common Crawl[11] as part of its pre-training data. ClueWeb2012-B and Common Crawl articles are filtered out and after tokenization with SentencePiece, introduced by Kudo and Richardson [2018], the total pre-training data for XLNet amounts to 32.89B subword pieces, which is an order of magnitude greater than the pre-training data used for BERT. XLNet's architecture has, similar to BERT, 24 Transformer layers with 4096 hidden dimensions and 16 attention heads. XLNet is pre-trained with batch size of 2048 and sequence length of 512 for 500,000 steps.

4.4.3. Implementation details

For fine-tune training of the language models, the hidden dimensions of the fully connected networks following the last layer of the Transformers is 1024. for cross-domain sentiment classification. The dropout probability is kept at 0.1. For the input, the maximum sequence length is set to 256 with batch size of 32. The learning rate is 2e-5 and optimization is done with Adam optimizer. Training and testing of TensorFlow implementations of BERT and XLNet are performed separately on a single Google Cloud TPU v2 and the total experiment time was over 400 hours for each TPU.

[11] http://commoncrawl.org

For comparison with other state-of-the-arts CDSC methods, the BERT and XLNet models are trained on 6000 labeled data from a source domain for 3000 steps and evaluate the prediction accuracy on all 6000 data of the remaining domains.

In addition, to show BERT and XLNet's effectiveness in low resource transfer learning scenarios, we train the models on different amount of source domain labeled data and test each trained model on all of the other domains. We compare the runtimes of these two language models in the same configuration scenarios with varying number of steps for the training phase and also with different number of samples for the testing phase.

4.4.4. Baselines

The baseline methods included in the comparison are following:

- **DAmSDA** in Ganin *et al.* [2016]: an adversarial network based domain adaptation method that utilizes representations encoded in a 30,000-dimensional feature vector.
- **CNN-aux** in Yu and Jiang [2016}: a CNN model based on the approach proposed by [Kim, 2014]. It jointly trains the cross-domain sentence embedding and the sentiment classifier.
- **AMN** in Li *et al.* [2017]: an adversarial network based method that learns domain-shared representations based on memory networks and adversarial training.
- **HATN** in Li *et al.* [2018]: an attention network with hierarchical positional encoding that focuses on both the word and sentence level sentiments.
- **HANP** in Manshu and Bing [2019]: a hierarchical attention network than can obtain both domain independent and domain specific features at the same time by adding prior knowledge.
- **BERT**: the proposed fine-tuned auto-encoding bidirectional contextualized language model pre-trained on Masked language modeling and the Next sentence prediction tasks.
- **XLNet**: the proposed fine-tuned auto-regressive bidirectional contextualized language model pre-trained on Permutation language modeling task.

We use classification accuracy as our performance metrics, which is defined as follows:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}.$$

Table 4.1. Cross-domain sentiment classification on Amazon sentiment dataset.

Source	Target	DAmSDA	CNN-aux	AMN	HATN	HANP	**BERT**	**XLNet**
Books	DVD	86.12%	84.42%	85.62%	87.07%	88.12%	92.49%	**95.10%**
	Electronics	79.02%	80.63%	80.55%	85.75%	85.81%	93.13%	**95.92%**
	Kitchen	81.05%	83.38%	81.88%	87.03%	88.91%	94.08%	**96.54%**
	Video	84.98%	84.43%	87.25%	87.80%	89.21%	91.75%	**94.54%**
DVD	Books	85.17%	83.07%	84.53%	87.78%	89.18%	93.67%	**95.68%**
	Electronics	76.17%	80.35%	80.42%	86.32%	86.87%	93.25%	**95.17%**
	Kitchen	82.60%	81.68%	81.67%	87.47%	88.54%	94.15%	**96.42%**
	Video	83.80%	85.87%	87.40%	89.12%	91.25%	93.88%	**95.82%**
Electronics	Books	79.92%	77.38%	77.52%	84.03%	85.67%	91.83%	**93.56%**
	DVD	82.63%	79.07%	80.53%	84.32%	85.29%	89.93%	**91.99%**
	Kitchen	85.80%	87.15%	87.83%	90.08%	91.08%	95.37%	**96.79%**
	Video	81.70%	78.78%	82.12%	84.18%	85.96%	89.33%	**91.79%**
Kitchen	Books	80.55%	78.47%	79.05%	84.88%	85.04%	91.74%	**95.29%**
	DVD	82.18%	79.07%	79.50%	84.72%	86.47%	90.34%	**94.44%**
	Electronics	88.00%	86.73%	86.68%	89.33%	90.43%	94.82%	**96.46%**
	Video	81.47%	78.82%	82.15%	84.85%	85.93%	89.82%	**94.31%**
Video	Books	83.00%	81.48%	83.50%	87.10%	88.94%	93.05%	**95.31%**
	DVD	85.90%	85.25%	86.88%	87.90%	88.54%	93.32%	**95.60%**
	Electronics	77.67%	82.32%	79.68%	85.98%	86.11%	92.87%	**95.71%**
	Kitchen	79.52%	81.28%	80.98%	86.45%	87.21%	93.35%	**96.11%**
Average		82.36%	81.98%	82.79%	86.61%	87.76%	92.61%	**95.13%**

4.4.5. Results

In Table 4.1 shows the classification accuracy of various state-of-the-arts methods in comparison to the bidirectional contextualized language models on the cross-domain sentiment classification task. For BERT and XLNet, we report the mean accuracy rate from 10 separate runs using all of the 6000 labeled data available in the source domain. It can be observed that the bidirectional contextualized Transformer language models of BERT and XLNet greatly outperforms the previous state-of-the-arts methods. BERT outperforms previous state-of-the-arts methods by at least 2% accuracy. However, XLNet produces results that further improves the CDSC accuracy by 2.5% in comparison to BERT. XLNet is the only method where all of the prediction accuracy rates are well above above 90%.

The most interesting results are observed in Figure 4.3 and Table 4.2. For BERT and XLNet, we report the mean bootstrapped results from predicting four target domain data with 95% confidence interval from 40 observations where source domain labeled data are selected randomly with replacement. BERT outperforms the previous SOTA methods using around 300 samples or around 20 times less data. XLNet outperforms previous state-of-the-arts methods after fine-tuning only with 50 source domain training samples, i.e., around 120 times less data than the previous SOTA methods. These results prove that pre-trained Transformer language models are very adaptive at capturing context with only few samples and are highly suitable for transfer learning. Also, it can be observed that XLNet is much more efficient at capturing contextualized representations than BERT that it can fine-tune its pre-trained parameters to very quickly pivot towards capturing sentiment polarity in the given sequences. This higher efficiency performance is due to the combination of different pre-training objective function, ability to capture dependencies longer than the sequence length and the larger pre-training datasets. Table 4.2 shows the CDSC accuracy rates with the corresponding margins of error.

Table 4.2. CDSC accuracy rates on different source domain training data size.

		Source domain				
		Books	DVD	Electronics	Kitchen	Video
Source domain training samples amount	10	55.85 (1.50)	56.15 (1.91)	58.27 (1.95)	56.04 (1.33)	52.95 (0.95)
		64.81 (3.25)	72.74 (3.48)	72.30 (3.13)	68.00 (3.09)	62.03 (3.47)
	20	61.08 (2.51)	60.09 (2.39)	61.83 (2.76)	62.06 (2.45)	57.61 (2.18)
		77.65 (3.39)	82.85 (2.54)	79.45 (3.11)	78.21 (3.22)	74.22 (3.38)
	30	66.41 (2.94)	62.37 (2.80)	67.57 (3.25)	66.75 (2.94)	63.43 (3.47)
		83.30 (2.42)	86.38 (1.79)	87.90 (1.08)	85.12 (2.91)	83.70 (2.63)
	40	64.17 (2.56)	70.72 (3.38)	71.71 (3.24)	67.23 (2.98)	60.99 (2.31)
		88.49 (1.04)	91.14 (0.55)	88.80 (0.89)	86.75 (2.32)	86.64 (2.85)
	50	67.29 (3.28)	81.41 (1.54)	72.78 (3.41)	75.26 (3.00)	68.97 (2.40)
		90.02 (0.93)	91.60 (0.57)	88.86 (1.10)	88.08 (1.58)	89.02 (2.14)
	60	75.48 (3.73)	74.54 (2.62)	75.61 (2.97)	70.11 (2.60)	75.38 (2.64)
		91.44 (0.54)	90.88 (1.44)	89.08 (1.17)	87.26 (2.37)	88.88 (1.96)
	70	73.24 (3.06)	78.69 (1.95)	79.13 (2.94)	75.93 (3.04)	76.44 (3.35)
		91.07 (1.08)	92.04 (0.50)	91.07 (0.73)	88.50 (1.72)	90.43 (1.81)
	80	81.16 (2.54)	76.35 (3.65)	81.43 (2.50)	81.41 (2.84)	80.79 (2.18)
		92.01 (0.65)	92.71 (0.40)	90.60 (0.92)	90.30 (1.03)	92.29 (0.55)
	90	75.88 (3.55)	80.74 (1.65)	84.47 (1.62)	83.74 (1.52)	80.80 (2.69)
		91.85 (0.56)	93.52 (0.35)	90.42 (1.21)	90.38 (1.23)	90.52 (1.82)
	100	82.98 (2.97)	85.43 (1.24)	84.57 (2.42)	86.15 (1.02)	79.20 (3.22)
		92.24 (0.55)	93.15 (0.34)	91.48 (0.76)	90.64 (0.93)	91.97 (1.49)
	300	90.17 (0.61)	89.30 (0.97)	89.54 (0.77)	89.10 (0.75)	91.06 (0.33)
		94.28 (0.28)	94.28 (0.29)	92.90 (0.63)	92.98 (0.58)	94.20 (0.30)
	500	91.18 (0.43)	91.66 (0.33)	90.09 (0.67)	90.09 (0.65)	91.60 (0.30)
		94.41 (0.33)	94.79 (0.25)	92.81 (0.69)	93.40 (0.54)	94.84 (0.22)
	1000	92.02 (0.34)	92.26 (0.30)	90.03 (0.79)	90.64 (0.65)	92.31 (0.25)
		94.83 (0.27)	95.19 (0.20)	93.14 (0.69)	93.95 (0.48)	95.00 (0.28)
	2000	92.37 (0.33)	92.91 (0.24)	90.93 (0.74)	91.27 (0.57)	92.77 (0.16)
		95.05 (0.27)	95.48 (0.22)	93.13 (0.90)	94.84 (0.29)	95.31 (0.16)
	4000	92.73 (0.32)	93.47 (0.12)	91.19 (0.77)	91.76 (0.58)	93.05 (0.11)
		95.35 (0.24)	95.76 (0.15)	93.20 (0.76)	95.04 (0.30)	95.61 (0.13)
	6000	92.86 (0.32)	93.73 (0.13)	91.62 (0.74)	91.68 (0.61)	93.15 (0.11)
		95.52 (0.26)	95.77 (0.17)	93.53 (0.68)	95.13 (0.29)	95.68 (0.11)
		88.01	88.96	87.00	86.97	87.70

🟩 **Previous SOTA** □ **BERT** ▨ **XLNet**

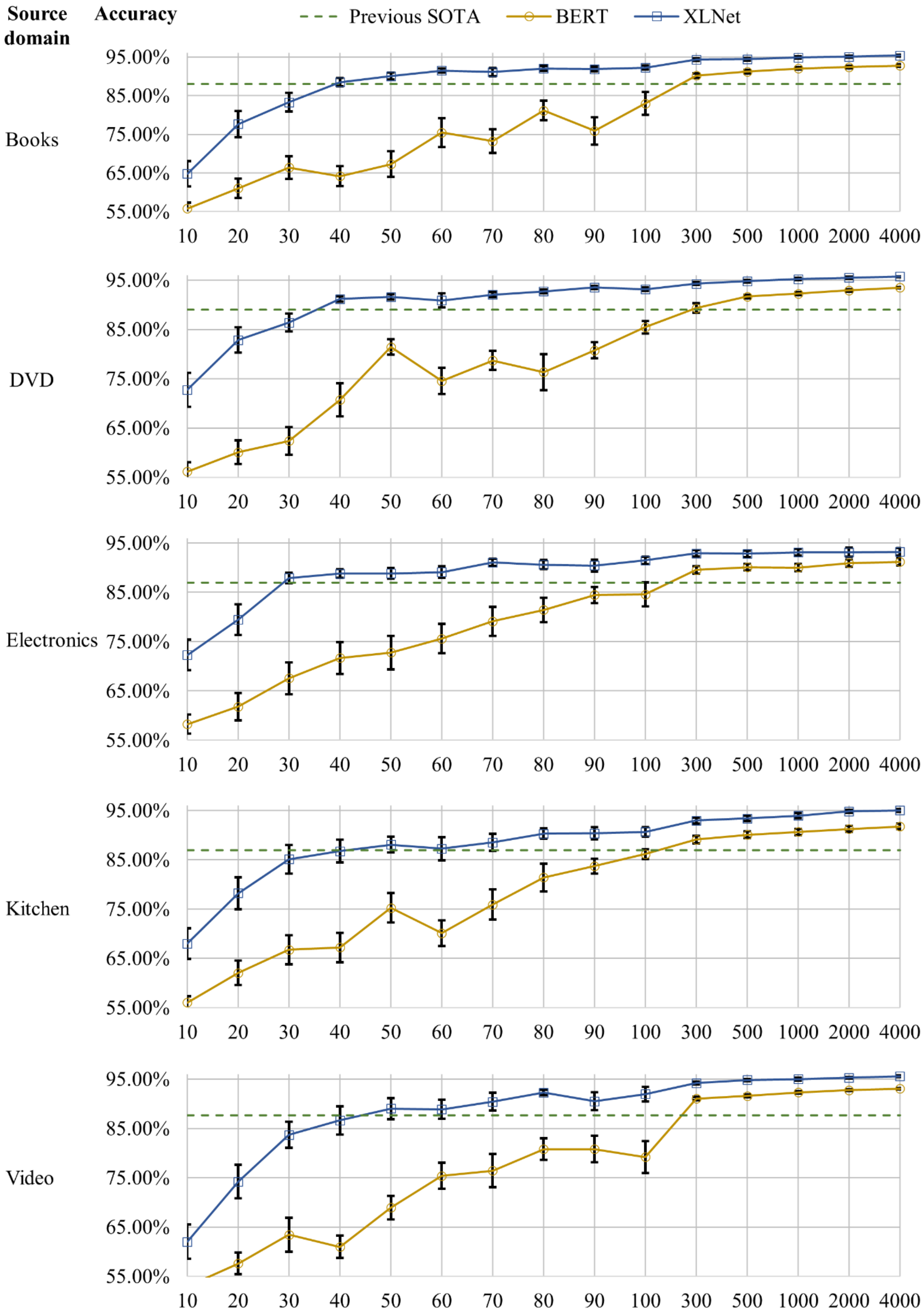


Figure 4.3. CDSC accuracy rates over different source domain labeled training data size. Horizontal axis is the training data size.

Table 4.3. Training run time over different training step numbers.

	300	1000	3000	9000	30000
BERT	458	627	1140	2605	7735
XLNet	499 (+9%)	713 (+14%)	1355 (+19%)	3259 (+25%)	9908 (+28%)

Table 4.4. Test run times over different test data sizes (20, ..., 6000)

	20	50	100	200	500	1000	2000	4000	6000
BERT	85	87	84	85	86	89	91	99	110
XLNet	77 (-9%)	78 (-11%)	79 (-6%)	78 (-8%)	79 (-8%)	81 (-10%)	83 (-9%)	86 (-13%)	90 (-18%)

In Figure 4.4 and Figure 4.5, we compare the runtimes of the two models during fine-tune training and testing. The reported results are the mean duration times from 10 separate runs for each training step size and test data size. The test data are identical for both models and are randomly selected with replacement. We can see that XLNet is more efficient than BERT during testing, on average around 10% less time spent on testing. However, XLNet has shown to be much more resource-hungry when it comes to training. In our case where the main SOTA results are reported

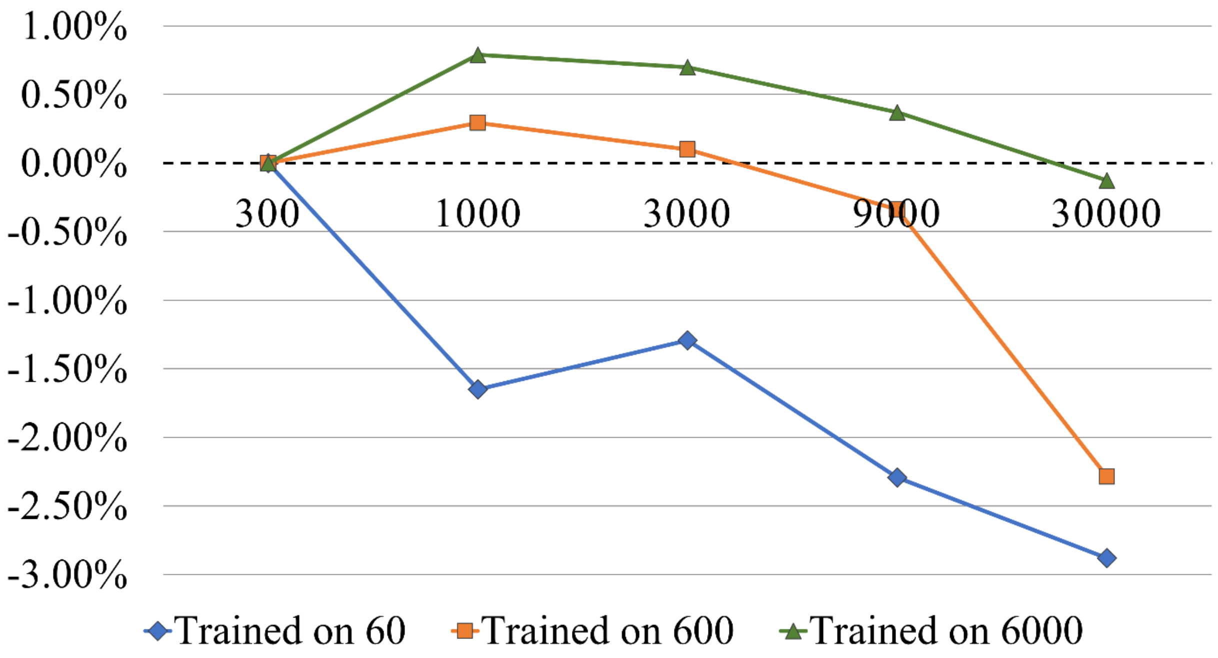


Figure 4.4. Shows how three XLNet models trained with different number of source domain labeled data (60, 600, 6000) performs over increasing number of training steps (300, 1000, 3000, 9000, 30000).

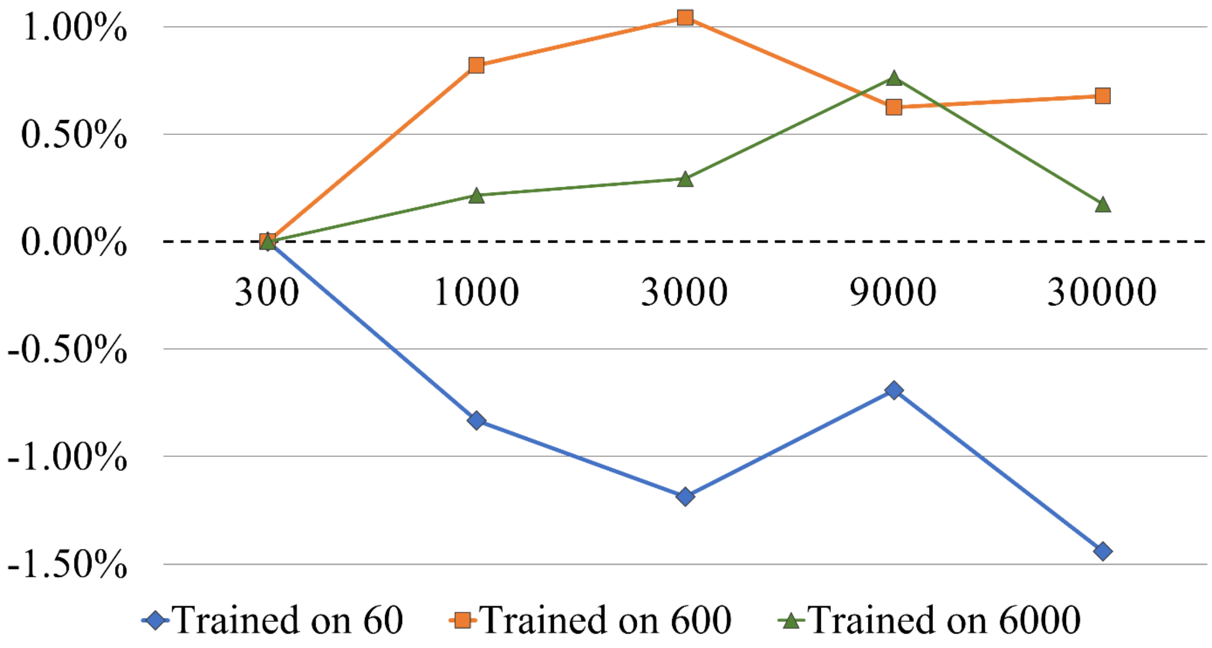


Figure 4.5. Shows how three BERT models trained with different number of source domain labeled data (60, 600, 6000) performs over increasing number of training steps (300, 1000, 3000, 9000, 30000).

from 3000 training steps, XLNet is almost 20% slower than BERT. XLNet's runtime is higher than BERT in training due to its segment recurrence mechanism for capturing context dependencies in documents longer than the maximum input sequence length. However, during testing, this segment recurrence mechanism actually decreases the runtime for XLNet to be less than BERT's because the representations from the previous segments can be reused instead of being computed from scratch as in the case of the standard Transformer.

In Figure 4.4 and Figure 4.5, we evaluate the effect of different number of training steps (300, 1000, 3000, 9000, 30000) on the CDSC accuracy rate. BERT and XLNet models are fine-tuned on varying amounts of labeled data (60, 600, 6000) from a source domain ('Books') and tested on all 6000 data of a target domain ('Video'). The results are the mean accuracy rate change over 10 separate runs for each step size and fine-tune training data size. We observe that in general and at least in the context of CDSC, there is a noticeable trade-off between amount of training data and training step size. For both models fine-tuned with only few labeled data, e.g., 60, the accuracy rate drops off immediately when trained for longer than the baseline 300 steps, meaning it overfits the source domain. For XLNet, there is recognizable decrease in performance after 1000 training steps for all models. We believe that XLNet captures the necessary contextual dependencies earlier in the training steps, when compared to BERT, and longer it trains, the parameters more overfit the source domain. Therefore, even though XLNet runs slower than BERT, it learns more quickly with fewer training steps.

4.5. Conclusion

In this chapter, we apply the bidirectional contextualized Transformer language models of BERT and XLNet on cross-domain sentiment classification task. Due to their unsupervised pre-training tasks utilizing large unlabeled datasets and their self-attention Transformer mechanisms, BERT and XLNet both greatly outperforms the previous state-of-the-arts methods for CDSC task. When compared closely, XLNet outperforms BERT on all CDSC tasks. XLNet is very efficient in capturing context and achieves state-of-the-arts results with only using 50 fine-tune training samples, i.e., around 120 times fewer data than the previous high-performing CDSC methods trained on. XLNet's better prediction accuracy is mostly due to its novel pre-training objective, ability to capture long-term dependencies, and larger pre-training dataset. XLNet is more resource-hungry than BERT, but learns contextual data much quicker than BERT with fewer fine-tuning steps.

Chapter 5. High-level common feature representation with parallel corpus for cross-lingual sentiment classification

5.1. Introduction

Despite its increasing research interest, most of the current efforts in sentiment classification have been done so in a monolingual scenarios, and specifically in English due to its dominance across the Web and availability of labeled datasets, as shown in Zhang *et al.* [2018]. Although there are works done in other languages such as Chinese in Peng *et al.* [2018], German in Cieliebak *et al.* [2017], Russian in Rogers *et al.* [2018], Japanese in Niitsuma *et al.* [2018], Nio and Murakami [2018], Bataa and Wu [2019], and others, they mostly rely on very few language-specific datasets and non-English sentiment classification works have been greatly lagging behind the general progress of English based works. This results in high inequality in services provided online for non-English users, even though around 75% of the Web users in April 2019 were non-English speakers[12].

To alleviate the lack of non-English sentiment datasets, transfer learning has been employed for *cross-lingual sentiment classification*, where sentiment knowledge is transferred from a source language with sufficient labeled data to a scarce resource target language, as in Lo *et al.* [2017]. The biggest challenge for cross-lingual sentiment classification is obtaining parallel corpus as a point of reference between languages to learn bilingual representations for the target language sentiment classification task. Most of the current cross-lingual approaches, described in Araújo *et al.* [2020], use off-the-shelf machine translation systems such as Google Translate[13] and Microsoft Bing Translator[14] to convert the non-English text into English and apply sentiment classification methods that have been developed using English sentiment datasets. Even though it is a very sensible and easy to implement such approaches, Chen *et al.* [2017] points out that the machine

[12] https://www.internetworldstats.com/stats7.htm (accessed December 20th, 2019)
[13] https://translate.google.com
[14] https://www.bing.com/translator

translation systems have difficulty in accurately capturing the *language discrepancy,* i.e. sentiments expressed in different patterns across languages. The machine translation systems are generally good at translating sentimental expressions that are similar across languages, such as positive sentiments of “pleasant” for English and “楽しい” for Japanese, but suffer when translating language specific expressions. For example, “湯水のように使う” in Japanese meaning “to use wastefully” is translated as “Use as hot water” in Google Translate, which loses the expressive meaning. Chen *et al.* [2019b] claims that most of these contextual confusions come from the fact that these machine translation systems are trained to capture similar patterns across languages, rather than patterns unique to languages and thus fail to retain language-specific sentiment knowledge. Another reason for such misunderstanding is the availability of very few English-Japanese parallel sentiment corpus or datasets publicly available to bridge the language discrepancy gap between English and Japanese.

Multi-language dataset proposed in Prettenhofer and Stein [2010] consists of Amazon product reviews of three product categories (books, DVDs, music) written in four language: English, German, French and Japanese. Each product category contains balanced training and test set of 1000 positive and 1000 negative reviews for each language. For each non-English review text, there is a corresponding English translation retrieved from Google Translate. Multilingual Amazon dataset is the most widely used dataset for cross-lingual sentiment classification. However, it is very task specific and difficult to be applicable for sentiment classification outside of product reviews. The National Institute of Information and Communications Technology (NICT) has created the Japanese-English bilingual corpus of Wikipedia Kyoto corpus [Kyoto corpus] by manually translating Japanese Wikipedia articles (related to Kyoto) into English. It has 500,000 pairs of manually-translated sentences concerning following categories: school, railway, family, building, Shinto, person name, geographic name, culture, road, Buddhism, literature, title, history, shrines and temples, and emperor. Kyoto corpus mostly consist of factual information and sentiment expressions are seldom used. Pryzant *et al.* [2017] proposed Japanese-English Subtitle Corpus (JESC) consisting of 3.2 million examples assembled by crawling and aligning various films’ subtitles found on the web. This is largest bilingual corpus for English and Japanese that contains various sentiment expressions.

In chapter 4, we have discussed how the bidirectional contextualized pre-trained language models such as BERT and XLNet are able to capture the contextual direction necessary for downstream classification tasks with only few samples and have outperformed previous the state-of-the-arts methods by significant margins for several NLP tasks. The models' extensive pre-training processes help them gather state-of-the-arts contextual understanding of the target language and during fine-tuning for downstream task such as sentiment classification, these models use the data for determining the direction of the contextual information necessary for classification, rather than gathering new understanding of the language. Although we have demonstrated through experiments in chapter 4 that XLNet is superior to BERT in terms of understanding English language sentiment text, one big advantage of BERT is its multi-lingual pre-trained models that can understand 104 languages and available to be fine-tuned for NLP tasks in those languages.

Motivated by the need for better understanding of sentiment expressions between languages without requiring any labeled data in the target language, we propose a BERT-based Unsupervised Cross-Lingual (BUCL) sentiment classification framework without any machine translation and a novel non-task specific English-Japanese parallel corpus that will provide the necessary sentiment knowledge mappings between the languages. The proposed *teacher-student* framework consists of two components: 1) a *teacher* BERT model trained on a source language (English) labeled sentiment data, and 2) a *student* BERT trained on the proposed parallel corpus to classify target language (Japanese) sentiment data. Our proposed non-task specific parallel corpus consists of translated English-Japanese subtitles for U.S. television series "Mad men".

Our main objective is to demonstrate high-level transfer learning for cross-lingual sentiment classification can be efficiently performed without any machine translation and only using high-level contextual mapping. The main contributions are as follows:

- we propose a novel cross-lingual sentiment classification framework and a English-Japanese parallel sentiment corpus.
- we verify the effectiveness of our approach in comparison with state-of-the-arts in cross-lingual sentiment classification and other non-task specific Japanese-English parallel corpora (JESC, Kyoto).

5.2. Related works

Due to the advance of research works done in English sentiment datasets, many multilingual works are some extensions of strategies used for English sentiment classification. Additionally, with the multi-lingual support improvements in free translation tools such as Google Translate and Microsoft Bing Translator, most cross-lingual sentiment classification methods utilize these machine translation systems. Shalunts *et al.* [2016] use machine translation to convert the sentiment text from German, Russian and Spanish into English and apply sentiment learning methods developed for English corpus. Araújo *et al.* [2020] retrieve English translation of non-English sentiment text from the machine translation systems of Google Translate, Microsoft Bing Translator and Yandex Translate[15] and evaluate the prediction performance of 13 English based sentiment classification methods across 14 different languages: Chinese, German, Spanish, Greek, Croatian, Hindi, Czech, Dutch, French, Haitian Creole, English, Portuguese, Russian, and Italian. Their results show that the automatic translation of the input from a non-English language to English and the subsequent analyze in English methods can be a competitive strategy if the suitable sentiment classification method is properly chosen. Chen *et al.* [2019b] employ emojis in combination with machine translation of non-English sentiment text to learn cross-lingual sentiment patterns. Zhou *et al.* [2019] propose a Sparse Heterogeneous Feature Representation (SHFR) approach to learn a feature mapping for Heterogeneous Domain Adaptation (HDA) with application to cross-lingual text classification. They formulate the problem of learning the feature mapping between domains as a Compressed Sensing problem and propose Error Correcting Output Correcting (ECOC) scheme to generate binary classifiers and leverage the weight vectors of the classifiers learned in the source and target domains to estimate a sparse feature mapping. They evaluate their approach on Amazon cross-lingual sentiment dataset. Chen *et al.* [2019c] leverage adversarial networks to learn language-invariant features and allows the target language to dynamically and selectively leverage language-specific features through a probabilistic attention-style mixture of experts mechanism. They combine their method with unsupervised cross-lingual word embeddings, proposed in Lample *et al.* [2018] and in Chen and Cardie [2018], to perform cross-lingual transfer learning. They also evaluate their approach on Amazon cross-lingual sentiment dataset.

[15] https://translate.yandex.com/

5.3. Bert-based unsupervised cross-lingual (BUCL) sentiment classification framework

We formulate our cross-lingual sentiment classification task as an unsupervised transfer learning problem. We are given a source language sentiment domain $\mathcal{D}_S = \{\mathcal{X}_S, P_S(X_S)\}$, with its sentiment classification task to train the teacher model $\mathcal{T}_S = \{\mathcal{Y}|P_S(Y_S|X_S)\}$ consisting of training data pairs of $x_i \in \mathcal{X}_S$ and $y_i \in \mathcal{Y}_S$ and a target language sentiment domain $\mathcal{D}_T$ with its sentiment classification task $\mathcal{T}_T = \{\mathcal{Y}|P_T(Y_T|X_T)\}$ consisting of only data $x_i \in \mathcal{X}$ available and the corresponding $y_i \in \mathcal{Y}_T$ unknown. Here we assume $\mathcal{X}_S \neq \mathcal{X}_T$, $P_S(Y_S|X_S)\} \neq P_T(Y_T|X_T)$ and $\mathcal{Y}_S = \mathcal{Y}_T$, i.e. they have different feature representations and conditional probability distributions but same label space. Then the objective of our *unsupervised cross-lingual sentiment classification* approach is to learn the conditional probability distribution $P_T(Y_T|X_T)$ in $\mathcal{D}_T$ by transferring the knowledge learned from $\mathcal{D}_S$ with task $\mathcal{T}_S$ to $\mathcal{D}_T$ through an intermediary mapping domain $\mathcal{D}_M = \{\mathcal{X}_{SM}, P_M(X_{SM}), \mathcal{X}_{MT}, P_M(X_{MT})\}$ where $\mathcal{X}_{SM} = \mathcal{X}_S$, $\mathcal{X}_{MT} = \mathcal{X}_T$, and $P_M(X_{SM}) = P_M(X_{MT})$. We shall create a sentiment classification task $\mathcal{T}_{SM} = \{\mathcal{Y}_{SM}|P_S(Y_{SM}|X_{SM})\}$ where we apply learned source domain model P_S on X_{SM}. Since $P_M(X_{SM}) = P_M(X_{MT})$, it follows that $y_{SM_i} = y_{MT_i}$ where $y_{SM_i} \in \mathcal{Y}_{SM}$ and $y_{MT_i} \in \mathcal{Y}_{MT}$, i.e. corresponding pair of texts in the parallel corpus have the same label values. Afterwards, we create another task to train the student model $\mathcal{T}_{MT} = \{\mathcal{Y}_{MT}|P_T(Y_{MT}|X_{MT})\}$ consisting of training data pairs of $x_{MT_i} \in X_{SM}$ and $y_{MT_i} \in Y_{MT}$, i.e. we train a target language sentiment classification task using the freshly labeled target language data in the parallel corpus. Finally, we have the trained target model P_T to acquire the target labels with $P_T(Y_T|X_T)$. In our approach, P_S and P_T are separate BERT models.

The proposed framework is illustrated in Figure 5.1.

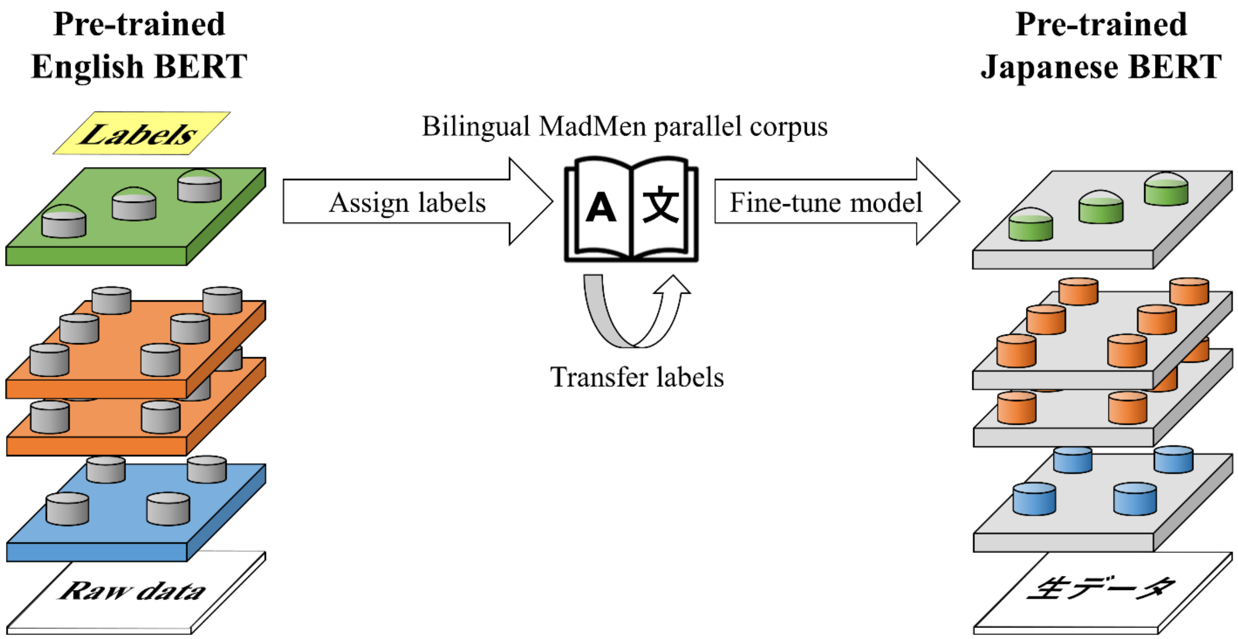


Figure 5.1. Bert-based Unsupervised Cross-Lingual (BUCL) transfer learning framework.

To create the sentiment mapping between the languages, we needed a parallel corpus that contains sufficient level of sentiment expressions. As mentioned before, we have found three publicly available Japanese-English parallel text corpora: Amazon dataset in Prettenhofer and Stein [2010], Kyoto corpus in [Kyoto corpus], and JESC corpus in Pryzant *et al.* [2017]. However, our wish to create new parallel sentiment corpus is motivated by the following reasons:

- Amazon dataset is too task-specific and would not generalize well outside of product review sentiment data.
- Kyoto corpus consist of mostly factual text information and lack sentiment expressions.
- JESC is on the opposite sentiment spectrum of Kyoto. It contains too many noisy sentiment expressions (curses, unrealistic or too personal conversations, etc.) and without any categorizations, difficult to know which parts are suitable for what downstream sentiment classification tasks.

Our intuition behind creating a new parallel sentiment corpus is to be somewhere in between the opposite sentiment spectrums of Kyoto and JESC, i.e. have a dataset that contains useful, realistic, sentiment expressions that can be utilized for various downstream sentiment classification tasks. Therefore, we propose the MadMen English-Japanese sentiment text corpus. It consists solely of

English-Japanese subtitles for American period drama television series “Mad Men”[16]. It was obtained from a free and open subtitle repository[17]. We manually aligned the subtitle texts without any grammatical corrections. Currently we have a total of 1000 pairs of subtitle texts, representing the first two episodes of the show’s first season.

5.4. Experiments

5.4.1. Baselines

We evaluate the effectiveness of our BUCL sentiment classification approach and the MadMen parallel corpus in comparison with 12 state-of-the-arts cross-lingual sentiment classification methods and with other Japanese-English parallel corpora.

All of the following 12 baseline methods have reported experiment results on the multilingual Amazon sentiment dataset with English as the source and Japanese as the target language.

1) **MT-BOW** in Prettenhofer and Stein [2010]: learns a linear classifier on the source language training data, retrieves the English translations for target language bag of words sentence from Google Translate, and applies the learned model on the target data.
2) **CL-SCL** in Prettenhofer and Stein [2010]: a Cross-Lingual Structural Correspondence Learning method that learns cross-lingual feature space by finding structural correspondence among the words from both languages via pivot words.
3) **HeMap** in Shi *et al.* [2010]: a Heterogeneous Spectral Mapping method that learns dense orthogonal mappings to project data from both source and target domain into based on spectral embedding using only unlabeled data.
4) **DAMA** in Wang and Mahadevan [2011]: a heterogeneous Domain Adaptation with Manifold Adaptation method that uses manifold regularization to align different feature spaces into a latent space. It uses labeled data from both domains to construct similarity constraints.
5) **ARC-t** in Kulis *et al.* [2011]: an Asymmetric Regularization Cross-domain Transformation method that learns asymmetric non-linear transformation using metric learning. Also creates similarity constraints using labeled data from both domains.

[16] https://en.wikipedia.org/wiki/Mad_Men

[17] https://www.opensubtitles.org

6) **HFA** in Duan and Tsang [2012]: a Heterogeneous Feature Augmentation approach extracts augments heterogeneous features of source and target domains by extracting common features in both domains using max-margin method.
7) **CL-RL** in Xiao and Guo [2013]: a semi-supervised Cross-Lingual Representation Learning method that learns cross-lingual discriminative distributed representations of words where part of the word vector is shared among languages. Similar to CL-SCL, uses labeled data from both source and target domains.
8) **Bi-PV** in Pham *et al.* [2015]: a Bi-Lingual Paragraph Vector method that learns bilingual distributed representations for phrases and sentences as a whole from unannotated parallel documents. Kyoto corpus is used as the parallel document.
9) **UMM** in Xu and Wan [2017]: an Universal Multilingual Model that learns multilingual sentiment-aware word embeddings based on the labeled reviews in English and unlabeled parallel corpus. Kyoto corpus is also used as the parallel corpus.
10) **CLDFA** in [Xu and Yang, 2017]: a Cross-Lingual Distillation with Feature Adaptation framework that distillates knowledge from the source language to the target language through a parallel corpus. Also uses unlabeled target documents to adapt the feature extractor. They use the non-English reviews and their machine translated text as a parallel corpus. The sentiment classifier is based on CNN.
11) **SHFR** in Zhou *et al.* [2019]: a Sparse Heterogeneous Feature Representation (SHFR) approach that learns a sparse feature mapping by leveraging the weight vectors of the binary classifiers learned using labeled data in the source and target domains.
12) **Man-Moe** in Chen *et al.* [2019c]: leverage adversarial networks to learn language-invariant features and allows the target language to dynamically and selectively leverage language-specific features through a probabilistic attention-style mixture of experts mechanism. They combine their method with unsupervised cross-lingual word embeddings introduced in Lample *et al.* [2018] and in Chen and Cardie [2018], to perform unsupervised cross-lingual sentiment classification.

5.4.2. Implementation details

We evaluate our approach on the multilingual Amazon dataset in the same fashion as the baseline methods. We use the Amazon dataset's English sentiment review data as the source domain and the Japanese review data as the target domain. We train the teacher BERT model on all 6000 labeled English review data, assign sentiment labels to the parallel corpus using the trained teacher BERT model and finally use the parallel corpus to train the student BERT model to predict the labels of 6000 Japanese review data. To show the effectiveness of our BUCL framework and the new MadMen corpus, we experiment on four different variations of our framework, each utilizing different parallel corpus. Since our MadMen corpus currently only has 1000 pairs of English-Japanese samples and to fairly compare the performances of the corpora, we randomly select same number of samples from the other corpora:

1) **BUCL (JESC)** – uses 1000 pairs of Japanese-English samples from the JESC corpus as the parallel corpus,
2) **BUCL (Kyoto)** – uses 1000 pairs of Japanese-English samples from the Kyoto corpus as the parallel corpus,
3) **BUCL (MadMen)** – uses 1000 pairs of Japanese-English samples from the proposed MadMen corpus as the parallel corpus,
4) **BUCL (Amazon)** – uses 1000 Japanese reviews and its machine translated English text from the multilingual Amazon sentiment dataset as the parallel corpus. The randomly selected 1000 Japanese reviews are not used when evaluating the prediction performance of the BUCL (Amazon) model, i.e. tested on the remaining 5000 reviews.

For the English classification model, we use the "BERT-Base-Uncased"[18] pre-trained model with 12 transformer layers, 768 hidden dimension, 12 attention heads and with total of 110M parameters. For the Japanese sentiment classier, we use the "BERT-Base-Multilingual-Cased" pre-trained model with the same configuration as the English model. For experimentation, we adapt the PyTorch implementation of BERT[19]. The maximum sequence length is 128. Training batch size is 32. Adam [Kingma and Ba, 2014] is used for optimization with an initial learning rate of 1e-5. The dropout probability for all fully connected networks in the embeddings, encoder and pooler is set to 0.1. For each variation of the BUCL, we report the best performance from the first 10 epochs.

5.4.3. Results

Figure 5.2 shows the comparison of performances on the multilingual Amazon dataset with English as source language and Japanese as the target language. The first observation is that the

[18] https://github.com/google-research/bert
[19] https://github.com/huggingface/transformers

CLDFA method has the best performance, closely followed by our proposed BUCL (Amazon). However, unlike the other methods, these two approaches use the task-specific Amazon dataset's Japanese sentiment reviews and their corresponding English machine translations as the parallel corpus. We also have to note CLDFA implies to have used all 6000 Japanese reviews and their translations for parallel corpus, as opposed to our BUCL(Amazon) model that only used 1000 Japanese-English pairs for parallel corpus and the model was evaluated on the unseen remaining 5000 reviews.

We can also observe that the BUCL model utilizing the proposed MadMen corpus outperforms the other BUCL models trained with the same size JESC and Kyoto bilingual corpora by notable margins. Given its simple teacher-student architecture, the BUCL (MadMen) model shows competitive results when compared with other more sophisticated supervised and unsupervised transfer learning approaches. UMM and Bi-PV method have the best accuracy rates among the models trained using non-task specific parallel corpora. However, they have used 500,000 pairs of parallel samples, as opposed to our BUCL (MadMen) that utilized only 1000 pairs of parallel samples.

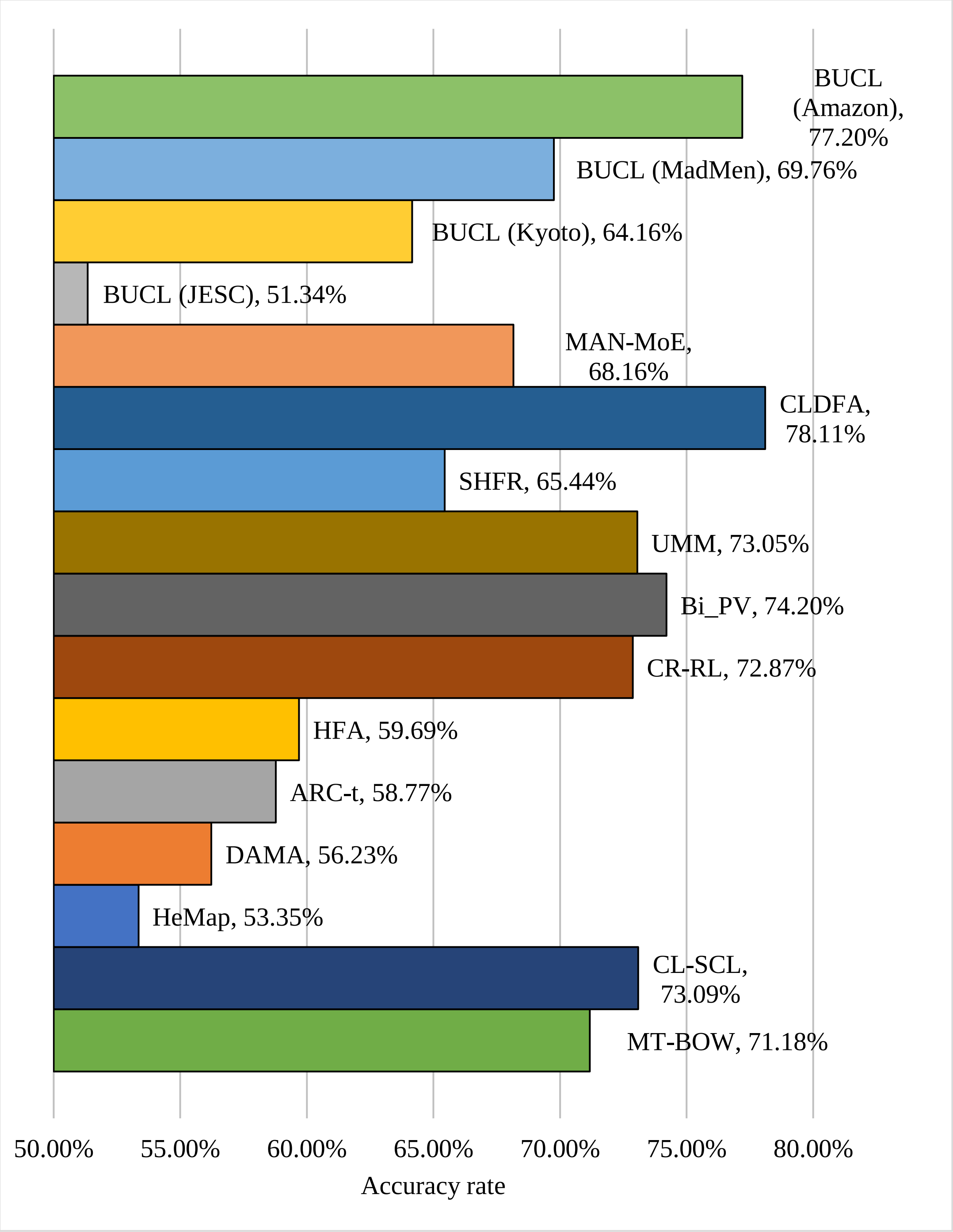


Figure 5.2. Cross-lingual sentiment classification accuracy on Japanese Amazon sentiment dataset.

Another interesting observation is the low performance of JESC corpus. Although it is also a dataset comprising of English-Japanese film subtitles, as our MadMen corpus, we speculate its lackluster prediction accuracy is due to its noisy data containing too many irrelevant or unrealistic expressions. Perhaps some form of categorization could benefit its adaption for various downstream NLP tasks.

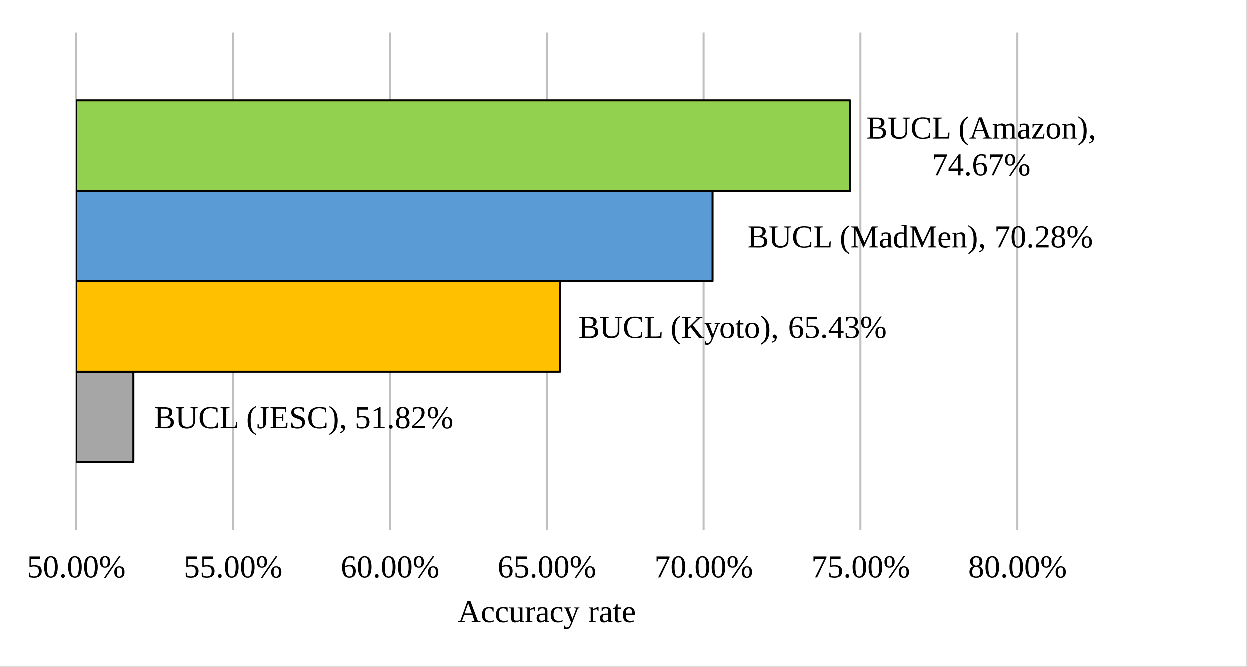


Figure 5.3. Cross-lingual sentiment classification accuracy on Japanese Rakuten sentiment dataset.

To further demonstrate MadMen corpus' potential, we additionally experiment our proposed method and corpus on publicly available Rakuten binary sentiment dataset [Zhang and LeCun, 2017], consisting of user reviews crawled from the Japanese online shopping website rakuten.co.jp. Here we can see that the performances of BUCL trained with non-task specific parallel corpora stay the same and even show slight accuracy rate improvements. Even though the BUCL (Amazon) model still outperforms the other corpora due the general topic similarity between Amazon and Rakuten datasets, its accuracy rate drops notably when facing slightly different target domain. We speculate that if given even more different Japanese sentiment classification tasks, such as twitter or blog post sentiment classification, the general sentiment capabilities of these corpora can be further discovered.

5.5. Conclusion

In this chapter we have proposed an unsupervised transfer cross-lingual English-to-Japanese sentiment classification method utilizing a combination of low-level feature mapping and bidirectional LSTM model. Our approach shows competitive results on Japanese Amazon sentiment dataset and has stable performance when applied on different sentiment dataset from Rakuten. Our proposed task-general MadMen parallel corpus outperforms other Japanese-English corpora such as Kyoto Wikipedia corpus and JESC film subtitle corpus, with the exception of task-specific Amazon dataset.

Chapter 6. Summary and future work

In recent years, there have been great advances both in supervised and unsupervised transfer learning for various NLP and computer vision tasks. However, in the resource-scarce machine learning application areas, works in unsupervised transfer learning have lacking greatly behind the leading fields. Therefore, in our thesis, we emphasize the effective application of multi-level transfer learning via creating low, mid and high level common feature representations to better leverage the labeled data in the source domains.

The main research contributions of this thesis are following:

- New low-level heterogeneous feature mappings based on sensor attributes and daily activity pattern (Chapter 3).
- A new cross-domain ADL recognition method that learns to discern daily activities from coarse-grain feature representations without using any labels in the target domain (Chapter 3).
- Extensive experimentation analysis of BERT and XLNet pre-trained language models in the context of cross-domain sentiment classification (CDSC) (Chapter 4).
- Updates the state-of-the-arts results in CDSC (Chapter 4).
- A new English-Japanese sentiment corpus composed of bilingually aligned subtitles. It outperforms, in the context of cross-lingual sentiment classification (CLSC), other similar-sized English-Japanese parallel corpora, such as JESC and Kyoto Wikipedia (Chapter 5).
- A new English-Japanese cross-lingual sentiment classification framework that learns to determine Japanese user product review’s sentiment without any machine translation or labeled Japanese sentiment text data (Chapter 5).

We obtained promising results from this thesis, but we acknowledge that our work do not fully solve the unsupervised transfer learning problems in ADL recognition and multi-lingual sentiment analysis. There are many potential transfer learning directions for future works:

- Heterogeneous ADL domain adaptation

  With the low-level feature mapping provided, there are many ways to improve cross-domain ADL recognition performance even further using various unsupervised domain adaptation methods and create more complex high-level mapping between sensor events.

Also learning to measure the activity pattern entropy between smart-homes could be a good direction to measure the suitability of transfer learning.

- Multi-lingual sentiment analysis

  To solve the English bias in NLP model, there are few options available. First is really the easiest solution of having more publicly available non-English sentiment datasets. Many multi-lingual NLP tasks have been relying on Wikipedia articles until now. Datasets with more sentiment expressions could play significant role in improving the NLP models' capabilities to capture more nuanced and complex sentiments. Another possible venue is learning shared representations of languages, which is a quite challenging problem by itself.

Bibliography

Akhtar, M.S., Kumar, A., Ekbal, A. and Bhattacharyya, P., 2016, December. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 482-493).

Alemdar, H., Ertan, H., Incel, O.D. and Ersoy, C., 2013, May. ARAS human activity datasets in multiple homes with multiple residents. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare* (pp. 232-235). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Alemdar, H., Tunca, C. and Ersoy, C., 2015. Daily life behaviour monitoring for health assessment using machine learning: bridging the gap between domains. *Personal and Ubiquitous Computing*, *19*(2), pp.303-315.

Alessia, D., Ferri, F., Grifoni, P. and Guzzo, T., 2015. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, *125*(3).

ali Hamad, R., Salguero, A.G., Bouguelia, M.R., Espinilla, M. and Quero, J.M., 2019. Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors. *IEEE journal of biomedical and health informatics*.

Alsheikh, M.A., Selim, A., Niyato, D., Doyle, L., Lin, S. and Tan, H.P., 2016, March. Deep activity recognition models with triaxial accelerometers. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.

Aminikhanghahi, S. and Cook, D.J., 2019. Enhancing activity recognition using CPD-based activity segmentation. *Pervasive and Mobile Computing*, *53*, pp.75-89.

Aqlan, A.A.Q., Manjula, B. and Naik, R.L., 2019. A Study of Sentiment Analysis: Concepts, Techniques, and Challenges. In *Proceedings of International Conference on Computational Intelligence and Data Engineering* (pp. 147-162). Springer, Singapore.

Araújo, M., Pereira, A. and Benevenuto, F., 2020. A comparative study of machine translation for multilingual sentence-level sentiment analysis. *Information Sciences*, *512*, pp.1078-1102.

Ba, J.L., Kiros, J.R. and Hinton, G.E., 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bakar, U.A.B.U.A., Ghayvat, H., Hasanm, S.F. and Mukhopadhyay, S.C., 2016. Activity and anomaly detection in smart home: A survey. In *Next Generation Sensors and Systems* (pp. 191-220). Springer, Cham.

Bataa, E. and Wu, J., 2019. An Investigation of Transfer Learning-Based Sentiment Analysis in Japanese. *arXiv preprint arXiv:1905.09642*.

Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), pp.157-166.

Blitzer, J., Dredze, M. and Pereira, F., Domain adaptation for sentiment classification. In *45th Annv. Meeting of the Assoc. Computational Linguistics (ACL'07)*.

Blitzer, J., McDonald, R. and Pereira, F., 2006, July. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 120-128).

Bravo-Marquez, F., Frank, E. and Pfahringer, B., 2016. Building a Twitter opinion lexicon from automatically-annotated tweets. *Knowledge-Based Systems*, *108*, pp.65-78.

Callan, J., Hoy, M., Yoo, C. and Zhao, L., 2009. Clueweb09 data set.

Castellucci, G., Croce, D. and Basili, R., 2015, June. Acquiring a large scale polarity lexicon through unsupervised distributional methods. In *International Conference on Applications of Natural Language to Information Systems* (pp. 73-86). Springer, Cham.

Chakraborty, B.K., Sarma, D., Bhuyan, M.K. and MacDorman, K.F., 2017. Review of constraints on vision-based gesture recognition for human–computer interaction. *IET Computer Vision*, *12*(1), pp.3-15.

Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S. and Ye, J., 2012. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(4), pp.1-26.

Chen, C., Chen, Z., Jiang, B. and Jin, X., 2019a, July. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 3296-3303).

Chen, L., Hoey, J., Nugent, C.D., Cook, D.J. and Yu, Z., 2012a. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), pp.790-808.

Chen, L., Nugent, C.D. and Wang, H., 2012b. A Knowledge-Driven Approach to Activity Recognition in Smart Homes. *IEEE Transactions on Knowledge and Data Engineering*, *6*(24), pp.961-974.

Chen, M., Xu, Z., Weinberger, K.Q. and Sha, F., 2012, June. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning* (pp. 1627-1634).

Chen, Q., Li, C. and Li, W., 2017, November. Modeling language discrepancy for cross-lingual sentiment analysis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 117-126). ACM.

Chen, X. and Cardie, C., 2018. Unsupervised Multilingual Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 261-270).

Chen, X., Hassan, A., Hassan, H., Wang, W. and Cardie, C., 2019c, July. Multi-Source Cross-Lingual Model Transfer: Learning What to Share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3098-3112).

Chen, Y. and Xue, Y., 2015, October. A deep learning approach to human activity recognition based on single accelerometer. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1488-1492). IEEE.

Chen, Z., Shen, S., Hu, Z., Lu, X., Mei, Q. and Liu, X., 2019b, May. Emoji-powered representation learning for cross-Lingual sentiment classification. In *The World Wide Web Conference* (pp. 251-262). ACM.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Cieliebak, M., Deriu, J.M., Egger, D. and Uzdilli, F., 2017, April. A twitter corpus and benchmark resources for german sentiment analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media* (pp. 45-51).

Collobert, R. and Weston, J., 2008, July. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

Conneau, A., Schwenk, H., Barrault, L. and Lecun, Y., 2017, April. 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* (pp. 1107-1116).

Cook, D.J. and Krishnan, N.C., 2015. *Activity learning: discovering, recognizing, and predicting human behavior from sensor data*. John Wiley & Sons.

Cook, D.J., Crandall, A.S., Thomas, B.L. and Krishnan, N.C., 2012. CASAS: A smart home in a box. *Computer*, *46*(7), pp.62-69.

Da Silva, N.F., Hruschka, E.R. and Hruschka Jr, E.R., 2014. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems, 66*, pp.170-179.

Dai, A.M. and Le, Q.V., 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems* (pp. 3079-3087).

Dai, W., Yang, Q., Xue, G.R. and Yu, Y., 2007, June. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning* (pp. 193-200).

Dai, Z., Yang, Z., Yang, Y., Cohen, W.W., Carbonell, J., Le, Q.V. and Salakhutdinov, R., 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

De, D., Bharti, P., Das, S.K. and Chellappan, S., 2015. Multimodal wearable sensing for fine-grained activity recognition in healthcare. *IEEE Internet Computing*, *19*(5), pp.26-35.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019, June. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171-4186).

Ding, X., Liu, B. and Yu, P.S., 2008, February. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231-240).

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T., 2014, January. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655).

Donaj, G. and Maučec, M.S., 2019. Extension of HMM-Based ADL Recognition With Markov Chains of Activities and Activity Transition Cost. *IEEE Access, 7,* pp.130650-130662.

Duan, L., Xu, D. and Tsang, I.W., 2012, October. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*.

Fahad, L.G. and Rajarajan, M., 2015, December. Anomalies detection in smart-home activities. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 419-422). IEEE.

Ferilli, S. and Esposito, F., 2013. A logic framework for incremental learning of process models. *Fundamenta Informaticae*, *128*(4), pp.413-443.

Feuz, K.D. and Cook, D.J., 2014. Heterogeneous transfer learning for activity recognition using heuristic search techniques. *International Journal of Pervasive Computing and Communications*.

Feuz, K.D. and Cook, D.J., 2017. Collegial activity learning between heterogeneous sensors. *Knowledge and information systems*, *53*(2), pp.337-364.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. and Lempitsky, V., 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, *17*(1), pp.2096-2030.

Gayathri, K.S., Easwarakumar, K.S. and Elias, S., 2017. Probabilistic ontology based activity recognition in smart homes using Markov Logic Network. *Knowledge-Based Systems*, *121*, pp.173-184.

Glorot, X., Bordes, A. and Bengio, Y., 2011, June. Domain adaptation for large-scale sentiment classification: a deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (pp. 513-520).

Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford, 1*(12), p.2009.

Goldberg, Y., 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, *10*(1), pp.1-309.

Gui, L., Lu, Q., Xu, R., Wei, Q. and Cao, Y., 2015, October. Improving transfer learning in cross lingual opinion analysis through negative transfer detection. In *International*

Conference on Knowledge Science, Engineering and Management (pp. 394-406). Springer, Cham.

Ha, S. and Choi, S., 2016, July. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 381-388). IEEE.

Helaoui, R., Riboni, D. and Stuckenschmidt, H., 2013, September. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (pp. 345-354). ACM.

Hendrycks, D. and Gimpel, K., 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, *9*(8), pp.1735-1780.

Howard, J. and Ruder, S., 2018, July. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 328-339).

Hu, D.H., Zheng, V.W. and Yang, Q., 2011. Cross-domain activity recognition via transfer learning. *Pervasive and Mobile Computing*, *7*(3), pp.344-358.

Hu, M. and Liu, B., 2004, August. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177).

Johnson, R. and Zhang, T., 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2015* (p. 103).

Johnson, R. and Zhang, T., 2016, June. Supervised and semi-supervised text categorization using LSTM for region embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48* (pp. 526-534).

Kabir, M.H., Hoque, M.R., Thapa, K. and Yang, S.H., 2016. Two-layer hidden Markov model for human activity recognition in home environments. *International Journal of Distributed Sensor Networks*, *12*(1), p.4560365.

Katz, S., Ford, A.B., Moskowitz, R.W., Jackson, B.A. and Jaffe, M.W., 1963. Studies of illness in the aged: the index of ADL: a standardized measure of biological and psychosocial function. *Jama*, *185*(12), pp.914-919.

Khan, M.A.A.H. and Roy, N., 2018, April. Untran: Recognizing unseen activities with unlabeled data using transfer learning. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)* (pp. 37-47). IEEE.

Khan, M.A.A.H., Roy, N. and Misra, A., 2018, March. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (pp. 1-9). IEEE.

Kim, Y., 2014, October. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746-1751).

Kim, Y., Denton, C., Hoang, L. and Rush, A.M., 2017. Structured attention networks. *arXiv preprint arXiv:1702.00887*.

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

Kiritchenko, S., Zhu, X. and Mohammad, S.M., 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, *50*, pp.723-762.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Kudo, T. and Richardson, J., 2018, November. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 66-71).

Kulis, B., Saenko, K. and Darrell, T., 2011, June. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR 2011* (pp. 1785-1792). IEEE.

Kyoto corpus. Japanese-English Bilingual Corpus of Wikipedia's Kyoto Articles Version 2.01. https://alaginrc.nict.go.jp/WikiCorpus/index_E.html

Lample, G., Conneau, A., Ranzato, M.A., Denoyer, L. and Jégou, H., 2018. Word translation without parallel data. In *International Conference on Learning Representations.*

Lan, Z., Zhu, Y., Hauptmann, A.G. and Newsam, S., 2017. Deep local video feature for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1-7).

Le, Q. and Mikolov, T., 2014, January. Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196).

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature, 521*(7553), pp.436-444.

Lee, S.M., Yoon, S.M. and Cho, H., 2017, February. Human activity recognition from accelerometer data using Convolutional Neural Network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 131-134). IEEE.

Li, Q., 2012. Literature survey: domain adaptation algorithms for natural language processing. *Department of Computer Science The Graduate Center, The City University of New York*, pp.8-10.

Li, S., Wang, Z., Zhou, G. and Lee, S.Y.M., 2011, June. Semi-supervised learning for imbalanced sentiment classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Li, Z., Fan, Y., Jiang, B., Lei, T. and Liu, W., 2019. A survey on sentiment analysis and opinion mining for social multimedia. *Multimedia Tools and Applications, 78*(6), pp.6939-6967.

Li, Z., Wei, Y., Zhang, Y. and Yang, Q., 2018, April. Hierarchical attention transfer network for cross-domain sentiment classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, Z., Zhang, Y., Wei, Y., Wu, Y. and Yang, Q., 2017, August. End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification. In *IJCAI* (pp. 2237-2243).

Lin, J. and Kolcz, A., 2012, May. Large-scale machine learning at twitter. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (pp. 793-804).

Liu, B., 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing, 2*(2010), pp.627-666.

Liu, R., Shi, Y., Ji, C. and Jia, M., 2019. A Survey of Sentiment Analysis Based on Transfer Learning. *IEEE Access*, *7,* pp.85401-85412.

Liu, Y., Nie, L., Liu, L. and Rosenblum, D.S., 2016. From action to activity: sensor-based activity recognition. *Neurocomputing*, *181*, pp.108-115.

Lo, S.L., Cambria, E., Chiong, R. and Cornforth, D., 2017. Multilingual sentiment analysis: from formal to informal and scarce resource languages. *Artificial Intelligence Review, 48*(4), pp.499-527.

Long, M., Wang, J., Ding, G., Sun, J. and Yu, P.S., 2013. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2200-2207).

Manshu, T. and Bing, W., 2019. Adding Prior Knowledge in Hierarchical Attention Neural Network for Cross Domain Sentiment Classification. *IEEE Access*, *7*, pp.32578-32588.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Nasukawa, T. and Yi, J., 2003, October. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70-77). ACM.

Ng, A.Y. and Jordan, M.I., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848).

Ni, Q., Garcia Hernando, A.B., la Cruz, D. and Pau, I., 2015. The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors*, *15*(5), pp.11312-11362.

Niitsuma, H., Kubota, D. and Ohta, M., 2018, September. Japanese sentiment analysis using simple alignment sentence classification. In *Proceedings of the 10th International Conference on Management of Digital EcoSystems* (pp. 126-131). ACM.

Nio, L. and Murakami, K., 2018, March. Japanese Sentiment Classification Using Bidirectional Long Short-Term Memory Recurrent Neural Network. In *Proceedings of the 24th Annual Meeting Association for Natural Language Processing* (pp. 1119-1122).

Nweke, H.F., Teh, Y.W., Al-Garadi, M.A. and Alo, U.R., 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications, 105*, pp.233-261.

Oukrich, N., El Bouazaoui Cherraqi, A.M. and Elghanami, D., 2018. Multi-resident Activity Recognition Method Based in Deep Belief Network. *Journal of Artificial Intelligence*, *11*, pp.71-78.

Pan, S.J. and Yang, Q., 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *10*(22), pp.1345-1359.

Pan, S.J., Ni, X., Sun, J.T., Yang, Q. and Chen, Z., 2010, April. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web* (pp. 751-760). ACM.

Pan, S.J., Tsang, I.W., Kwok, J.T. and Yang, Q., 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, *22*(2), pp.199-210.

Parker, R., Graff, D., Kong, J., Chen, K. and Maeda, K., 2011. English gigaword fifth edition, june. *Linguistic Data Consortium, LDC2011T07*, *12*.

Patel, A. and Shah, J., 2019. Sensor-based activity recognition in the context of ambient assisted living systems: A review. *Journal of Ambient Intelligence and Smart Environments*, *11*(4), pp.301-322.

Peng, H., Ma, Y., Li, Y. and Cambria, E., 2018. Learning multi-grained aspect target sequence for Chinese sentiment analysis. *Knowledge-Based Systems*, *148*, pp.167-176.

Pennington, J., Socher, R. and Manning, C., 2014, October. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT* (pp. 2227-2237).

Pham, H., Luong, T. and Manning, C., 2015, June. Learning distributed representations for multilingual text sequences. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing* (pp. 88-94).

Prettenhofer, P. and Stein, B., 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1118-1127).

Pryzant, R., Chung, Y., Jurafsky, D. and Britz, D., 2017. JESC: japanese-english subtitle corpus. *arXiv preprint arXiv:1710.10639*.

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Rafferty, J., Nugent, C.D., Liu, J. and Chen, L., 2017. From activity recognition to intention recognition for assisted living within smart homes. *IEEE Transactions on Human-Machine Systems*, *47*(3), pp.368-379.

Riboni, D., Sztyler, T., Civitarese, G. and Stuckenschmidt, H., 2016, September. 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* (pp. 1-12). ACM.

Rogers, A., Romanov, A., Rumshisky, A., Volkova, S., Gronas, M. and Gribov, A., 2018, August. Rusentiment: An enriched sentiment analysis dataset for social media in russian. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 755-763).

Ruder, S., 2019. *Neural Transfer Learning for Natural Language Processing* (Doctoral dissertation, National University of Ireland, Galway).

Saeed, A., Ozcelebi, T. and Lukkien, J., 2019. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *3*(2), p.61.

Saif, H., He, Y., Fernandez, M. and Alani, H., 2016. Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, *52*(1), pp.5-19.

Schuster, M. and Paliwal, K.K., 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), pp.2673-2681.

Shalunts, G., Backfried, G. and Commeignes, N., 2016. The impact of machine translation on sentiment analysis. *Data Analytics*, *63*, pp.51-56.

Shang, C., Chang, C.Y., Chen, G., Zhao, S. and Chen, H., 2019. BIA: Behavior Identification Algorithm using Unsupervised Learning Based on Sensor Data for Home Elderly. *IEEE Journal of Biomedical and Health Informatics*.

Sharif Razavian, A., Azizpour, H., Sullivan, J. and Carlsson, S., 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 806-813).

Shi, X., Liu, Q., Fan, W., Philip, S.Y. and Zhu, R., 2010, December. Transfer learning on heterogenous feature spaces via spectral transformation. In *2010 IEEE international conference on data mining* (pp. 1049-1054). IEEE.

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.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A. and Potts, C., 2013, October. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631-1642).

Soulas, J., Lenca, P. and Thépaut, A., 2015. Unsupervised discovery of activities of daily living characterized by their periodicity and variability. *Engineering Applications of Artificial Intelligence, 45*, pp.90-102.

Sukor, A., Syafiq, A., Zakaria, A., Rahim, N.A., Kamarudin, L.M., Setchi, R. and Nishizaki, H., 2019. A hybrid approach of knowledge-driven and data-driven reasoning for activity recognition in smart homes. *Journal of Intelligent & Fuzzy Systems*, (Preprint), pp.1-12.

Sun, B., Feng, J. and Saenko, K., 2016, March. Return of frustratingly easy domain adaptation. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Sztyler, T., Stuckenschmidt, H. and Petrich, W., 2017. Position-aware activity recognition with wearable devices. *Pervasive and mobile computing*, *38*, pp.281-295.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M., 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, *37*(2), pp.267-307.

Tahir, S.F., Fahad, L.G. and Kifayat, K., 2019. Key feature identification for recognition of activities performed by a smart-home resident. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-11.

Tang, D., Qin, B. and Liu, T., 2015, September. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1422-1432).

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D. and Kappas, A., 2010. Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, *61*(12), pp.2544-2558.

Van Kasteren, T., Noulas, A., Englebienne, G. and Kröse, B., 2008, September. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 1-9). ACM.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Wan, J., Li, M., O'Grady, M., Gu, X., Alawlaqi, M.A. and O'Hare, G., 2018. Time-bounded activity recognition for ambient assisted living. *IEEE Transactions on Emerging Topics in Computing*.

Wan, X., 2009, August. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-volume 1* (pp. 235-243). Association for Computational Linguistics.

Wang, C. and Mahadevan, S., 2011, June. Heterogeneous domain adaptation using manifold alignment. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

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

Wemlinger, Z. and Holder, L., 2011, June. The cose ontology: Bringing the semantic web to smart environments. In *International Conference on Smart Homes and Health Telematics* (pp. 205-209). Springer, Berlin, Heidelberg.

Wemlinger, Z.E. and Holder, L.B., 2018. Cross-environment activity recognition using a shared semantic vocabulary. *Pervasive and Mobile Computing*, *51*, pp.150-159.

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. and Klingner, J., 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xia, R., Zong, C., Hu, X. and Cambria, E., 2013. Feature ensemble plus sample selection: domain adaptation for sentiment classification. *IEEE Intelligent Systems*, *28*(3), pp.10-18.

Xiao, M. and Guo, Y., 2013, October. Semi-supervised representation learning for cross-lingual text classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1465-1475).

Xu, K. and Wan, X., 2017, September. Towards a universal sentiment classifier in multiple languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 511-520).

Xu, R. and Yang, Y., 2017, July. Cross-lingual Distillation for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1415-1425).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q.V., 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. and Hovy, E., 2016, June. 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* (pp. 1480-1489).

Yao, S., Hu, S., Zhao, Y., Zhang, A. and Abdelzaher, T., 2017, April. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 351-360). International World Wide Web Conferences Steering Committee.

Yu, J. and Jiang, J., 2016, November. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 236-246).

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T. and Saminger-Platz, S., 2017. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*.

Zhang, H.B., Zhang, Y.X., Zhong, B., Lei, Q., Yang, L., Du, J.X. and Chen, D.S., 2019. A Comprehensive Survey of Vision-Based Human Action Recognition Methods. *Sensors*, *19*(5), p.1005.

Zhang, L., Wang, S. and Liu, B., 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), p.e1253.

Zhang, X. and LeCun, Y., 2017. Which encoding is the best for text classification in chinese, english, japanese and korean?. *arXiv preprint arXiv:1708.02657*.

Zhou, J.T., Tsang, I.W., Pan, S.J. and Tan, M., 2014, April. Heterogeneous domain adaptation for multiple classes. In *Artificial intelligence and statistics* (pp. 1095-1103).

Zhou, J.T., Tsang, I.W., Pan, S.J. and Tan, M., 2019. Multi-class Heterogeneous Domain Adaptation. *Journal of Machine Learning Research*, *20*(57), pp.1-31.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S., 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision* (pp. 19-27).