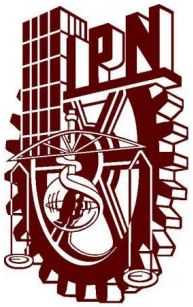


Instituto Politécnico Nacional

Centro de Investigación en Computación

TESIS:

Multimodal Sentiment Analysis in Social Media using Deep Learning with Convolutional Neural Networks

QUE PARA OBTENER EL GRADO DE:

Maestría en Ciencias de la Computación

PRESENTA:

Navonil Majumder

DIRECTOR DE TESIS:

Dr. Alexander Gelbukh

México, CDMX Junio 2017

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 18:30 horas del día 31 del mes de mayo de 2017 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Multimodal Sentiment Analysis in Social Media using Deep Learning with Convolutional Neural Networks"

Presentada por el alumno:

MAJUMDER		**NAVONIL**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	1	5	1	4	1	3

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de Tesis

Dr. Alexander Gelbukh

Dr. Sergio Suárez Guerra

Dr. Grigori Sidorov

Dra. Olga Kolesnikova

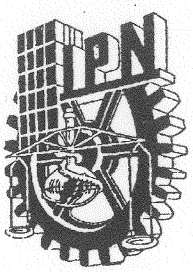
Dra. Sofía Natalia Galicia Haro

Dr. Ildar Batyrshin

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Marco Antonio Ramírez Salinas

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día **21** del mes **Junio** del año **2017**, el que suscribe **Navonil Majumder** alumno del Programa de **Maestría en Ciencias de la Computación** con número de registro **B151413**, adscrito al **Centro de Investigación en Computación**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de **Dr. Alexander Gelbukh** y cede los derechos del trabajo intitulado **"Multimodal Sentiment Analysis in Social Media using Deep Learning with Convolutional Neural Networks"**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: **n.majumder.2009@gmail.com**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Navonil Majumder.

Navonil Majumder

Nombre y firma

Resumen

Gracias al rápido crecimiento de Internet y la proliferación de tabletas y teléfonos inteligentes, las plataformas de medios sociales como Facebook y YouTube se han vuelto de gran relevancia. La gente comparte todo tipo de contenido en dichas plataformas, lo que lleva a una enorme cantidad de datos que requieren procesamiento. Las empresas están tratando de utilizar esta plétora de datos a su favor mediante el desarrollo de sistemas automatizados para varias aplicaciones. Una aplicación de este tipo es la retroalimentación automatizada de los clientes, acumulada en las reseñas de los usuarios, donde el problema fundamental es la minería de los sentimientos de los usuarios asociados con un producto o servicio en particular. Resolver un problema tan complejo de una gran cantidad de datos requiere sistemas de *análisis de sentimientos* eficientes y eficaces.

En esta tesis tratamos específicamente del análisis de sentimientos en videos, donde la información está disponible en tres modalidades: audio, video y texto, de ahí el análisis de sentimiento multimodal. Uno de los mayores desafíos del análisis de sentimiento multimodal es la fusión de las modalidades. Presentamos dos nuevos métodos de fusión de múltiples modalidades: la fusión de forma jerárquica utilizando perceptrones, y el empleo del mecanismo de atención. Con ambos métodos obtuvimos mejoras sobre la fusión basada en concatenación simple.

En la literatura, la mayoría de las publicaciones consideran que los enunciados en un video son independientes entre sí. En realidad, esto no es del todo cierto, ya que los enunciados circundantes pueden proporcionar contexto a un enunciado dado. Modelamos esta influencia entre los enunciados utilizando la LSTM (memoria a corto plazo larga, por sus signos en inglés: *long short-term memory*), que es un tipo especial de red neuronal recurrente. Esto mejora muy considerablemente el rendimiento de clasificación de sentimientos en comparación con el mejor método existente del estado del arte. También agregamos la red de atención en la parte superior de la LSTM, que mejora aún más el rendimiento gracias a la amplificación de la información relevante.

Además presentamos un enfoque basado en la red neuronal profunda para la detección de la personalidad a partir de datos textuales, específicamente ensayos. Utilizamos una combinación de características extraídas automáticamente y características construidas manualmente. Además, demostramos que la eliminación de las oraciones emocionalmente neutrales mejora aún más el rendimiento. Observamos una mejora del desempeño sobre el estado del arte para todos los rasgos de personalidad.

Abstract

Due to the rapid growth of internet and proliferation of smartphones and tablets, social-media platforms like Facebook and YouTube have risen to great relevance. People share all sorts of contents in such platforms, leading to huge amount of data requiring processing. Companies are trying use these plethora of data to their advantage by developing automated systems for several applications. One such application is automated customer feedback gathering from user reviews, where the fundamental problem is to mine user sentiment associated with a particular product or service. Solving such a complex problem from a huge amount of data requires efficient and effective *sentiment analysis* systems.

In this thesis, we specifically deal with the sentiment analysis of videos, where information is available in three modalities: audio, video, and text, hence multimodal sentiment analysis. One of the biggest challenge of multimodal sentiment analysis is modality fusion. We present two novel methods of fusion multiple modalities: fusing in a hierarchical fashion using perceptrons and employing attention mechanism. We obtained improvement over simple concatenation based fusion for both of the methods.

In literature, most of the works consider utterances in a video to be independent of each other. In reality, this is not quite true, since the surrounding utterances can provide context to a given utterance. We model such inter-utterance influence using Long Short-Term Memory (LSTM), which is a special kind of recurrent neural network. This greatly improves sentiment classification performance over state-of-the-art method. We also added attention network on top of LSTM, which further improves the performance due to the amplification of relevant information.

We also present a deep network based approach for personality detection from textual data, specifically essays. We use a combination of automatically extracted features and hand-crafted features. Also, elimination of emotionally-neutral sentences further improves performance. We observe improved performance over the state-of-the-art for all of the personality traits.

Acknowledgements

"The greatest enemy of knowledge is not ignorance;
it is the illusion of knowledge."

Stephen Hawkings

My time as Master's student at *Centro de Investigación en Computación* (CIC) of *Instituto Politécnico Nacional* (IPN) has been one of the joyous periods of my life. I enjoyed the classes given by the respected professors of this institute. Most of all, I built up a strong affinity towards research. I would like to express my gratitude towards the persons and organizations to whom I owe this thesis.

First of all, I would thank my advisor *Dr. Alexander Gelbukh* for his endless support through his expert guidance and motivation. A lot of appreciation for my friend and collaborator *Dr. Soujanya Poria*, without whose encouragement and ideas my masters study would not have been possible. I am also grateful for his guidance in network architectures and his kindness for letting me part of his research group which in turn exposed me to a lot of opportunities.

I greatly appreciate the administration of CIC for their through-out support and patience when I did not speak any Spanish. Also, I am grateful to the academic body of CIC who deemed me to be worthy of this esteemed institution. It goes without saying, the kindness of the professors of this institute who shared their valuable knowledge with us which helped us excel in our research.

Many thanks to *Consejo Nacional de Ciencia y Tecnología* (CONACYT) for their financial support, due to which I could devote myself full-time to research and study. I commend *Sociedad Mexicana de Inteligencia Artificial* (SMIA) for their support in attendance of several reputed conferences. Also, I am thankful to *Red Temática en Tecnologías del Lenguaje* (Red TTL) for their insightful and comprehensive workshops.

A lot of thanks to my friends and fellow students of CIC for their invaluable support. I thank the great country of México for their warm hospitality and strong endorsement of scientific research and development.

Last but not the least, I express my great appreciation to my parents whose upbringing made me who I am today.

Contents

List of Figures

List of Tables

List of Abbreviations

BOW	**B**ag **O**f **W**ords
CBOW	**C**ontinuous **B**ag **O**f **W**ords
CNN	**C**onvolutional **N**eural **N**etworks
DBF	**D**eep **B**elief **N**etwork
GRU	**G**ated **R**ecurrent **U**nit
LIWC	**L**inguistic **I**nquiry and **W**ord **C**ount
LLD	**L**ow **L**evel **D**escriptor
LSTM	**L**ong **S**hort **T**erm **M**emory
MKL	**M**ultiple **K**ernel **L**earning
MLP	**M**ulti **L**ayer **P**erceptron
MSD	**M**ean **S**quared **D**eviation
MSE	**M**ean **S**quared **E**rror
NLP	**N**atural **L**anguage **P**rocessing
ReLU	**Re**ctified **L**inear **U**nit
RNN	**R**ecurrent **N**eural **N**etworks
RNTN	**R**ecursive **N**eural **T**ensor **N**etwork
SGD	**S**tochastic **G**radient **D**escent
SLP	**S**ingle **L**ayer **P**erceptron
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

In this chapter, we first introduce the reader to the problems we strive to address in this thesis. We also discuss necessary background to understand the problem and its relevance in our daily life. Then, we present our solution in an abstract fashion with its novelty. Finally, we discuss the contributions of this work to the scientific community. In the last section of this chapter we discuss the structure of the subsequent chapters.

1.1 Background

Since the beginning of this millennium, Internet has become backbone of our society in several aspects. Modern society heavily relies on Internet for many fundamental tasks, such as communication, finance, education, entertainment etc. Due to the rise of e-commerce websites like Amazon, people are able purchase necessary items by the push of a button, without even leaving their home. Even groceries are supplied to the residences in some small towns.

In recent years, social media platforms like Facebook, YouTube have risen to high relevance due to proliferation of smartphones and tablets. Nowadays, people tend to share their views on food, movies, music, politics, newly released products in such social media platforms either as posts or in form of videos.

This huge amount of openly available data has many critical uses if can be processed properly. For example, recommender systems can suggest users products they might be interested in based on their previous purchases and their thoughts on those purchases. Since, opinion of one person can influence others in social media, effective marketing strategy can be devised based on connectivity among people in social media and their influence among each other.

Making use of such a huge amount of data calls for large infrastructure and proper research and development. Large enterprises are investing a large capital in social-media mining and receiving even larger return in terms of revenue. This makes social-media mining a very attractive and active field of research.

In this thesis we deal with select few problems of social-media mining. In the next subsection (Section 1.2) we discuss those problems.

We primarily propose deep neural network based solutions due to their successful application in various machine learning related tasks, such as image classification, machine

translation. In Chapter 2, we discuss various neural networks on which our solutions rely.

1.2 Problem

In this thesis, we primarily deal with Multimodal Sentiment Analysis. Also, in the later part we deal with Personality Detection from text, which is quite similar to sentiment analysis in essence.

1.2.1 Multimodal Sentiment Classification

1.2.1.1 Sentiment Analysis

In sentiment analysis, the task is to assign appropriate sentiment polarity to a given text, which could be a sentence, or utterance, or a document. Usually, three sentiment polarities are considered: positive, negative, and neutral. Table 1.1 shows examples of different sentiments in text. However, in the presented work we only considered positive

TABLE 1.1: Different Sentiment Polarities

Polarity	Attribute(s)	Example
Positive	Happy, Satisfied	*The display is gorgeous.*
Negative	Unhappy, Angry	*The phone is too bulky.*
Neutral	Non-emotional	*It is powered by an Intel Core i5 processor.*

and negative polarities. Also, we perform sentiment classification in utterance (spoken sentence) level.

1.2.1.2 Sentiment Analysis with Multiple Modalities

Videos usually consist of three different information channels, which are audio, video, and text (in form of speech). These channels are also called modalities. Hence, assigning sentiment polarity to such videos is called Multimodal Sentiment Analysis.

Text modality usually contains a lot of sentimental information. However, visual modality may provide additional information, not captured in textual modality. Such information could be facial expression, subtle muscle movements (such as frown, grin etc.). Audio modality provides relevant information to the sentiment classification in form of pitch, change in tone etc.

In Chapters 3,4,5, we present different ways to perform multimodal sentiment classification.

1.2.2 Personality Detection from Text

Personality detection is very similar in essence to sentiment analysis. Here, the job is to detect personality traits *extroversion, neuroticism, agreeableness, conscientiousness,*

and openness from a given text. In Chapter 6 we present a method for automatic personality detection form text, specifically from essay. Although, we planned to perform this task on multimodal data, due to unavailability of multimodal dataset for personality detection we chose to perform this task on textual data only.

1.3 Relevance

In this section, we discuss the relevance of the problem we tackle in thesis in our society and daily life.

1.3.1 Multimodal Sentiment Analysis

With the increasing availability of smartphones and tablets people tend to share their opinions on various topics on social-media platforms in form of video reviews, particularly on the newly released products. They share what works about the product, which features are gimmick, which features need polish etc. Big product-based companies are very much interested in user reception, as it helps them decide their future course regarding their products. For example, if a company launches a new smartphone in the market which has a general negative consensus about its battery life, the company will want to improve its battery life in the future iterations.

Recommender systems can take advantage of the users' experience with their past purchases to make more accurate and relevant recommendations.

Sentiment analysis can be very helpful in risk management, where the companies can form their risk management strategy based on the user reception of their products.

There can be numerous applications of Multimodal Sentiment Analysis beyond the ones mentioned above, only bound by our imagination.

1.3.2 Personality Detection

Personality detection has several practical applications. Recommender systems can make recommendations of products based on users' personality. Also, it can used used in mental health diagnosis. In forensics, it can be helpful in psychological profiling. The companies can hire employees based on their personalities, rejecting unsuitable candidates.

1.4 Novelty

1.4.1 Multimodal Sentiment Analysis

One of the main challenges in multimodal sentiment analysis is the fusion of multiple modalities. The traditional method for modality fusion has been concatenation of the feature vectors from different modalities. However, it can lead to redundancy of features

in multimodal feature vectors, which can lead to reduced performance of the classification process. In Chapter 3, we discuss a method which merges different modalities in a hierarchical fashion , leading to filtration of noisy features and improved performance over concatenation method. Also, in Chapter 5 we employ attention mechanism for feature fusion and modeling inter-utterance dependency which leads to better performance against concatenation method.

Majority of the recent works considers utterances in a video independent of each other, which is not accurate, since meaning of one utterance can influence the meaning of other utterances. It is specially valid in case of sarcasm. Our approach models inter-utterance dependencies, which enables better sentiment classification as demonstrated in Chapters 4, 5.

1.4.2 Personality Detection from Text

For personality detection from essay type text, we employed a distinctive network architecture, where it finds sentence representations from word embeddings and later merges them to document level representation. Additionally, we filtered out emotionally-neutral sentences from the text based on an emotion-lexicon, which led to improved performance.

1.5 Contributions

The contributions of this thesis are as follows:

- **Improved Feature Fusion.** Development of two different more effective feature fusion methods opposed to regular concatenation based fusion methods;
- **Inter-Utterance Dependency.** Modeling of inter-utterance dependency in the network architecture, leading to better sentiment classification;
- **Hierarchical Document Modeling.** Improved automatic personality detection from text using hierarchical document modeling technique and elimination of emotionally-neutral sentences from documents.

1.6 Structure of This Document

This chapter serves as an introduction to the problems we address in this thesis. Chapter 2 provides the reader with theoretical background necessary to understand the content of this thesis. Chapter 3 discusses a novel multimodal fusion method. Chapter 4 explains the method to model inter-utterance dependency, which yields improved performance in sentiment analysis. Chapter 5 improves upon the model discussed in Chapter 4 by the use of attention mechanism. Chapter 6 deals with personality detection from text, where it uses deep networks to achieve state of the art results. Finally, Chapter 7 concludes

this thesis by discussing contribution in more detailed fashion, along with mentioning the publications and planned future work.

Chapter 2

Theoretical Framework

In this chapter, we briefly explain different methods, tools, measures which are widely used in the work described. We describe Multi-Layer Perceptron (MLP), Support Vector Machine (SVM) as classifiers. Next, several types of neural networks, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), are described along with word2vec vectors [1], followed by different evaluation measures.

2.1 Classification Techniques

2.1.1 Perceptron

In simplest form of classification, data-points of two different labels are to be separated using a hyperplane. Single-Layer Perceptron (SLP) performs exactly that task.

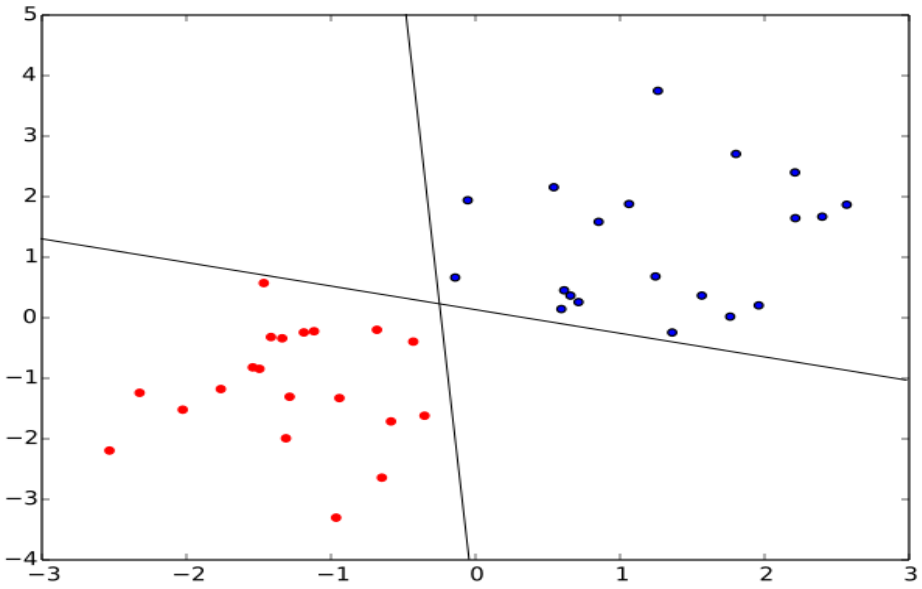


FIGURE 2.1: Linearly separable data classified by a hyperplane

Figure 2.1 shows two possible straight lines (hyperplane in 2D) separating two classes of data-points. We generalize the equation of hyperplane for n dimensions, as

$$\mathbf{w}^\intercal.\mathbf{x} + b = 0, \tag{2.1}$$

where $\mathbf{w}$ is a vector with n dimensions and b is a scalar. We tune the parameter $\mathbf{w}$ and b of Equation (2.1) to obtain a hyperplane that linearly separates data-points $\mathbf{x}$, i.e. the decision-function can be defined as

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^\intercal.\mathbf{x} + b > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

To tune the aforementioned parameters, we use the following steps:

1. Initialize the weights $\mathbf{w}$ and b with small random value close to zero.
2. For each sample x_j in the training set perform

$$w(t+1) = w(t) + (y_j - \hat{y_j}(t))x_j \tag{2.3}$$
$$b(t+1) = b(t) + (y_j - \hat{y_j}(t))x_j \tag{2.4}$$

3. We repeat step 2 as long as the error value $\frac{1}{s}\sum_{j=1}^{s}|y_j - \hat{y_j}|$, where $s =$ number of samples, is greater than some predefined threshold.

2.1.2 Logistic Regression / Single-Layer Perceptron

The issue with this simple perceptron is that it cannot separate data-points which are not linearly separable, as depicted in Figure 2.2. To introduce non-linearity we plug a sigmoid function (σ) to the output of $\mathbf{w}^\intercal.\mathbf{x} + b$, where

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{2.5}$$

Figure 2.3 depicts the shape of Sigmoid function.

So, we redefine the decision-function as

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \sigma(\mathbf{w}^\intercal\mathbf{x} + b) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Since, sigmoid always returns values in $(0, 1)$, the output can be assumed as probability. That is, $\sigma(\mathbf{w}^\intercal\mathbf{x}+b)$ is the probability of sample $\mathbf{x}$ belonging to class 1 and $1-\sigma(\mathbf{w}^\intercal\mathbf{x}+b)$ for class 0. We need to maximize the probability of each sample belonging to their respective expected class. Following this, we can define the total probability for all the samples as

$$P = \prod_{j=1}^{s}[y_j\mathcal{P}(\mathbf{x_j}) + (1 - y_j)(1 - \mathcal{P}(\mathbf{x_j}))], \tag{2.7}$$

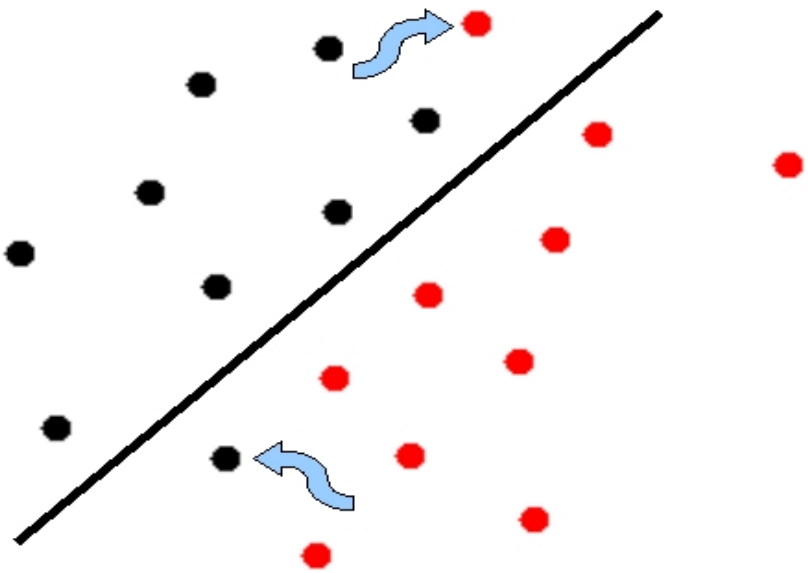


FIGURE 2.2: Linearly non-separable data

where $\mathcal{P}(\mathbf{x}) = \sigma(\mathbf{w}^\intercal \mathbf{x} + b)$ and y_j is the expected label of sample j. Our objective is to maximize the value of P. To achieve so, we use Stochastic Gradient Descent (SGD). However, we need to transform this maximization problem to minimization problem.

P contains $O(s)$ number of products, which is computationally expensive. So, we apply log function on P as follows:

$$\log P = \sum_{j=1}^{s} \log \left[y_j \mathcal{P}(\mathbf{x_j}) + (1 - y_j)(1 - \mathcal{P}(\mathbf{x_j})) \right] \tag{2.8}$$

Since, for a sample j exactly one of y_j and $1 - y_j$ will be zero and the other one, the following can be said:

$$\log P = \sum_{j=1}^{s} [y_j \log \mathcal{P}(\mathbf{x_j}) + (1 - y_j) \log (1 - \mathcal{P}(\mathbf{x_j}))] \tag{2.9}$$

Now, we negate $\log P$ to convert the problem to minimization problem. Also, we normalize the value by dividing the quantity with s. Finally, we get our objective function

$$J = -\frac{1}{s} \sum_{j=1}^{s} [y_j \log \mathcal{P}(\mathbf{x_j}) + (1 - y_j) \log (1 - \mathcal{P}(\mathbf{x_j}))] \tag{2.10}$$

Function J is called log-loss or binary cross-entropy. We now apply SGD or one of its variants to minimize J.

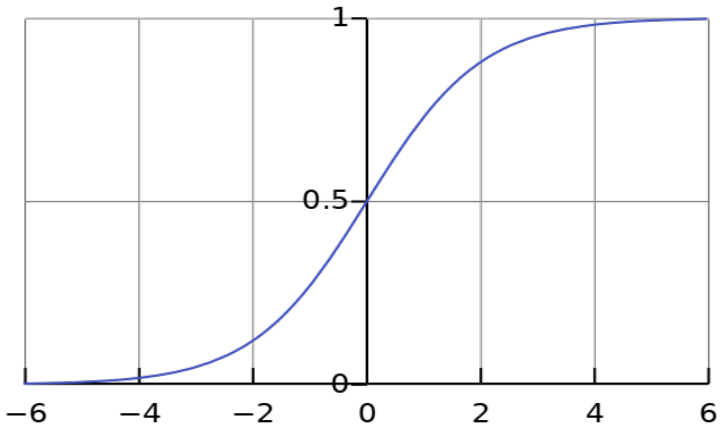


FIGURE 2.3: Sigmoid function (σ)

2.1.3 Multi-Layer Perceptron

Logistic regression is capable of separating almost linearly separable data-points. However, there are certain arrangements of data-points that cannot be classified even with logistic regression. For example, Figure 2.4 shows four points with two classes (XOR function), which cannot separated using a single straight-line (even with logistic-curve). Clearly, it requires two straight-lines to make the classification.

To solve this, we employ multiple perceptrons. We plug two perceptrons (Hidden Layer) with the input layer. Each perceptron stands for a straight-line. Then we feed the output of those two perceptrons to another perceptron, which is the output output layer. This network is depicted in Figure 2.4.

The design of XOR network is quite trivial, since we are aware of the arrangement of data-points. However, in practice it is very hard to gauge the distribution of data-points over vector space, often impossible because of the dimensionality of input. So, usually the networks are built with higher number of perceptrons (or nodes in network terminology) and tuned based on the performance of the network. We perform the tuning of parameters using SGD or one of its variants, as discussed in the next subsection 2.4.1.

2.1.4 Support Vector Machine (SVM)

2.1.4.1 Linearly Separable, Binary Classification

The simplest form of classification is, the classification of linearly separable data-points with two labels. The diagram below, shows an example of such a dataset.
The goal is to find a hyperplane (also known as *decision boundary*) of the form $\mathbf{w}^\intercal\mathbf{x}+b = 0$, which separates data-points of two different classes (blue and red).

We can label any point above the decision boundary with 1 and points below with -1. The advantage of this labeling scheme is, the decision function can be concisely written as $f(x) = sign(\mathbf{w}^\intercal\mathbf{x}+b)$, since $\mathbf{w}^\intercal\mathbf{x}+b > 0$ for all points above the hyperplane and $\mathbf{w}^\intercal\mathbf{x} + b < 0$ for all points below.

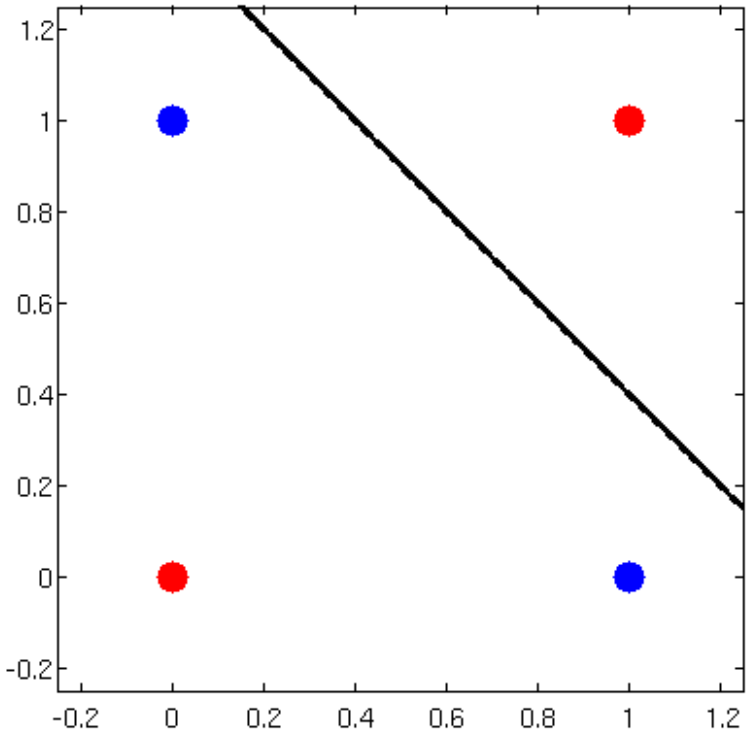


FIGURE 2.4: XOR function: Linearly non-separable

Now, you might notice that, there is a gap between the decision boundary and the nearest point(s) (also known as *support vectors*) of either of the classes. We need to choose the hyperplane in such a way that, the gap is maximized. Such a hyperplane is called *Maximal Marginal Hyperplane* (MMH). The idea is to keep the classes as far as possible from the decision boundary to minimize classification error.

Let, $\mathbf{w}^\intercal\mathbf{x} + b = 1$ be a hyperplane, such that all points on and above it belong to class +1. Similarly, $\mathbf{w}^\intercal\mathbf{x} + b = -1$ be a hyperplane, such that all points on and below it belong to class -1. So, we can check if a data-point $\mathbf{x_i}$, labeled y_i(= +1 or -1), has been correctly classified by verifying if $y_i(\mathbf{w}^\intercal\mathbf{x_i} + b) \geq 1$.

Hyperplanes $\mathbf{w}^\intercal\mathbf{x} + b = 1$ and $\mathbf{w}^\intercal\mathbf{x} + b = -1$ are parallel, since they have the same normal vector given by $\mathbf{w}$. So, we need to maximize the distance between the given hyperplanes.

The distance between the hyperplanes can be given by $\frac{|(b-1)-(b+1)|}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$. Hence, the objective is to maximize $\frac{2}{||\mathbf{w}||}$. It is equivalent to minimizing $\frac{||\mathbf{w}||}{2} = \frac{\sqrt{\mathbf{w}^\intercal\mathbf{w}}}{2}$. Again, it is equivalent to minimizing $\frac{\mathbf{w}^\intercal\mathbf{w}}{2}$.

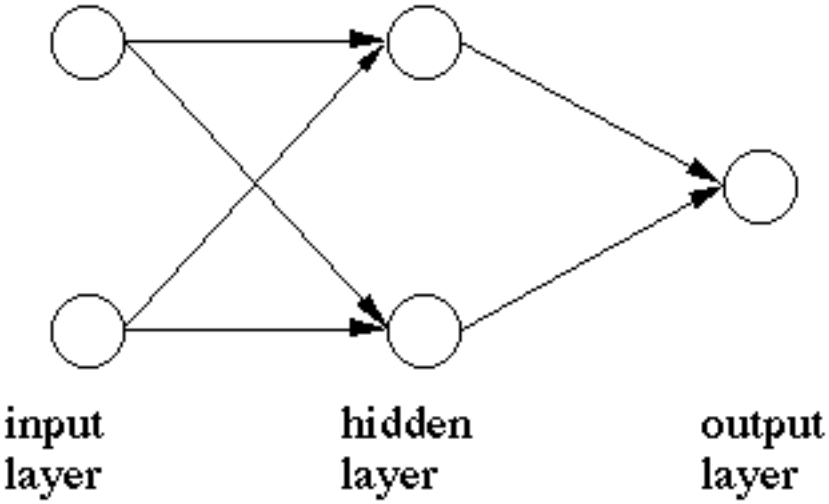


FIGURE 2.5: Multi-Layer Perceptron for XOR function

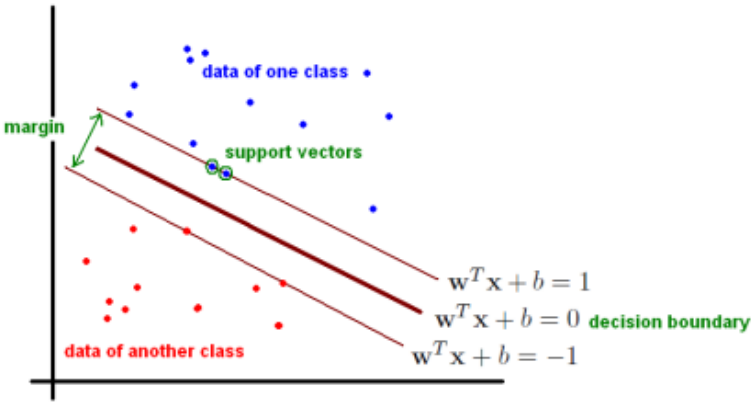


FIGURE 2.6: Example of linearly separable data

Finally, the optimization problem can be summarized as:

$min_{\mathbf{w},b}\dfrac{\mathbf{w}^\intercal\mathbf{w}}{2}$

subject to $y_i(\mathbf{w}^\intercal\mathbf{x_i}+b)\geq 1$, for all $i=1,2\ldots,m$.

2.1.4.2 Soft-Margin extension

Now, consider that the data-points are not completely linearly separable. In such case, we need to allow some data-points to lie on the other side of the decision boundary.

We introduce *slack variables* $\zeta_i \geq 0$ for all x_i and rewrite the problem as follows:

$$min_{\mathbf{w},b,\zeta} \frac{\mathbf{w}^\intercal \mathbf{w}}{2} + C \sum_{i=1}^{m} \zeta_i$$

subject to $y_i(\mathbf{w}^\intercal \mathbf{x_i} + b) \geq 1 - \zeta_i,\ \zeta_i \geq 0$ for all $i = 1, 2 \ldots, m$.

Slack variables allow the quantity $y_i(\mathbf{w}^\intercal \mathbf{x_i} + b)$ to be less than one; meaning $\mathbf{x_i}$ lies on the other side of the decision boundary. However, we penalize such occurrences by adding slack variables to the objective function.

2.1.4.3 Non-Linear Decision Boundary

The data-points $\mathbf{x_i}$ may not be linearly separable in the original space. However, if we map the data-points to a higher dimensional space, it might be linearly separable in that new space.

We use a mapping function ϕ to map the data-points. Hence, we rewrite the optimization problem as follows:

$$min_{\mathbf{w},b,\zeta} \frac{\mathbf{w}^\intercal \mathbf{w}}{2} + C \sum_{i=1}^{m} \zeta_i$$

subject to $y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b) \geq 1 - \zeta_i,\ \zeta_i \geq 0$ for all $i = 1, 2 \ldots, m$.

Example of a mapping function:

$$\phi : \mathbb{R}^2 \mapsto \mathbb{R}^3$$
$$(x_1, x_2) \rightarrow (z_1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$$

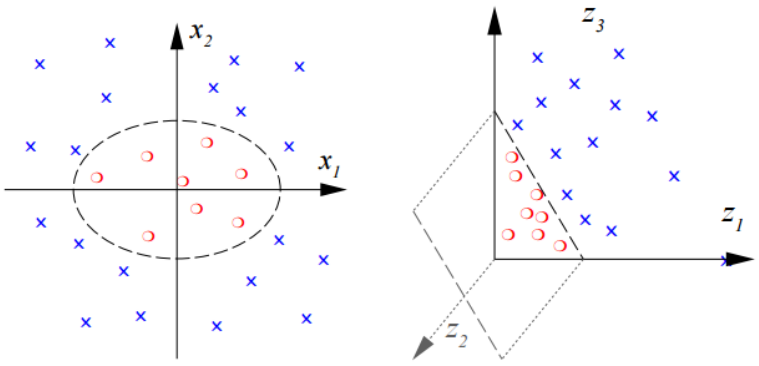


FIGURE 2.7: Polynomial Mapping

2.1.4.4 Formulation as a Lagrangian Optimization

We can use *Lagrangian Multiplier* to incorporate the constraints.

Now, $\zeta_i \geq 1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)$. So, in order for ζ_i to be near zero, $1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)$ has to be as close to zero as possible.

We can capture this condition by adding the following to the objective function: $max_{\alpha_i \geq 0} \alpha_i [1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)]$. If $y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b \geq 1$, then α_i becomes zero; since $1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)$ becomes negative . Otherwise, $\alpha_i \to \infty$. In this way, we are penalizing misclassified points, since $max_{\alpha_i \geq 0} \alpha_i [1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)]$ becomes positive and huge .

So, we rewrite the objective function as:

$$min_{\mathbf{w},b}[\frac{\mathbf{w}^\intercal \mathbf{w}}{2} + \sum_{i=1}^{m} max_{\alpha_i \geq 0} \alpha_i [1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)]].$$

To allow soft-margin, we limit the value of α_i to lie within $[0, C]$.

Now, we can rewrite the formulation as below:

$$min_{\mathbf{w},b}[\frac{\mathbf{w}^\intercal \mathbf{w}}{2} + \sum_{i=1}^{m} max_{\alpha_i \geq 0} \alpha_i [1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)]]$$

$$= min_{\mathbf{w},b}[max_{\alpha \geq 0}[\frac{\mathbf{w}^\intercal \mathbf{w}}{2} + \sum_{i=1}^{m} \alpha_i [1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)]]]$$

$$= min_{\mathbf{w},b}[max_{\alpha \geq 0} J(\mathbf{w}, b, \alpha)], \text{ where } J(\mathbf{w}, b, \alpha) = \frac{\mathbf{w}^\intercal \mathbf{w}}{2} + \sum_{i=1}^{m} \alpha_i [1 - y_i(\mathbf{w}^\intercal \phi(\mathbf{x_i}) + b)]$$

$$= max_{\alpha \geq 0}[min_{\mathbf{w},b} J(\mathbf{w}, b, \alpha)]$$

So, now we minimize J with respect to $\mathbf{w}, b$ and then maximize the result with respect to α.

Now, we set $\frac{\partial J}{\partial \mathbf{w}} = 0$, and find $\mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x_i})$. Again, setting $\frac{\partial J}{\partial b} = 0$, we get the relation $\sum_i \alpha_i y_i = 0$.

After, substitution of the above mentioned values and relation, we get:

$$min_{\mathbf{w},b} J(\mathbf{w}, b, \alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x_i})^\intercal \phi(\mathbf{x_j}).$$

Thus, finally, the dual problem is:

$$max_{\alpha \geq 0}[\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x_i})^\intercal \phi(\mathbf{x_j})]$$

Subject to $\sum_i \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C$

2.1.4.5 Kernel Trick

Since, we are mapping the data-points to a very high-dimensional space, calculating the term $\phi(\mathbf{x_i})^\intercal\phi(\mathbf{x_j})$ may become intractable. However, we can avoid the dot-product by introducing special *kernel functions*, which evaluates the dot-product by operating on the original lower-dimension.

For example, let $\phi : \mathbb{R}^3 \mapsto \mathbb{R}^{10}$, where:

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_2x_3, \sqrt{2}x_1x_3)$$

Now, we can use kernel function $K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i})^\intercal\phi(\mathbf{x_j}) = (1 + \mathbf{x_i}^\intercal\mathbf{x_j})^2$.

So, we can rewrite the dual problem as:

$$max_{\alpha\geq 0}[\sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j})]$$

Subject to $\sum_i \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C$

2.1.4.6 Decision Function

We learn the parameters α and b using Lagrangian Optimization. Then, we classify a test instance $\mathbf{x}$, by calculating $f(\mathbf{x}) = sign(\mathbf{w}^\intercal\phi(\mathbf{x}) + b)$. We substitute $\mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x_i})$ and get $f(x) = sign(\sum_i \alpha_i y_i \phi(\mathbf{x_i})^\intercal\phi(\mathbf{x}) + b) = sign(\sum_i \alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) + b)$.

2.1.5 Other Classification Techniques

There are other classification techniques which are out of the scope of this thesis. For example, Naive Bayes, Random Forest, Decision Tree etc.

2.2 Text Representations

2.2.1 Word2vec Embeddings

Mikolov et al. (2013) [1] revolutionized deep-learning in NLP with Word2vec word embeddings. Word2vec maps words to vectors in such a way that the words retains relative meaning and relationship among themselves in the vectors space. Word2vec has two models, namely Continuous-Bag-of-Words (CBOW) and SkipGram. CBOW model predicts the missing word, given its surrounding k words. This collection of k words is called context. On the other hand, SkipGram model does the opposite thing, it predicts the surrounding words given a single word. In the following subsections, we discuss the problem with regular one-hot representation (Bag of Words) of words and aforementioned two models.

2.2.1.1 Problem with Bag of Words (BOW)

The major problem with Bag of Words is dimensionality. The words are represented by vectors of the size of the vocabulary, which is huge. Moreover, it does not capture the relationship among the words.

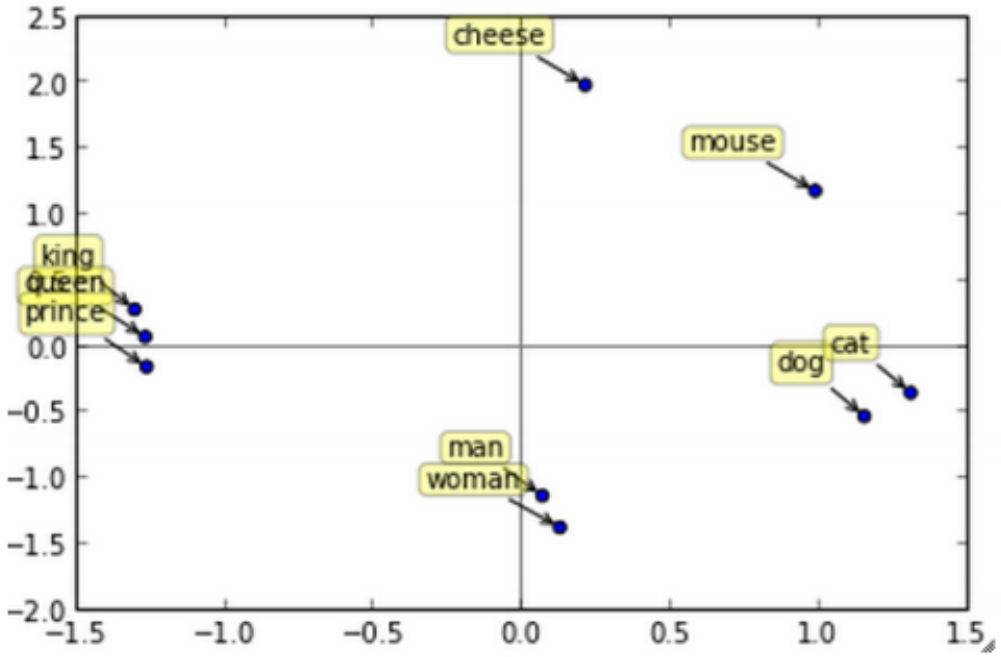


FIGURE 2.8: Words in 2D projection of word2vec vector space: semantically similar words are closer

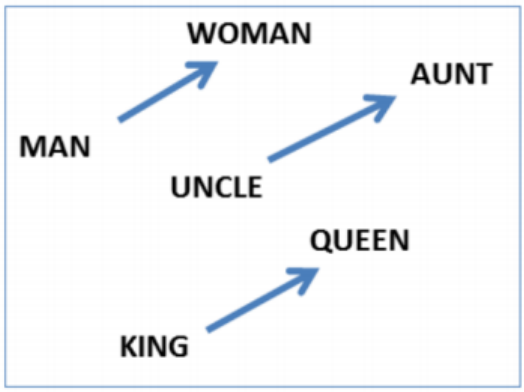
Word2vec vectors are trained to have much smaller dimensionality (Google word2vec vectors are of size 300) and similar words are closer in vector space. Figure 2.8 takes 2D projection of word2vec vector space. In this figure it is visible that similar words are clustered together.

Also, Figure 2.9 shows that depicts that pairs of words having similar relationship between have same distance in vector space.

2.2.1.2 Continuous Bag of Words (CBOW)

Figure 2.10 shows the basic CBOW, where the input layer accepts the one-hot representation of context words $(w(t-2), w(t-1), w(t+1), w(t+2))$.

Let us assume a single hidden layer neural network with $\mathbf{V}$ neurons in the input and output layer and $\mathbf{N}$ number of neurons in the hidden layer. The layers are fully connected and input to the network is the context word represented as a one-hot vector. The layers are connected by weight matrix $\mathbf{W_{V\times N}}$ and $\mathbf{W'_{N\times V}}$ respectively. Here, each word is finally represented as two learnt vectors $\mathbf{V_c}$ and $\mathbf{V_w}$, representing their roles as context



Source: *Linguistic Regularities in Continuous Space Word Representations*, Mikolov et al, 2013

FIGURE 2.9: Vector differences between words

and target words respectively. Given an input one-hot vector $\mathbf{x}$, the hidden layer can be calculated as follows

$$\mathbf{h} = \mathbf{W^T x} = \mathbf{W_{(k,.)}}$$

Thus this $\mathbf{k^{th}}$ row of $\mathbf{W}$ represents the output vector $\mathbf{V_c}$ for the input vector $\mathbf{x}$. Also for each output unit let,

$$\mathbf{y_i} = softmax(\mathbf{u_i})$$

$$where, \mathbf{u} = \mathbf{W'^T h}$$

Thus,

$$\mathbf{u_i} = \mathbf{W'_{(.,i)} h}$$

The $\mathbf{k^{th}}$ column of matrix $\mathbf{W'}$ therefore represents the output vector $\mathbf{V_w}$.

Overall,

$$\mathbf{P}(\frac{\mathbf{w_i}}{\mathbf{c}}) = \mathbf{y_i} = \frac{e^{u_i}}{\sum_{i=1}^{V} e^{u_i}}$$

$$where, \mathbf{u_i} = \mathbf{V_{w_i}^T V_c}$$

The parameters $\theta = \{\mathbf{V_w}, \mathbf{V_c}\}$ are learnt by defining the objective function as the log-likelihood,

$$\mathbf{l}(\theta) = \sum_{\mathbf{w \in Vocabulary}} \mathbf{log}(\mathbf{P}(\frac{\mathbf{w}}{\mathbf{c}}))$$

$$= \sum_{\mathbf{w \in Vocabulary}} \mathbf{log}(\frac{\mathbf{e^{V_w^T V_c}}}{\sum_{\mathbf{i=1}}^{\mathbf{V}} \mathbf{e^{V_{w_i}^T V_c}}})$$

Finding the gradient,

$$\frac{\partial \mathbf{l}(\theta)}{\partial \mathbf{V_w}} = \mathbf{V_c} - \frac{\mathbf{e^{V_w^T V_c} V_c}}{\sum_{\mathbf{w' \in Vocab}} \mathbf{e w'^T V_c}}$$

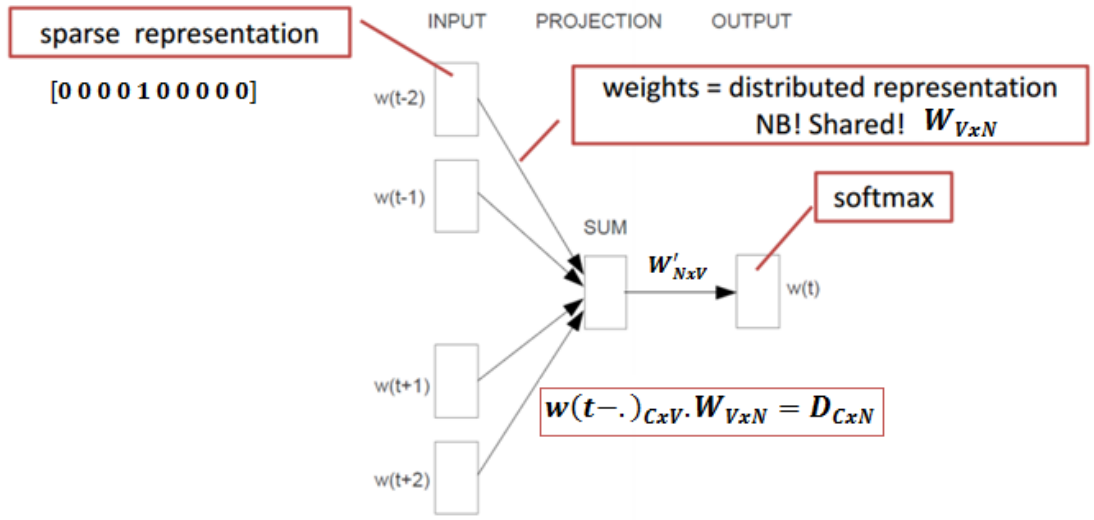


FIGURE 2.10: CBOW: Continuous bag of words model

$$\equiv \frac{\partial \mathbf{l}(\theta)}{\partial \mathbf{V_w}} = \mathbf{V_c}(\mathbf{1} - \mathbf{P}(\frac{\mathbf{w}}{\mathbf{c}})$$

After this, updating of weights is done using Gradient Descent:

$$\mathbf{V_w^{new}} = \mathbf{V_w^{old}} - \eta \frac{\partial \mathbf{l}(\theta)}{\partial \mathbf{V_w}}$$

Similarly doing for $\mathbf{V_c}$.

In the General Model, all the one-hot vectors of context words are taken as input simultaneously, i.e,

$$\mathbf{h} = \mathbf{W^T}(\mathbf{x_1} + \mathbf{x_2} + ... + \mathbf{x_c}).$$

Negative Sampling It is observable here that in the formula mentioned above, the normalization factor iterates over the whole vocabulary for each update. This is therefore intractable when the vocabulary is very large. To overcome this, a different approach is taken where the task considers pairs of target and context words and breaks down to classify where this pair belongs to the corpus or not (wrt. window size restrictions).

Classification Problem Let, $\mathbf{D} \rightarrow$ Set of all (w,c) pair of words in the corpus. and, $\mathbf{D}^{'} \rightarrow$ Set of all (w,c) pair of words **not** in the corpus.
$\mathbf{P}(\mathbf{z} == \mathbf{1}|(\mathbf{w}, \mathbf{c}))$ is the probability that a pair is in $\mathbf{D}$

Therefore,

$$\mathbf{P}(\mathbf{z} == \mathbf{1}|(\mathbf{w}, \mathbf{c})) = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{e}^{-\mathbf{V_c^T V_w}}}$$

$$\mathbf{P(z == 0|(w, c)) = \frac{e^{-V_c^T V_w}}{1 + e^{-V_c^T V_w}}}$$

now,

$$\mathbf{P(z|(w, c)) = \Big(\frac{1}{1 + e^{-V_c^T V_w}}\Big)^z \Big(\frac{e^{-V_c^T V_w}}{1 + e^{-V_c^T V_w}}\Big)^{1-z}}$$

Hence the Likelihood can be defined as,

$$\mathbf{L}(\theta) = \prod_{(w,c) \in D \cup D'} P(z|(w, c))$$

Defining the log-likelihood as the objective function to minimize:

$$\mathbf{l}(\theta) = \sum_{\mathbf{(w,c) \in D \cup D'}} \mathbf{z log}(\frac{1}{1 + e^{-V_c^T V_w}})$$

$$\mathbf{+(1 - z) log}(\frac{1}{1 + e^{V_c^T V_w}})$$

$$\mathbf{l}(\theta) = \sum_{\mathbf{(w,c) \in D}} \mathbf{log}(\frac{1}{1 + e^{-V_c^T V_w}}) + \sum_{\mathbf{(w,c) \in D'}} \mathbf{log}(\frac{1}{1 + e^{V_c^T V_w}})$$

From this we can find the respective gradients and perform the required updates using Gradient Descent. The computational complexity in the above model is no more intractable.

Although **Word2Vec** and similar counterparts like **GloVe** [2] are very efficient in capturing semantic details of words in their vector space representations, they are not very efficient in the task of sentimental analysis. This is mainly due to the fact that these embeddings sometimes cluster semantically similar words which have opposing sentiment polarity. Thus the downstream model used for the sentiment analysis task would not be able to identify this contrasting polarities leading to poor performance. Many solutions to this problem have been proposed based on the modification of the original C&W model by Collobert et al. [3]. This is an open field which high scope of improvement in the said task.

2.2.1.3 SkipGram

Unlike CBOW, in SkipGram model the network predicts the context from given word, i.e. predict $w(t-2), w(t-1), w(t+1), w(t+2)$ from $w(t)$ as shown in Figure 2.11.

2.3 Neural Network Architectures

In this section, we discuss different useful network architectures, which are used in the works presented in this thesis. In the subsequent subsections, we discuss CNN, RNN, LSTM networks.

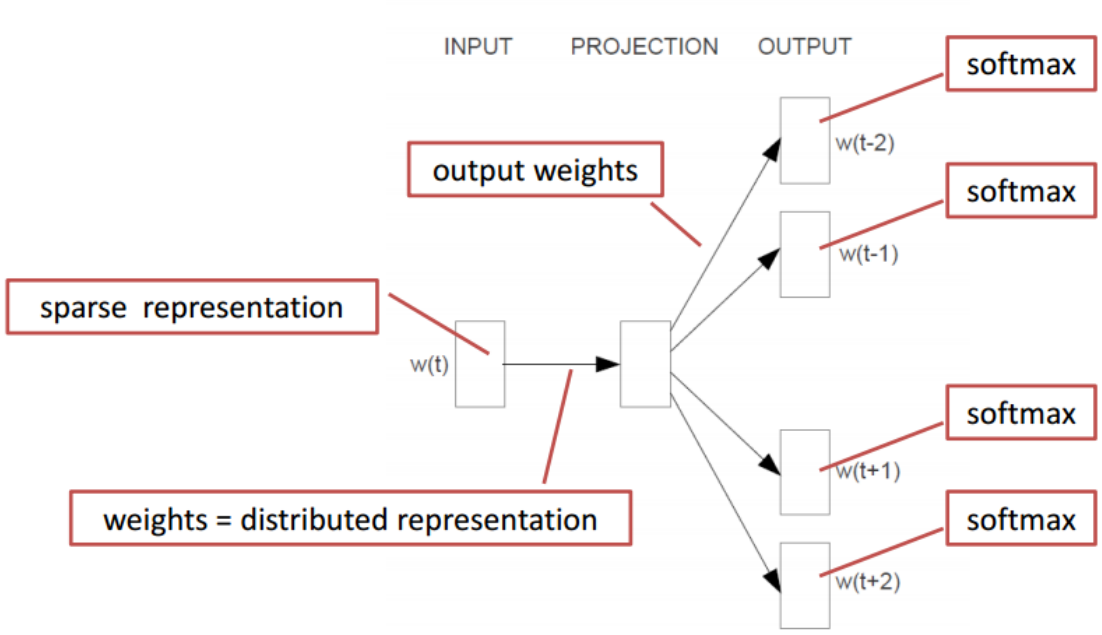


FIGURE 2.11: SkipGram model

2.3.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are a variant of feed-forward neural networks inspired by biology. In CNN, the arrangement of neurons is influenced by visual cortex of animals. Visual cortex is composed of complex arrangement of cells. These cells are sensitive to sub-regions in the visual field, called receptive field. These receptive fields cover the whole visual field and act as local filters over the visual input space. In CNN, this concept of spatially local correlation is exploited.

2.3.1.1 Convolution

CNN extracts features from images by repeated application of convolution filter on sub-regions over the whole image, as depicted in Figure 2.12. The output matrix obtained from convolution filter is called feature map. From a single image multiple such feature maps can be obtained using convolution filters having different parameter sets. So, k-th feature map can be represented by f^k, where

$$f^k_{ij} = ReLU((W^k * x)_{ij} + b^k), \tag{2.11}$$

$$ReLU(x) = \max(0, x). \tag{2.12}$$

Here, W^k is the weight and b^k is the bias of convolutional filter, i and j are row and column position of a neuron in feature map. $ReLU$ is called Rectified Linear Unit, widely used in deep networks, introduces non-linearity. All the feature maps produced can be represented as in Figure 2.13.

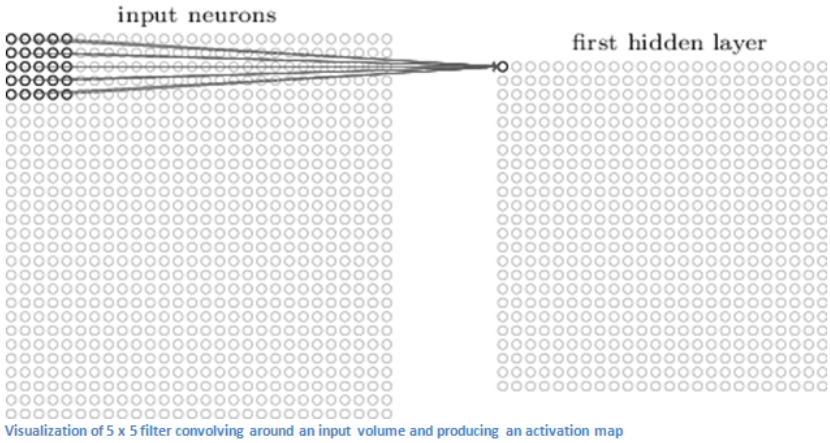


FIGURE 2.12: Feature Map derived from original image

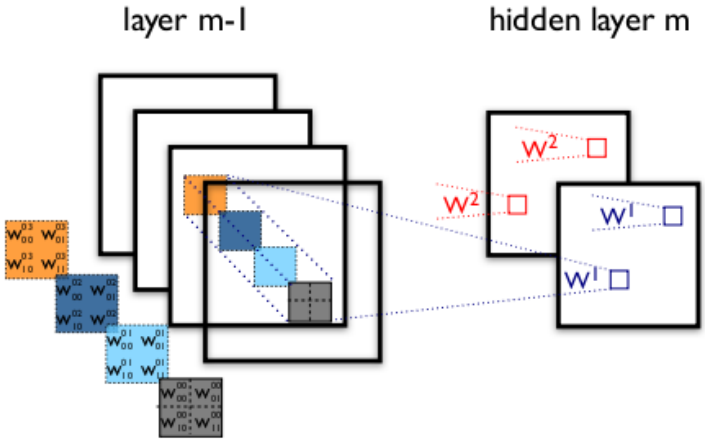


FIGURE 2.13: Feature Maps extracted from feature maps of previous layer

2.3.1.2 Max Pooling

Max pooling is an important part of CNN networks. It basically downscales the feature maps which reduces computation in the higher layers. Also, less important features are eliminated. Similar to convolution filter max pool operator applies to a certain window of neurons over the feature maps and keeps only the maximum value from each window. This creates downscaled feature maps.

2.3.1.3 Classification

The output of max pooling operation is flattened to a vector which is used as feature vector for the image. This vector is fed to MLP for final classification. Figure 2.14 depicts a full scale CNN for image classification.

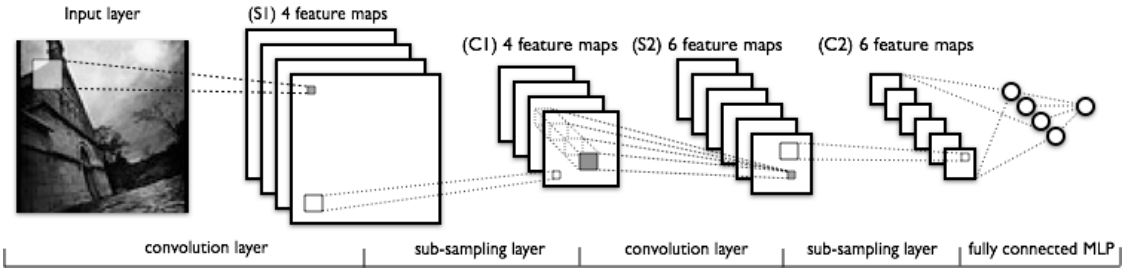


FIGURE 2.14: Full CNN for image classification

2.3.1.4 Applications of CNN in NLP

In NLP CNN is applied in a different way. CNN is usually employed in finding the sentence representation from word embeddings (Word embeddings are discussed in Section 2.2.1). Word embeddings are arranged in a column to create a matrix which is similar to an image. The height of the convolution filter is analogous to n-gram, i.e., filter with height n extracts n-gram features. Then we apply max pooling to the output feature maps and flatten the output to obtain sentence representation. If needed, multiple n-gram features can be used in a single network. Chapter 6 discusses a nice application of CNN in NLP.

2.3.2 3D Convolutional Neural Network (3D-CNN)

3D-CNN is 3D extension of regular 2D-CNN (Section 2.3.1). Here, the convolution operation is performed on a series of frames. Figure 2.15 depicts the architecture of a simple 3D-CNN network with 3D-Maxpooling and a classifier at the end.

Let $V \in \mathbb{R}^{c \times f \times h \times w}$ be a video, where c = number of channels in an image (e.g., $c = 3$ for RGB images), f = number of frames, h = height of the frames, and w = width of the frames. Again, we consider the 3D convolutional filter $F \in \mathbb{R}^{fm \times c \times fl \times fh \times fw}$, where fm = number of feature maps, c = number of channels, fd = number of frames (in other words depth of the filter), fh = height of the filter, and fw = width of the filter. Similar to 2D-CNN, F slides across video V and generates output

$$C \in \mathbb{R}^{fm \times c \times (f-fd+1) \times (h-fh+1) \times (w-fw+1)}.$$

Next, we apply max pooling to C to select only relevant features. The pooling will be applied only to the last three dimensions of the array C.

2.3.3 Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNN) are a class of neural networks where the connection among the neurons forms at least one directed cycle.

RNNs make use for sequential information. This property is very useful in NLP, since sentences are a sequence of words and documents are that of sentences. RNNs are called recurrent because it performs the same set of operations for every element

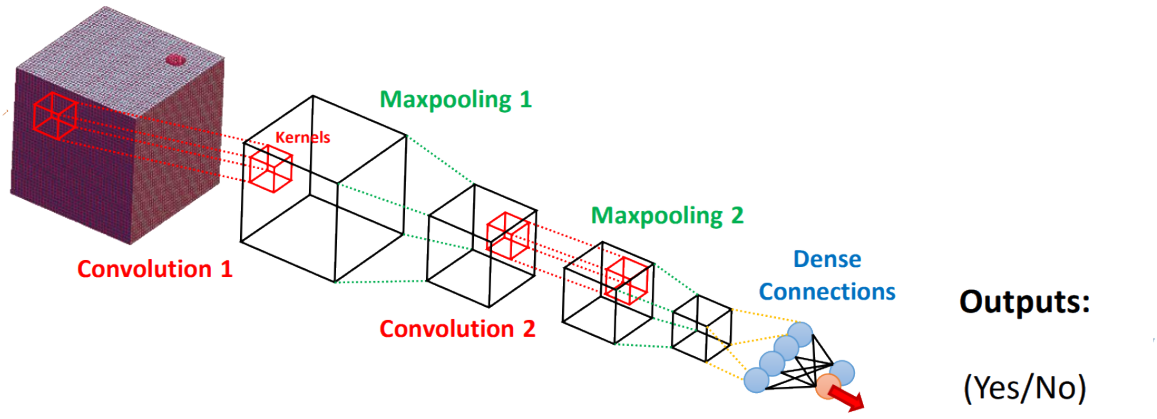


FIGURE 2.15: 3D-CNN

in a sequence. RNNs have the ability to memorize the information it has seen before in the sequence. However, in practice they can look back only a few steps, hence not very effective to model long term dependencies. Also it suffers from vanishing gradient problem. To overcome this limitation, a specially designed RNN called Long Short-Term Memory (LSTM) is used in most of the application, which is far less susceptible to these issues. We discuss LSTM in Section 2.3.4.

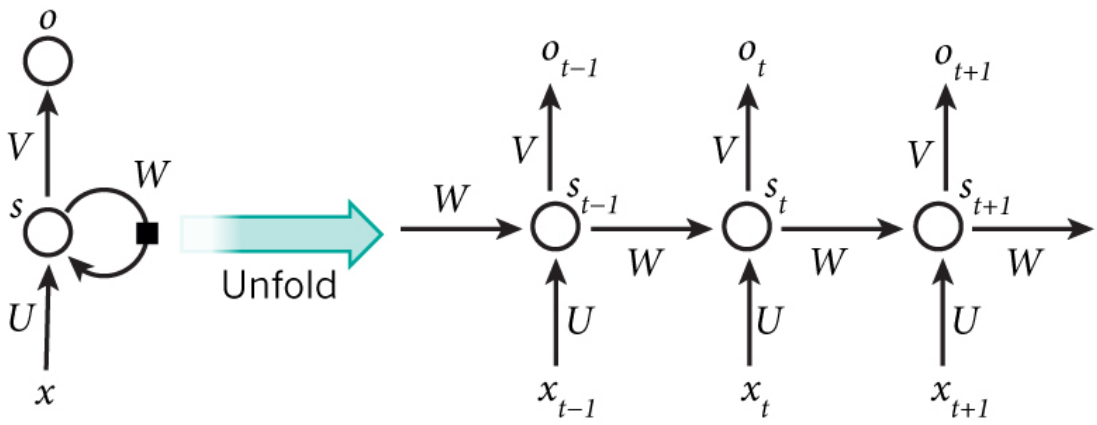


FIGURE 2.16: Recurrent Neural Network (RNN)

Figure 2.16 describes a simple RNN architecture being unfolded into a full network. Here, W, U, and V are network parameters. In this network, x_t is the input at time-step t, which could be word embedding such as word2vec or one-hot representation; s_t is hidden state at time-step t:

$$s_t = \mathcal{N}(U\ x_t + W\ s_{t-1}), \tag{2.13}$$

where $\mathcal{N}$ is non-linearity such as $ReLU$ or $\tanh$, the initial state s_{-1} being usually initialized to zero; o_t is the output at step t:

$$o_t = \text{softmax}(V\ s_t), \tag{2.14}$$

which is the probability of the input belonging to a particular class.

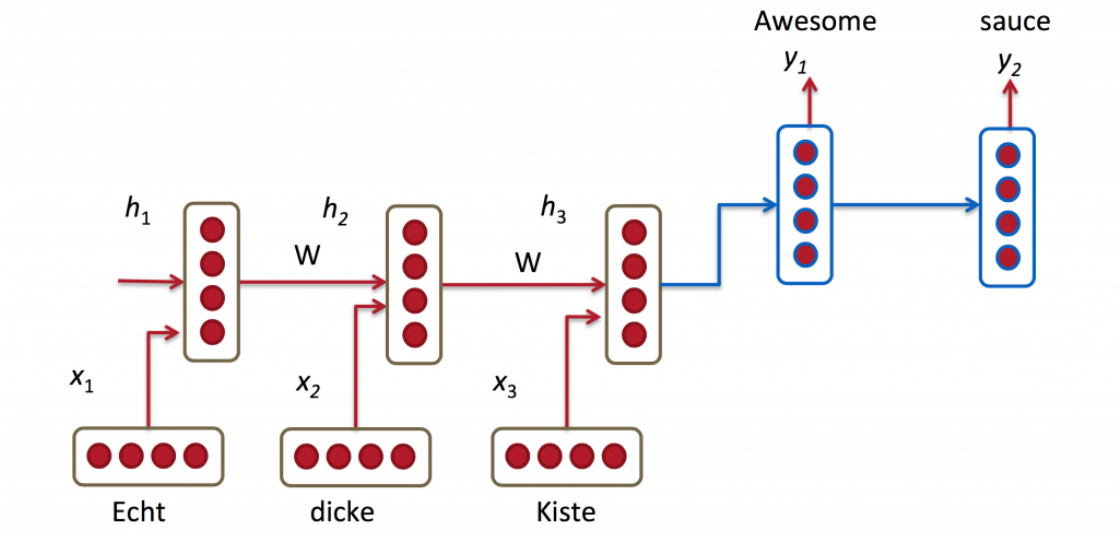


FIGURE 2.17: RNN for machine translation

Figure 2.17 shows an RNN for translating German to English.

2.3.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) [4] is specially designed RNN which is less prone to vanishing gradient problem. Also, it is better capable of modeling long distance dependencies in a sequence. These features make it really useful for NLP applications where sentences and documents are sequential by nature.

LSTM is designed in such a way that it only memorizes relevant part of the sequence which it has seen so far. This behavior is enabled by the usage of gates in the network, which decides what to keep in the memory based on current input and hidden state.

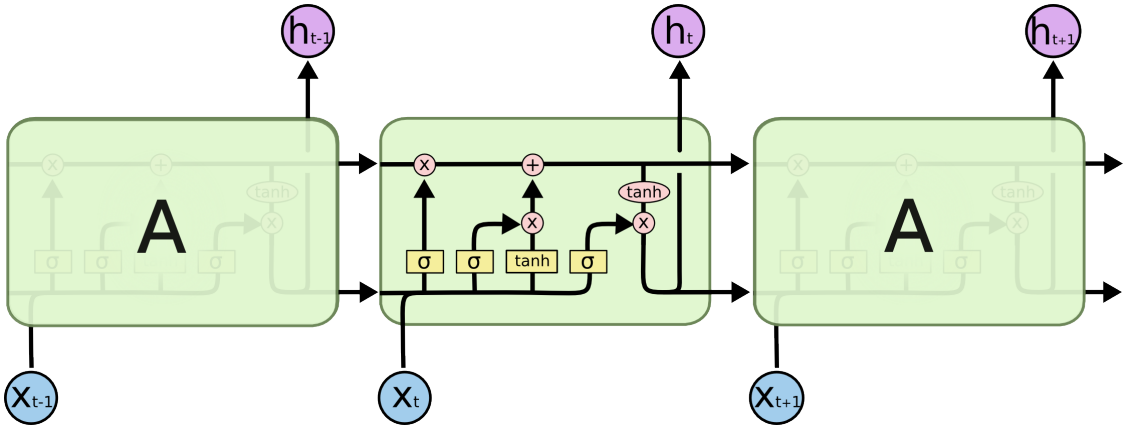


FIGURE 2.18: Long Short-Term Memory (LSTM)

Figure 2.18 shows the basic architecture of an LSTM cell. The same repeats over different time-steps. We describe the governing equations of LSTM below:

$$i_t = \sigma(W_i\, x_t + U_i\, h_{t-1} + b_i), \tag{2.15}$$
$$o_t = \sigma(W_o\, x_t + U_o\, h_{t-1} + b_o), \tag{2.16}$$
$$f_t = \sigma(W_f\, x_t + U_f\, h_{t-1} + b_f), \tag{2.17}$$
$$C_{in} = \tanh(W_c\, x_t + U_c\, h_{t-1} + b_c), \tag{2.18}$$
$$C_t = i_t * C_{in} + f_t * C_{t-1}, \tag{2.19}$$
$$h_t = o_t * \tanh(C_t), \tag{2.20}$$

where i_t = input gate at time-step t, o_t = output gate at time-step t, f_t = forget gate at time-step t, C_{in} = candidate state value at time-step t, C_t = cell state at time-step t (memory), h_t = hidden output of LSTM cell at time-step t. The gate values are dependent on the current input and output of previous cell. It means, that it considers previous information along with current one to make decision about that to keep in the memory and what to forget. In equation (2.19), new cell memory is formed by forgetting some part of current memory and incorporating some part of new input x_t. In equation (2.20), LSTM cell outputs part of the new memory as output. The output of final cell is usually taken as the final output, which represents the whole sequence.

In some applications, a variant of LSTM is used called peephole LSTM. The difference is the gates are function of the cell value of previous cell rather than output of previous cell.

2.4 Training Neural Networks

In this section, we discuss the algorithms for training neural networks in supervised scenario.

2.4.1 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a optimization method that minimizes a differentiable objective function by iteratively updating the parameters by a fraction of its gradients until a state of saturation is reached. Figure 2.19 depicts the gradual descent to the absolute minima for a convex function. Algorithm 1 summarizes SGD.

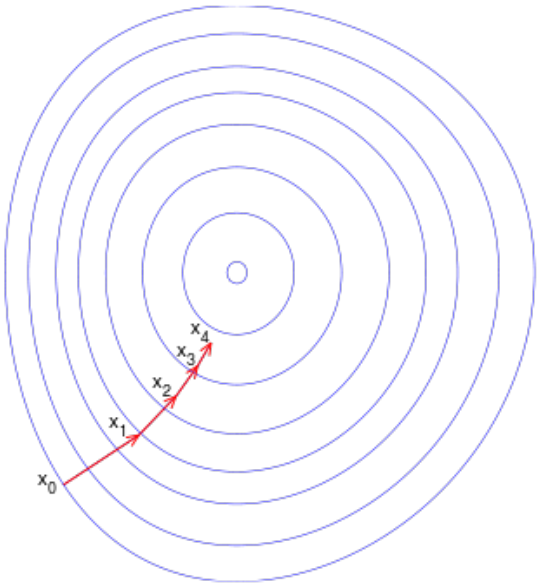


FIGURE 2.19: Gradient Descent for a convex function

Algorithm 1 Stochastic Gradient Descent algorithm

1: **procedure** SGD(J, w, η, ϵ)▷ J = Objective function which is a function of w, η = learning rate, ϵ = change tolerance
2: Initialize parameters w with random values close to zero
3: **repeat**
4: $w^{(t)} = w^{(t-1)} - \eta \nabla J(w^{(t-1)})$
5: **until** $|J^{(t)} - J^{(t-1)}| < \epsilon$ ▷ $J^{(t)}$ = value of J after iteration t
6: **return** w

The values of hyper-parameters, which are η and ϵ, are to be supplied by the user. ϵ should be quite small, like 0.0001. η (or learning-rate) should chosen very carefully to a small value like 0.001 and should redefine through many SGD executions. Too small learning-rate will lead to prolonged optimization time, even may not converge in realistic amount of time. Too large η will lead to oscillation of J, in other words the minima can be missed through-out the iterations.

In this section, we left out the back-propagation algorithm, which is the procedure to calculate gradient for neural networks.

2.5 Model Validation Techniques

Cross Validation Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

In **k-fold cross-validation**, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

For classification problems, one typically uses stratified k-fold cross-validation, in which the folds are selected so that each fold contains roughly the same proportions of class labels.

Bootstrapping Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid over fitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.

Bootstrapping is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. One standard choice for an approximating distribution is the empirical distribution function of the observed data. In the case where a set of observations can be assumed to be from an independent and identically distributed population, this can be implemented by constructing a number of re-samples with replacement, of the observed dataset (and of equal size to the observed dataset).

2.6 Model Evaluation Techniques

In this section we describe some of the model evaluation techniques to calculate the quality of the regression and classification models.

2.6.1 Evaluating Regression Quality

Mean Squared Error Validation Techniques In Machine Learning, the Mean Squared Error (MSE) or mean squared deviation (MSD) of an estimator measures the average of the squares of the errors or deviations, that is, the difference between the estimator and what is estimated. MSE is a risk function, corresponding to the expected value of the

squared error loss or quadratic loss. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.If $\hat{Y}$ is a vector of n predictions, and Y is the vector of observed values corresponding to the inputs to the function which generated the predictions, then the MSE of the predictor can be estimated by

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

I.e., the MSE is the mean $(\frac{1}{n}\sum_{i=1}^{n})$ of the square of the errors $((\hat{Y}_i - Y_i)^2)$. This is an easily computable quantity for a particular sample (and hence is sample-dependent).

2.6.2 Evaluating Classification Techniques

Precision Precision and recall in information retrieval are defined in terms of retrieved documents i.e., assuming that a search query is made on a search engine, the documents retrieved are used to define precision. Therefore in terms of retrieved documents,precision is the fraction of retrieved documents that are relevant to the query. The formula used for precision is :- $precision = \frac{|(RelaventDocuments) \cap (RetrievedDocuments)|}{|(RetrievedDocuments)|}$

Recall Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved. The formula is :-

$$Recall = \frac{|(RelaventDocuments) \cap (RetrievedDocuments)|}{|(RelaventDocuments)|}$$

F-Score A measure that combines precision and recall is the harmonic mean of precision and recall is called the F-Score. The formula :-

$$FScore = \frac{Precision.Recall}{Precision + Recall}$$

Accuracy is a weighted arithmetic mean of Precision and Inverse Precision (weighted by Bias) as well as a weighted arithmetic mean of Recall and Inverse Recall. Accuracy is given by the below formula:-

$$Accuracy = \frac{\sum TruePositive + \sum TrueNegative}{TotalPopulation}$$

Chapter 3

Multimodal Sentiment Analysis with Hierarchical Fusion

3.1 Introduction

Internet is easily available to all people since the dawn of this millennium. This has resulted in the rise of numerous social media platforms, such as YouTube, Facebook, Instagram etc., where people share their opinions on all kinds of topics in the form of posts, images and videos. Proliferation of smartphones and tablets has greatly boosted content sharing. Since recently, people increasingly share their opinions on newly released product or on other topics in form of video reviews or comments. This is an excellent opportunity for large companies to capitalize on by extracting user sentiment, suggestions, and complaints on their products from these video reviews.

Videos convey information through three channels: audio, video and text (in form of speech). Mining opinion from this plethora of multimodal data calls for a solid multimodal sentiment analysis technology. One of the major problems faced in multimodal sentiment analysis is the fusion of features pertaining to different modalities. The majority of the recent works in multimodal sentiment analysis have simply concatenated the feature vectors for different modalities. However, this does not take into consideration that different modalities may carry conflicting information. We hypothesize that the method presented in this chapter deals with this issue better, and present experimental evidence showing improvement over simple concatenation of feature vectors.

In our method, we first transform the feature vectors of the three modalities to have the same dimensionality. We assume that these transformed vectors contain abstract features representing the attributes relevant to sentiment classification. Next, we compare and combine each bimodal combination of these abstract features using fully-connected layers of a neural network. This results in fused bimodal feature vectors. Finally, we combine these bimodal vectors into a trimodal vector using, again, fully-connected layers. We show that the feature vector obtained in this manner is more useful for the sentiment classification task.

3.2 Related Work

As to text-based sentiment analysis methods, they can be classified into two major classes: knowledge-based and statistics-based methods [5]. These groups of methods have significantly different approaches and vary in importance with time. Knowledge-based systems were more widely used in the past, while recently, statistics-based systems have dominated overwhelmingly; especially, supervised statistical methods [6], [7] has played a major role.

As to other modalities, Ekman [8] in the early 1970s showed through extensive studies that facial expressions provide sufficient clues for emotion detection. Recent studies [9] have started to focus also on acoustic features such as pitch, intensity of utterance, bandwidth, duration etc.

In the field of emotion recognition, early works by De Silva et al. [10] and Chen et al. [11] showed that fusion of audio and visual systems, creating a bimodal signal, yielded a higher accuracy than any unimodal system. Such fusion has been analyzed at both feature level [12] and decision level [13].

Although there is much work done on audio-visual fusion for emotion recognition, exploring contribution of text along with audio and visual modalities in multimodal emotion detection has been little explored. Wollmer et al. [14] and Rozgic et al. [15] fused information from audio, visual and textual modalities to extract emotion and sentiment. Metallinou et al. [16] and Eyben et al. [17] fused audio and textual modalities for emotion recognition. Both approaches relied on a feature-level fusion. Wu et al. [18] fused audio and textual clues at decision level. The current state of the art, set forth by Poria et al. [19], uses CNN to extract features from the modalities and then employs multiple-kernel learning (MKL) for sentiment analysis and emotion recognition.

3.3 Our Method

Our method for polarity detection consists in the following. First, we use a feature extraction method for each modality: audio, video, and text. Then, we fuse the features from the three modalities into a unified vector space. Finally, we train a classifier on such multimodal feature vector. In the sequel we describe each step.

3.3.1 Unimodal Feature Extraction

In this section, we discuss the method of feature extraction for three different modalities: audio, video, and text.

3.3.1.1 Textual Feature Extraction

The source of textual modality is the transcription of the spoken words. To extract features from the textual modality, we use a deep convolutional neural network (CNN) [20]. First, we represent each utterance as a concatenation of vectors of the constituent words,

which in our experiments were the publicly available 300-dimensional `word2vec` vectors trained on 100 billion words from Google News [1].

The convolution kernels are thus applied to these concatenated word vectors instead of individual words. Each utterance is wrapped in a window of 50 words, which serves as the input to the CNN. The CNN has two convolutional layers; the first layer has two kernels of size 3 and 4, with 50 feature maps each, and the second layer has a kernel of size 2 with 100 feature maps.

The convolution layers are interleaved with max-pooling layers of window 2×2. This is followed by a fully-connected layer of size 500 and softmax output. We use rectified linear unit (ReLU) [21] as the activation function. The activation values of the fully-connected layer are taken as the features of utterances for text modality. The convolution of the CNN over the utterance learns abstract representations of the phrases equipped with implicit semantic information, which with each successive layer spans over increasing number of words and ultimately the entire utterance.

3.3.1.2 Audio Feature Extraction

Audio features are extracted at 30 Hz frame-rate with a sliding window of 100 ms. To compute the features, we use openSMILE [22], an open-source software that automatically extracts audio features such as pitch and voice intensity. Voice normalization is performed and voice intensity is thresholded to identify samples with and without voice. Z-standardization is used to perform voice normalization. Both of these tasks were performed using openSMILE.

The features extracted by openSMILE consist of several Low Level Descriptors (LLD) and statistical functionals of them. Some of the functionals are *amplitude mean, arithmetic mean, root quadratic mean, standard deviation, flatness, skewness, kurtosis, quartiles, inter-quartile ranges, linear regression slope* etc. Specifically, we use "IS13-ComParE" configuration file in openSMILE. Taking into account all functionals of each LLD, we obtained 6372 features.

3.3.1.3 Visual Feature Extraction

We use 3D-CNN to obtain visual features from the video. We hypothesize that 3D-CNN will not only be able to learn relevant features from each frame, but will also be able to learn the changes among given number of consecutive frames.

In the past, 3D-CNN has been successfully applied to object classification on 3D data [23]. Its ability to achieve state-of-the-art results motivated us to use it.

Let $vid \in \mathbb{R}^{c \times f \times h \times w}$ be a video, where c = number of channels in an image (in our case c = 3, since we consider only RGB images), f = number of frames, h = height of the frames, and w = width of the frames. Again, we consider the 3D convolutional filter $filt \in \mathbb{R}^{fm \times c \times fl \times fh \times fw}$, where fm = number of feature maps, c = number of channels, fd = number of frames (in other words depth of the filter), fh = height of the filter, and fw = width of the filter. Similar to 2D-CNN, $filt$ slides across video vid and generates output $convout \in \mathbb{R}^{fm \times c \times (f-fd+1) \times (h-fh+1) \times (w-fw+1)}$. Next, we apply max pooling to

$convout$ to select only relevant features. The pooling will be applied only to the last three dimensions of the array $convout$.

In our experiments, we obtained best results with 32 feature maps (fm) with the filter-size of $5 \times 5 \times 5$ (or $fd \times fh \times fw$). In other words, the dimension of the filter is $32 \times 3 \times 5 \times 5 \times 5$ (or $fm \times c \times fd \times fh \times fw$). Subsequently, we apply max pooling on the output of convolution operation, with window-size being $3 \times 3 \times 3$. This is followed by a dense layer of size 300 and softmax. The activations of this dense layer are finally used as the video features for each utterance.

3.3.2 Hierarchical Multimodal Fusion

In this section, the mechanism to fuse the unimodal features vectors:

- $f_a \in \mathbb{R}^{d_a}$ (acoustic feature),
- $f_v \in \mathbb{R}^{d_v}$ (visual feature),
- $f_t \in \mathbb{R}^{d_t}$ (textual feature)

obtained in section 3.3.1 into a single multimodal feature vector is described.

The unimodal features may have different dimensionalities, i.e. $d_a \neq d_v \neq d_t$. So, we map them to the same dimensionality, say d_m (we obtained best results with $d_m = 250$), using fully-connected layer as follows:

$$g_a = \tanh(W_a f_a + b_a), \tag{3.1}$$
$$g_v = \tanh(W_v f_v + b_v), \tag{3.2}$$
$$g_t = \tanh(W_t f_t + b_t), \tag{3.3}$$

where $W_a \in \mathbb{R}^{d_m \times d_a}$, $b_a \in \mathbb{R}^{d_m}$, $W_v \in \mathbb{R}^{d_m \times d_v}$, $b_v \in \mathbb{R}^{d_m}$, $W_t \in \mathbb{R}^{d_m \times d_t}$, and $b_t \in \mathbb{R}^{d_m}$. We can represent the mapping for each dimension as

$$g_x = (c_1^x, c_2^x, c_3^x, \cdots, c_{d_m}^x), \tag{3.4}$$

where $x \in \{v, a, t\}$ and c_l^x are scalars for all $l = 1, 2, \cdots, d_m$. We can see these values c_l^x as more abstract feature values derived from fundamental feature values (which are the components of f_a, f_v, and f_t). For example, an abstract feature can be the angriness of a speaker in a video. We can infer the degree of angriness from visual features (f_v; facial muscle movements), acoustic features (f_a; pitch, raised voice etc.), or textual features (f_t; the language, choice of words etc.). Therefore, the degree of angriness can be represented by c_l^x where $x = a, v, t$ and l is some fixed integer between 1 and d_m.

Now, the evaluation of abstract feature values from all the modalities may not have the same merit or may even contradict each other. So, we need the network to make comparison among the feature values derived from different modalities to make a more refined evaluation of the degree of anger. To achieve so, we take each bimodal combination (which are audio-video, audio-text, and video-text) at a time and compare-and-combine each of their respective abstract feature values (i.e. c_l^v with c_l^t, c_l^v with c_l^a, and

c_l^a with c_l^t) using fully-connected layers as follows:

$$i_l^{va} = \tanh(w_l^{va}.[c_l^v, c_l^a]^\intercal + b_l^{va}), \tag{3.5}$$
$$i_l^{at} = \tanh(w_l^{at}.[c_l^a, c_l^t]^\intercal + b_l^{at}), \tag{3.6}$$
$$i_l^{vt} = \tanh(w_l^{vt}.[c_l^v, c_l^t]^\intercal + b_l^{vt}), \tag{3.7}$$

where $w_l^{va} \in \mathbb{R}^2$, b_l^{va} is scalar, $w_l^{at} \in \mathbb{R}^2$, b_l^{at} is scalar, $w_l^{vt} \in \mathbb{R}^2$, and b_l^{vt} is scalar, for $l = 1, 2, \cdots, d_m$. We hypothesize that it will enable the network to compare the decisions from each modality against the others and help achieve a better fusion of modalities.

Bimodal fusion Equations (3.5) to (3.7) are used for bimodal fusion. The bimodal fused features for video-audio, audio-text, video-text are defined as

$$F^{va} = (i_1^{va}, i_2^{va}, \cdots, i_{d_m}^{va}), \tag{3.8}$$
$$F^{at} = (i_1^{at}, i_2^{at}, \cdots, i_{d_m}^{at}), \tag{3.9}$$
$$F^{vt} = (i_1^{vt}, i_2^{vt}, \cdots, i_{d_m}^{vt}) \tag{3.10}$$

respectively.

Trimodal fusion We combine all three modalities using fully-connected layers as follows:

$$z_l = \tanh(w_l.[i_l^{va}, i_l^{at}, i_l^{vt}]^\intercal + b_l), \tag{3.11}$$

where $w_l \in \mathbb{R}^3$ and b_l is a scalar for all $l = 1, 2, \cdots, d_m$. So, we define the fused features as

$$F^{avt} = (z_0, z_2, \cdots, z_{d_m}). \tag{3.12}$$

3.3.3 Classification

In order to perform classification, we feed the fused features F to a softmax layer with $C = 2$ outputs. The classifier can be described as follows:

$$\mathcal{P} = \text{softmax}(W_{softmax} F^q + b_{softmax}), \tag{3.13}$$
$$\hat{y} = \underset{j}{\text{argmax}}(\mathcal{P}[j]), \tag{3.14}$$

where $W_{softmax} \in \mathbb{R}^{C \times d_m}$, $b_{softmax} \in \mathbb{R}^C$, $\mathcal{P} \in \mathbb{R}^C$, j = class value (0 or 1), $q = va, at, vt, avt$, and $\hat{y}$ = estimated class value.

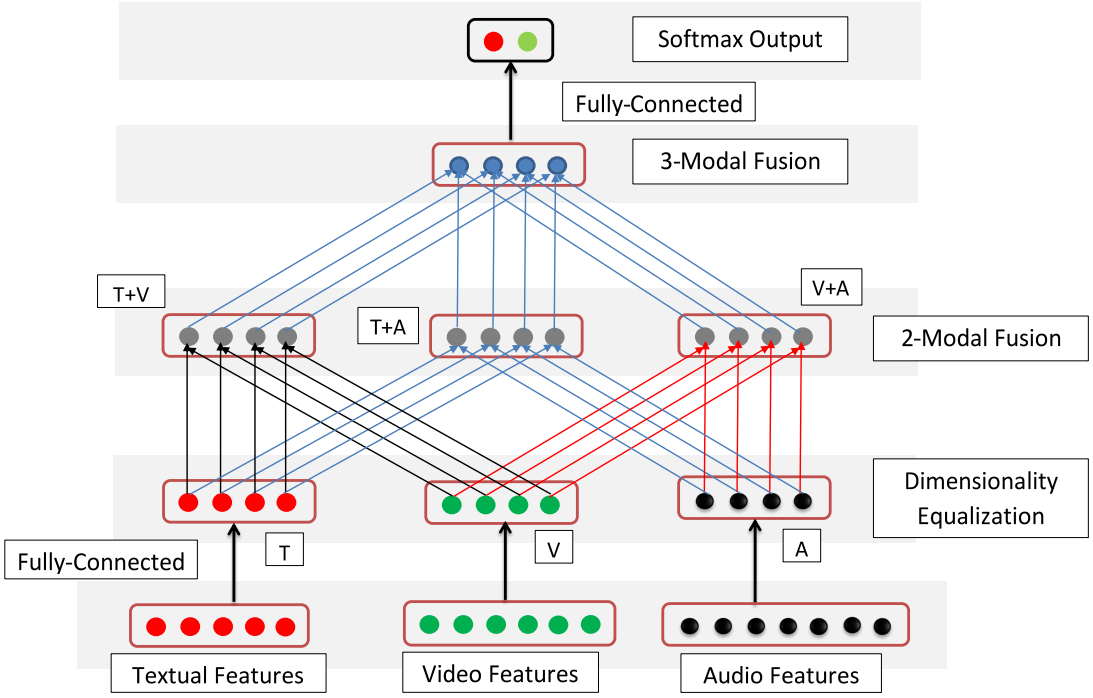


FIGURE 3.1: Hierarchical Fusion

3.3.4 Training

We employ categorical cross-entropy as loss function (J) for training,

$$J = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{C-1} y_{ij}\log \mathcal{P}_i[j], \tag{3.15}$$

where N = number of samples, i = index of a sample, j = class value, and

$$y_{ij} = \begin{cases} 1, & \text{if expected class value of sample } i \text{ is } j \\ 0, & \text{otherwise.} \end{cases}$$

AdaDelta [24] is used as optimizer due to its ability to adapt learning rate for each parameter individually. We train the network for 300-500 epochs, where we optimize the parameter set $\theta = \{W_a, W_v, W_t, b_a, b_v, b_t, w_1^{va}, w_2^{va}, \cdots, w_{d_m}^{va}, w_1^{at}, w_2^{at}, \cdots, w_{d_m}^{at}, w_1^{vt}, w_2^{vt}, \cdots, w_{d_m}^{vt}, b_1^{va}, b_2^{va}, \cdots, b_{d_m}^{va}, b_1^{at}, b_2^{at}, \cdots, b_{d_m}^{at}, b_1^{vt}, b_2^{vt}, \cdots, b_{d_m}^{vt}, w_1, w_2, \cdots, w_{d_m}, b_1, b_2, \cdots, b_{d_m}, W_{softmax}, b_{softmax}\}$. Algorithm 2 summarizes the proposed method.

Algorithm 2 Proposed Algorithm

procedure TRAINANDTESTMODEL(U, V) ▷ U = train set, V = test set
  Unimodal feature extraction:
  for `i:[1,`N`]` **do** ▷ extract baseline features
    $x_i^a \leftarrow AudioFeatures(u_i)$
    $x_i^v \leftarrow VideoFeatures(u_i)$
    $x_i^t \leftarrow TextFeatures(u_i)$
  Multimodal Fusion:
  Dimensionality equalization:
  $g^a \leftarrow MapToSpace(x^a)$
  $g^v \leftarrow MapToSpace(x^v)$
  $g^t \leftarrow MapToSpace(x^t)$
  Bimodal fusion:
  $f^{va} \leftarrow BimodalFusion(g^v, g^a)$
  $f^{at} \leftarrow BimodalFusion(g^a, g^t)$
  $f^{vt} \leftarrow BimodalFusion(g^v, g^t)$
  Trimodal fusion:
  $F^{avt} \leftarrow TrimodalFusion(f^{va}, f^{at}, f^{vt})$
  Train the above network for x_a, x_v, and x_t.
  $TestModel(V)$
  return F^{avt}
procedure TESTMODEL(V)
  Similar to training phase, V is passed through the learnt models to get the features and classification outputs. Section 3.3.4 mentions the trainable parameters (θ).
procedure TRIMODALFUSION(f^{z_1}, f^{z_2}, f^{z_3}) ▷ for modality combination z_1, z_2, and z_3, where $z_1 \neq z_2 \neq z_3$
  for `i:[1,`d_m`]` **do**
    $F_i \leftarrow \tanh(w_i.[f_i^{z_1}, f_i^{z_2}, f_i^{z_3}]^\intercal + b_i)$
  $F \leftarrow (F_1, F_2, \cdots, F_{d_m})$
  return F
procedure BIMODALFUSION(g^{z_1}, g^{z_2}) ▷ for modality z_1 and z_2, where $z_1 \neq z_2$
  for `i:[1,`d_m`]` **do**
    $f_i^{z_1 z_2} \leftarrow \tanh(w_i^{z_1 z_2}.[g_i^{z_1}, g_i^{z_2}]^\intercal + b_i^{z_1 z_2})$
  $f^{z_1 z_2} \leftarrow (f_1^{z_1 z_2}, f_2^{z_1 z_2}, \cdots, f_{d_m}^{z_1 z_2})$
  return $f^{z_1 z_2}$
procedure MAPTOSPACE(x^z) ▷ for modality z
  $g^z \leftarrow \tanh(W_z x^z + b_z)$
  return g^z

TABLE 3.1: Accuracy for Different Combinations of Modalities; bold font signifies best accuracy for the corresponding feature-set and modality/modalities, where T = text, V = video, and A = audio

Modality Combination	Poria et al. (2015) feature-set		Our feature-set	
	Poria et al. (2015)	Our method	Early fusion	Our method
T	–	–	75%	
V	–	–	55.31%	
A	–	–	56.91%	
T+V	73.2%	**74.4%**	77.13%	**77.79%**
T+A	73.2%	**74.15%**	77.13%	**77.26%**
A+V	55.7%	**57.5%**	56.52%	**56.78%**
A+V+T	73.55%	**74.6%**	76.99%	**77.92%**

3.4 Experimental Results

3.4.1 Dataset Used

Most research-works in multimodal sentiment analysis are performed on datasets where train and test splits may share certain speakers. Since, each individual has an unique way of expressing emotions and sentiments, finding generic and person-independent features for sentiment analysis is crucial.

CMU-MOSI CMU-MOSI dataset [25] is rich in sentimental expressions, where 89 people review various topics in English. The videos are segmented into utterances where each utterance is annotated with scores between -3 (strongly negative) and $+3$ (strongly positive) by five annotators. We took the average of these five annotations as the sentiment polarity and considered only two classes (positive and negative). Given every individual's unique way of expressing sentiments, real world applications should be able to model generic person independent features and be robust to person variance. To this end, we perform person-independent experiments to emulate unseen conditions. Our train/test splits of the dataset are completely disjoint with respect to speakers. The train/validation set consists of the first 62 individuals in the dataset. The test set contains opinionated videos by rest of the 31 speakers. In particular, 1447 and 752 utterances are used for training and test, respectively.

3.4.2 Baselines

We consider the following strong baselines for comparison against the proposed method.

Poria et al. (2015). We have implemented and compared our method with the current state-of-the-art approach, proposed by Poria et al. [19]. In their approach, they extracted visual features using CLM-Z, audio features using openSMILE, and textual features using CNN. Multiple Kernel Learning (MKL) was then applied to the features obtained

from concatenation of the unimodal features. However, they did not conduct speaker independent experiments.

Poria et al. (2015) In order to perform a fair comparison with Poria et al. (2015), we employ the proposed fusion method on the features extracted by Poria et al. (2015).

Early-fusion. We extract unimodal features (Section 3.3.1) and simply concatenate them to produce multimodal features. Followed by Support Vector Machine (SVM) being applied on this feature vector for the final sentiment classification.

3.4.3 Results and Discussion

The results of our experiments are presented in Table 3.1. We evaluated our model with two feature-sets: the feature-set used in Poria et al. (2015) [19] and the set of unimodal features discussed in Section 3.3.1.

Our model outperformed Poria et al. (2015), which employed MKL, for all bimodal and trimodal scenarios by a margin of 1-1.8%. This leads us to present two observations – firstly the features used in Poria et al. (2015) is inferior to the features extracted in our approach and secondly our proposed hierarchical fusion method is better than their fusion method.

It is already established in the literature [19], [26] that multimodal analysis outperforms unimodal analysis. We also observe the same trend in our experiments where trimodal and bimodal classifiers outperform unimodal classifiers. The textual modality performed best among others with a higher unimodal classification accuracy of 75%. Although other modalities contribute to improve the performance of multimodal classifiers, that contribution is little in compare to the textual modality.

On the other hand, we compared our model with early-fusion (Section 3.4.2) for aforementioned feature-sets (Section 3.3.1). The proposed fusion mechanism surpassed early-fusion for all combination of modalities in a consistent fashion. This supports our hypothesis that the proposed hierarchical fusion method captures the inter-relation among the modalities and produce better performance vector than early-fusion. Text is the strongest individual modality, and we observe that text-modality paired with remaining two modalities results in consistent performance improvement.

Overall, the results give a strong indication that the comparison among the abstract feature values dampens the effect of less important modalities, which was our hypothesis. For example, we can notice that for early fusion T+V and T+A both yield the same performance. However, with our method text with video performs better than text with audio, which is more aligned with our expectations, since facial muscle movements usually carry more emotional nuances than voice.

3.5 Conclusions

There are only a few research-works performed on multimodal fusion strategy. A novel and comprehensive fusion strategy is presented in this chapter. Our method outperformed widely used early-fusion in two different instances. In the future, we plan to apply this method with Long Short-Term Memory (LSTM), which considers inter-utterance dependencies.

Chapter 4

Contextual Multimodal Sentiment Analysis

4.1 Introduction

Emotion recognition and sentiment analysis have become a new trend in social media, helping users to automatically extract the opinions expressed in user-generated content, especially videos. Thanks to the high availability of computers and smartphones, and the rapid rise of social media, consumers tend to record their reviews and opinions about products or films and upload them on social media platforms, such as YouTube or Facebook. Such videos often contain comparisons, which can aid prospective buyers make an informed decision.

The primary advantage of analyzing videos over text is the surplus of behavioral cues present in vocal and visual modalities. The vocal modulations and facial expressions in the visual data, along with textual data, provide important cues to better identify affective states of the opinion holder. Thus, a combination of text and video data helps to create a better emotion and sentiment analysis model [27].

Recently, a number of approaches to multimodal sentiment analysis, producing interesting results, have been proposed [14], [19], [26]. However, there are major issues that remain unaddressed, such as the role of speaker-dependent versus speaker-independent models, the impact of each modality across the dataset, and generalization ability of a multimodal sentiment classifier. Leaving these issues unaddressed has presented difficulties in effective comparison of different multimodal sentiment analysis methods.

An utterance is a unit of speech bound by breathes or pauses. Utterance-level sentiment analysis focuses on tagging every utterance of a video with a sentiment label (instead of assigning a unique label to the whole video). In particular, utterance-level sentiment analysis is useful to understand the sentiment dynamics of different aspects of the topics covered by the speaker throughout his/her speech. The true meaning of an utterance is relative to its surrounding utterances.

In this chapter, we consider such surrounding utterances to be the context, as the consideration of temporal relation and dependency among utterances is key in human-human communication. For example, the MOSI dataset [25] contains a video, in which a girl reviews the movie ‘Green Hornet’. At one point, she says “The Green Hornet did something similar”. Normally, doing something similar, i.e., monotonous or repetitive

might be perceived as negative. However, the nearby utterances "It engages the audience more", "they took a new spin on it", "and I just loved it" indicate a positive context.

In this chapter, we discard the oversimplifying hypothesis on the independence of utterances and develop a framework based on Long Short-Term Memory (LSTM) to extract utterance features that also consider surrounding utterances.

Our model enables consecutive utterances to share information, thus providing contextual information in the classification process. Experimental results show that the proposed framework has outperformed the state of the art on benchmark datasets by $5 - 10\%$. The chapter is organized as follows: Section 4.2 provides a brief literature review on multimodal sentiment analysis; Section 4.3 describes the proposed method in detail; experimental results and discussion are shown in Section 4.4; finally, Section 4.5 concludes the chapter.

4.2 Related Work

Text-based sentiment analysis systems can be broadly categorized into knowledge-based and statistics-based systems [5]. While the use of knowledge bases was initially more popular for the identification of emotions and polarity in text, sentiment analysis researchers have recently been using statistics-based approaches, with a special focus on supervised statistical methods [6], [7].

In 1970, Ekman [8] carried out extensive studies on facial expressions which showed that universal facial expressions are able to provide sufficient clues to detect emotions. Recent studies on speech-based emotion analysis [9] have focused on identifying relevant acoustic features, such as fundamental frequency (pitch), intensity of utterance, bandwidth, and duration.

As for fusing audio and visual modalities for emotion recognition, two of the early works were done by De Silva et al. [10] and Chen et al. [11]. Both works showed that a bimodal system yielded a higher accuracy than any unimodal system. More recent research on audio-visual fusion for emotion recognition has been conducted at either feature level [12] or decision level [13].

While there are many research papers on audio-visual fusion for emotion recognition, only a few have been devoted to multimodal emotion or sentiment analysis using textual clues along with visual and audio modalities. Wollmer et al. [14] and Rozgic et al. [15], [28] fused information from audio, visual, and textual modalities to extract emotion and sentiment. Metallinou et al. [16] and Eyben et al. [17] fused audio and textual modalities for emotion recognition. Both approaches relied on a feature-level fusion. Wu et al. [18] fused audio and textual clues at decision level.

4.3 Our Method

In this work, we propose a LSTM network that takes as input all utterances in a video and extracts contextual unimodal and multimodal features by modeling the dependencies among the input utterances. Below, we propose an overview of the method -

1. **Context-Independent Unimodal Utterance-Level Feature Extraction**

   First, the unimodal features are extracted without considering the contextual information of the utterances (Section 4.3.1). Table 4.1 presents the feature extraction methods used for each modality.

2. **Contextual Unimodal and Multimodal Classification**

   The context-independent unimodal features (from Step 1) are then fed into a LSTM network (termed contextual LSTM) that allows consecutive utterances in a video to share semantic information in the feature extraction process (which provides context-dependent unimodal and multimodal classification of the utterances). We experimentally show that this proposed framework improves the performance of utterance-level sentiment classification over traditional frameworks.

Videos, comprising of its constituent utterances, serve as the input. We represent the dataset as U:

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & ... & u_{1,L_1} \\ u_{2,1} & u_{2,2} & u_{2,3} & ... & u_{2,L_2} \\ . & . & . & ... & . \\ u_{M,1} & u_{M,2} & u_{M,3} & ... & u_{M,L_M} \end{bmatrix}.$$

Here, $u_{i,j}$ denotes the j^{th} utterance of the i^{th} video and $L = [L_1, L_2, ..., L_M]$ represents the number of utterances per video in the dataset set.

4.3.1 Context-Independent Unimodal Features Extraction

Initially, the unimodal features are extracted from each utterance separately, i.e., we do not consider the contextual relation and dependency among the utterances (Table 4.1). Below, we explain the textual, audio, and visual feature extraction methods.

4.3.1.1 Textual Feature Extraction

For feature extraction from textual data, we use a convolutional neural network (CNN).

The idea behind convolution is to take the dot product of a vector of k weights, w_k, known as kernel vector, with each k-gram in the sentence $s(t)$ to obtain another sequence of features $c(t) = (c_1(t), c_2(t), \ldots, c_{\mathrm{L}}(t))$:

$$c_j = w_k^T \cdot \mathbf{x}_{i:i+k-1}.$$

We then apply a max pooling operation over the feature map and take the maximum value $\hat{c}(t) = \max\{\mathbf{c}(t)\}$ as the feature corresponding to this particular kernel vector. We use varying kernel vectors and window sizes to obtain multiple features.

The process of extracting textual features is as follows -

First, we represent each sentence as the concatenation of vectors of the constituent words. These vectors are the publicly available 300-dimensional word2vec vectors

trained on 100 billion words from Google News [1]. The convolution kernels are thus applied to these word vectors instead of individual words. Each sentence is wrapped to a window of 50 words which serves as the input to the CNN.

The CNN has two convolutional layers - the first layer having a kernel size of 3 and 4, with 50 feature maps each and a kernel size 2 with 100 feature maps for the second. The convolution layers are interleaved with pooling layers of dimension 2. We use ReLU as the activation function. The convolution of the CNN over the sentence learns abstract representations of the phrases equipped with implicit semantic information, which with each successive layer spans over increasing number of words and ultimately the entire sentence.

TABLE 4.1: Methods for extracting context independent baseline features from different modalities.

Modality	Model
Text	*text-CNN*: Deep Convolutional Neural Network with word embeddings
Video	*3d-CNN*: 3-dimensional CNNs employed on utterances of the videos
Audio	*openSMILE*: Extracts low level audio descriptors from the audio modality

4.3.1.2 Audio Feature Extraction

Audio features are extracted in 30 Hz frame-rate; we use a sliding window of 100 ms. To compute the features, we use the open-source software openSMILE [22] which automatically extracts pitch and voice intensity. Voice normalization is performed and voice intensity is thresholded to identify samples with and without voice. Z-standardization is used to perform voice normalization.

The features extracted by openSMILE consist of several low-level descriptors (LLD) and their statistical functionals. Some of the functionals are amplitude mean, arithmetic mean, root quadratic mean, etc. Taking into account all functionals of each LLD, we obtained 6373 features.

4.3.1.3 Visual Feature Extraction

We use 3D-CNN to obtain visual features from the video. We hypothesize that 3D-CNN will not only be able to learn relevant features from each frame, but will also be able to learn the changes among given number of consecutive frames.

In the past, 3D-CNN has been successfully applied to object classification on 3D data [23]. Its ability to achieve state-of-the-art results motivated us to use it.

Let $vid \in \mathbb{R}^{c \times f \times h \times w}$ be a video, where c = number of channels in an image (in our case c = 3, since we consider only RGB images), f = number of frames, h = height of

the frames, and w = width of the frames. Again, we consider the 3D convolutional filter $filt \in \mathbb{R}^{fm \times c \times fl \times fh \times fw}$, where fm = number of feature maps, c = number of channels, fd = number of frames (in other words depth of the filter), fh = height of the filter, and fw = width of the filter. Similar to 2D-CNN, $filt$ slides across video vid and generates output $convout \in \mathbb{R}^{fm \times c \times (f-fd+1) \times (h-fh+1) \times (w-fw+1)}$. Next, we apply max pooling to $convout$ to select only relevant features. The pooling will be applied only to the last three dimensions of the array $convout$.

In our experiments, we obtained best results with 32 feature maps (fm) with the filter-size of $5 \times 5 \times 5$ (or $fd \times fh \times fw$). In other words, the dimension of the filter is $32 \times 3 \times 5 \times 5 \times 5$ (or $fm \times c \times fd \times fh \times fw$). Subsequently, we apply max pooling on the output of convolution operation, with window-size being $3 \times 3 \times 3$. This is followed by a dense layer of size 300 and softmax. The activations of this dense layer are finally used as the video features for each utterance.

4.3.2 Context-Dependent Unimodal Feature Extraction

We hypothesize that, within a video, there is a high probability of utterance relatedness with respect to their sentimental and emotional clues. Since most videos tend to be about a single topic, the utterances within each video are correlated, e.g., due to the development of the speaker's idea, co-references, etc. This calls for a model which takes into account such inter-dependencies and the effect these might have on the current utterance. To capture this flow of informational triggers across utterances, we use a LSTM-based recurrent network scheme [29].

4.3.2.1 Long Short-Term Memory (LSTM)

LSTM is a kind of recurrent neural network (RNN), an extension of conventional feed-forward neural network. Specifically, LSTM cells are capable of modeling long-range dependencies, which other traditional RNNs fail to do given the vanishing gradient issue. Each LSTM cell consists of an input gate i, an output gate o, and a forget gate f, which enables it to remember the error during the error propagation. Current research [30] indicates the benefit of using such networks to incorporate contextual information in the classification process.

In our case, the LSTM network serves the purpose of context-dependent feature extraction by modeling relations among utterances. We term our architecture 'contextual LSTM'. We propose several architectural variants of it later in the chapter.

4.3.2.2 Contextual LSTM Architecture

Let unimodal features have dimension k, each utterance is thus represented by a feature vector $\boldsymbol{x}_{i,t} \in \mathbb{R}^k$, where t represents the t^{th} utterance of the video i. For a video, we collect the vectors for all the utterances in it, to get $\boldsymbol{X_i} = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, ..., \boldsymbol{x}_{i,L_i}] \in \mathbb{R}^{L_i \times k}$, where L_i represents the number of utterances in the video. This matrix X_i serves as the input to the LSTM. Figure 4.1 demonstrates the functioning of this LSTM module.

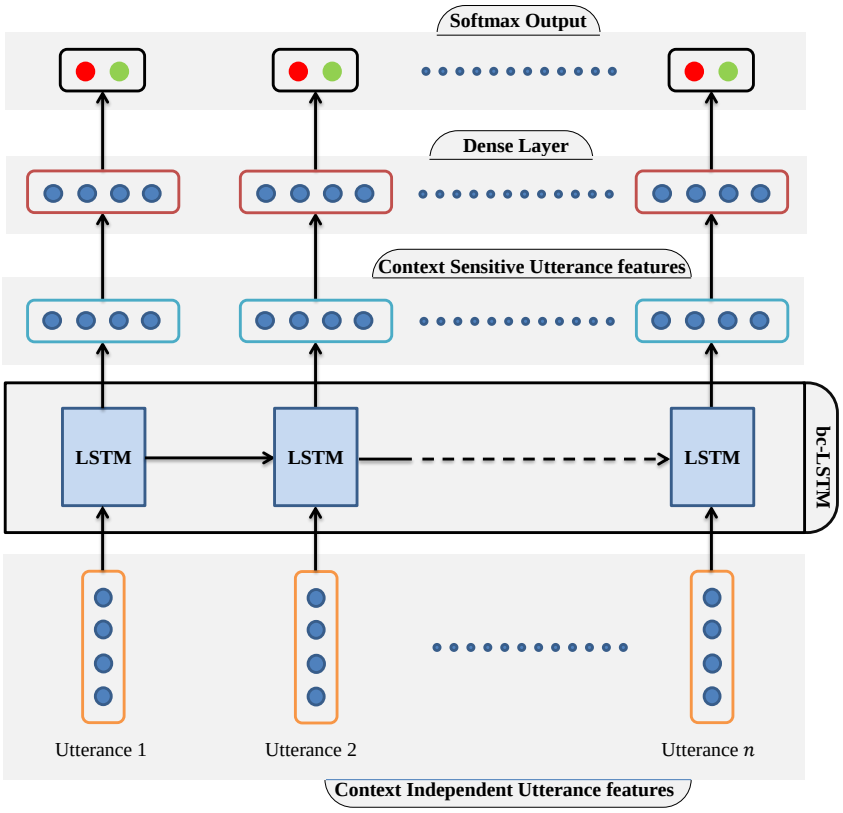


FIGURE 4.1: Contextual LSTM network: input features are passed through an unidirectional LSTM layer, followed by a dense layer and then a softmax layer. Categorical cross entropy loss is taken for training. The dense layer activations serve as the output features.

In the procedure *getLstmFeatures(X_i)* of Algorithm 3, each of these utterance $x_{i,t}$ is passed through a LSTM cell using the equations mentioned in line 32 to 37. The output of the LSTM cell $h_{i,t}$ is then fed into a dense layer and finally into a softmax layer (line 38 to 39). The activations of the dense layer $z_{i,t}$ are used as the context-dependent features of contextual LSTM.

4.3.2.3 Training

The training of the LSTM network is performed using categorical cross entropy on each utterance's softmax output per video, i.e.,

$$loss = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c}\log_2(\hat{y_{n,c}}),$$

where N = total number of utterances in a video, $y_{n,c}$ = original output of class c, and $\hat{y_{n,c}}$ = predicted output.

A dropout layer between the LSTM cell and dense layer is introduced to check overfitting. As the videos do not have same the number of utterances, padding is introduced to serve as neutral utterances. To avoid the proliferation of noise within the network, masking is done on these padded utterances to eliminate their effect in the network. Parameter tuning is done on the train set by splitting it into train and validation components with $80/20\%$ split. RMSprop has been used as the optimizer which is known to resolve Adagrad's radically diminishing learning rates [31]. After feeding the train set to the network, the test set is passed through it to generate their context-dependent features.

Different Network Architectures We consider the following variants of the contextual LSTM architecture in our experiments -

sc-LSTM This variant of the contextual LSTM architecture consists of unidirectional LSTM cells. As this is the simple variant of the contextual LSTM, we termed it as simple contextual LSTM (sc-LSTM)

h-LSTM We also test on an architecture where the dense layer after the LSTM cell is omitted. Thus, the output of the LSTM cell $h_{i,t}$ provides our context-dependent features and the softmax layer provides the classification. We call this architecture hidden LSTM (h-LSTM).

bc-LSTM Bi-directional LSTMs are two unidirectional LSTMs stacked together having opposite directions. Thus, an utterance can get information from other utterances occurring before and after itself in the video. We replaced the regular LSTM with a bi-directional LSTM and named the resulting architecture as bi-directional contextual LSTM (bc-LSTM). The training process of this architecture is similar to sc-LSTM.

uni-SVM In this setting, we first obtain the unimodal features as explained in Section 4.3.1, concatenate them and then send to a SVM for the final classification. It should be noted that using a Gated Recurrent Unit (GRU) instead of LSTM did not improve the performance.

4.3.3 Context-Dependent Multimodal Fusion

We accomplish multimodal fusion in two different ways as explained below -

4.3.3.1 Non-Hierarchical Framework

In non-hierarchical framework, we concatenate context-independent unimodal features (from Section 4.3.1) and feed that into the contextual LSTM networks, i.e., sc-LSTM, bc-LSTM, and h-LSTM.

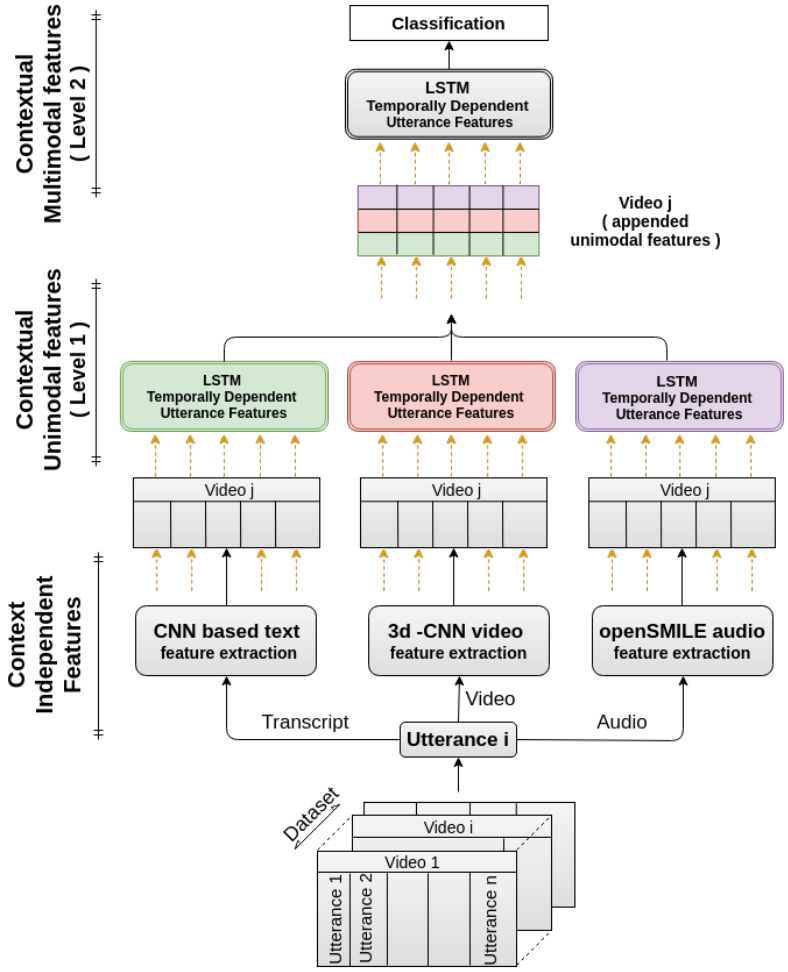


FIGURE 4.2: Hierarchical architecture for extracting context-dependent multimodal utterance features. LSTM module has been described in Figure 4.1.

4.3.3.2 Hierarchical Framework

Contextual unimodal features, taken as input, can further improve performance of the multimodal fusion framework explained in Section 4.3.3.1. To accomplish this, we propose a hierarchical deep network which comprises of two levels –

Level-1. context-independent unimodal features (from 4.3.1) are fed to the proposed LSTM network (Section 4.3.2.2) to get *context-sensitive* unimodal feature representations for each utterance. Individual LSTM networks are used for each modality.

Level-2. consists of a contextual LSTM network similar to Level-1 but independent in training and computation. Output from each LSTM network in Level-1 are concatenated and fed into this LSTM network, thus providing an inherent fusion scheme - the prime objective of this level (Fig 4.2). The performance of the second level banks on the quality of the features from the previous level, with better features aiding the fusion

Algorithm 3 Proposed Architecture

procedure TRAINARCHITECTURE(U, V)
  Train context-independent models with U
  for `i:[1,M]` **do** ▷ extract baseline features
    for `j:[1,`L_i`]` **do**
      $x_{i,j} \leftarrow TextFeatures(u_{i,j})$
      $x^{'}_{i,j} \leftarrow VideoFeatures(u_{i,j})$
      $x^{"}_{i,j} \leftarrow AudioFeatures(u_{i,j})$
  Unimodal:
  Train LSTM at Level-1 with $X, X^{'} and X^{"}$.
  for `i:[1,`M`]` **do** ▷ unimodal features
    $Z_i \leftarrow getLSTMFeatures(X_i)$
    $Z^{'}_i \leftarrow getLSTMFeatures(X^{'}_i)$
    $Z^{"}_i \leftarrow getLSTMFeatures(X^{"}_i)$
  Multimodal:
  for `i:[1,M]` **do**
    for `j:[1,`L_i`]` **do**
      if `Non-hierarchical fusion` **then**
        $x^*_{i,j} \leftarrow (x_{i,j}||x^{'}_{i,j}||x^{"}_{i,j})$ ▷ concatenation
      else
        if `Hierarchical fusion` **then**
          $x^*_{i,j} \leftarrow (z_{i,j}||z^{'}_{i,j}||z^{"}_{i,j})$ ▷ concatenation
  Train LSTM at Level-2 with X^*.
  for `i:[1,`M`]` **do** ▷ multimodal features
    $Z^*_i \leftarrow getLSTMFeatures(X^*_i)$
  testArchitecture(V)
  return Z^*

procedure TESTARCHITECTURE(V)
  Similar to training phase. V is passed through the learnt models to get the features and classification outputs. Table 4.2 shows the trainable parameters.

procedure GETLSTMFEATURES(X_i) ▷ for i^{th} video
  $Z_i \leftarrow \phi$
  for `t:[1,`L_i`]` **do** ▷ Table 4.2 provides notation
    $i_t \leftarrow \sigma(W_i \Delta x_{i,t} + P_i.h_{t-1} + b_i)$
    $\widetilde{C_t} \leftarrow tanh(W_c x_{i,t} + P_c h_{t-1} + b_c)$
    $f_t \leftarrow \sigma(W_f x_t + P_f h_{t-1} + b_f)$
    $C_t \leftarrow i_t * \widetilde{C_t} + f_t * C_{t-1}$
    $o_t \leftarrow \sigma(W_o x_t + P_o h_{t-1} + V_o C_t + b_o)$
    $h_t \leftarrow o_t * tanh(C_t)$ ▷ output of lstm cell
    $z_t \leftarrow ReLU(W_z h_t + b_z)$ ▷ dense layer
    $prediction \leftarrow softmax(W_{sft} z_t + b_{sft})$
    $Z_i \leftarrow Z_i \cup z_t$
  return Z_i

process. Algorithm 3 describes the overall computation for utterance classification. For the hierarchical framework, we train Level 1 and Level 2 successively but separately.

TABLE 4.2: Summary of notations used in Algorithm 3. Note: d: dimension of hidden unit; k: dimension of input vectors to LSTM layer; c: number of classes.

Weight		Bias	
W_i, W_f, W_c, W_o	$\in \mathbb{R}^{d \times k}$	b_i, b_f, b_c, b_o	$\in \mathbb{R}^{d}$
$P_i, P_f, P_c, P_o V_o$	$\in \mathbb{R}^{d \times d}$	b_z	$\in \mathbb{R}^{m}$
W_z	$\in \mathbb{R}^{m \times d}$	b_{sft}	$\in \mathbb{R}^{c}$
W_{sft}	$\in \mathbb{R}^{c \times m}$		

4.4 Experimental Results

4.4.1 Dataset Used

Most of the research in multimodal sentiment analysis is performed on datasets with speaker overlap in train and test splits.

Because each individual has a unique way of expressing emotions and sentiments, finding generic, person-independent features for sentimental analysis is very tricky. In real-world applications, the model should be robust to person variance but it is very difficult to come up with a generalized model from the behavior of a limited number of individuals To this end, we perform person-independent experiments to emulate unseen conditions. Our train/test splits of the datasets are completely disjoint with respect to speakers.

While testing, our models have to classify emotions and sentiments from utterances by speakers they have never seen before.

IEMOCAP: The IEMOCAP contains the acts of 10 speakers in a two way conversation segmented into utterances. The database contains the following categorical labels: anger, happiness, sadness, neutral, excitement, frustration, fear, surprise, and other, but we take only the first four so as to compare with the state of the art [15] and other authors. Videos by the first 8 speakers are considered in the train set. The train/test split details are provided in table 4.3.

MOSI: The MOSI dataset is a dataset rich in sentimental expressions where 93 persons review topics in English. It contains *positive and negative* classes as its sentiment labels. The train/validation set comprises of the first 62 individuals in the dataset.

MOUD: This dataset contains product review videos provided by around 55 persons. The reviews are in Spanish (we use Google Translate API[1] to get the english transcripts). The utterances are labeled to be either *positive, negative or neutral*. However, we drop the *neutral* label to maintain consistency with previous work. The first 59 videos are considered in the train/val set.

Table 4.3 provides information regarding train/test split of all the datasets. In these splits it is ensured that 1) *No two utterances from the train and test splits belong to the same video.* 2) *The train/test splits have no speaker overlap.* This provides the speaker-independent setting.

Table 4.3 also provides cross dataset split details where the complete datasets of MOSI and MOUD are used for training and testing respectively. The proposed model being used on reviews from different languages allows us to analyze its robustness and generalizability.

TABLE 4.3: Utterance; Person-Independent Train/Test split details of each dataset ($\approx$ 70/30 % split). Note: X$\rightarrow$Y represents train: X and test: Y; Validation sets are extracted from the shuffled train sets using 80/20 % train/val ratio.

Dataset	Train		Test	
	uttrnce	*video*	*uttrnce*	*video*
IEMOCAP	4290	120	1208	31
MOSI	1447	62	752	31
MOUD	322	59	115	20
MOSI $\rightarrow$ MOUD	2199	93	437	79

It should be noted that the datasets' individual configuration and splits are same throughout all the experiments (i.e., context-independent unimodal feature extraction, LSTM-based context-dependent unimodal and multimodal feature extraction and classification).

4.4.2 Performance of Different Models and Comparison

In this section, we present unimodal and multimodal sentiment analysis performance of different LSTM network variants as explained in Section 4.3.2.3 and comparison with the state of the art.

Hierarchical vs Non-hierarchical Fusion Framework - As expected, trained contextual unimodal features help the hierarchical fusion framework to outperform the non-hierarchical framework. Table 4.4 demonstrates this by comparing both hierarchical and non-hierarchical framework using the bc-LSTM network. Due to this fact, we provide all further analysis and results using the hierarchical framework. Non-hierarchical model outperforms the performance of the baseline *uni-SVM*. This further leads us to conclude

[1] http://translate.google.com

that it is the context-sensitive learning paradigm which plays the key role in improving performance over the baseline.

Comparison among Network Variants - It is to be noted that both sc-LSTM and bc-LSTM perform quite well on the multimodal emotion recognition and sentiment analysis datasets. Since, bc-LSTM has access to both the preceding and following information of the utterance sequence, it performs consistently better on all the datasets over sc-LSTM.

The usefulness of the dense layer in improving the performance is prominent from the experimental results as shown in Table 4.4. The performance improvement is in the range of 0.3% to 1.5% on MOSI and MOUD datasets. On the IEMOCAP dataset, the performance improvement of bc-LSTM and sc-LSTM over h-LSTM is in the range of 1% to 5%.

TABLE 4.4: Comparison of models mentioned in Section 4.3.2.3. The table reports the accuracy of classification. Note: non-hier ← Non-hierarchical bc-lstm. For remaining fusion hierarchical fusion framework is used (Section 4.3.3.2)

Modality	MOSI					MOUD					IEMOCAP				
	hierarchical (%)				non-hier (%)	hierarchical (%)				non-hier (%)	hierarchical (%)				non-hier (%)
	uni-SVM	h-LSTM	sc-LSTM	bc-LSTM		uni-SVM	h-LSTM	sc-LSTM	bc-LSTM		uni-SVM	h-LSTM	sc-LSTM	bc-LSTM	
T	75.5	77.4	77.6	**78.1**		49.5	50.1	51.3	**52.1**		65.5	68.9	71.4	**73.6**	
V	53.1	55.2	55.6	**55.8**		46.3	48.0	48.2	**48.5**		47.0	52.0	52.6	**53.2**	
A	58.5	59.6	59.9	**60.3**		51.5	56.3	57.5	**59.9**		52.9	54.4	55.2	**57.1**	
T + V	76.7	78.9	79.9	**80.2**	78.5	50.2	50.6	51.3	**52.2**	50.9	68.5	70.3	72.3	**75.4**	73.2
T + A	75.8	78.3	78.8	**79.3**	78.2	53.1	56.9	57.4	**60.4**	55.5	70.1	74.1	75.2	**75.6**	74.5
V + A	58.6	61.5	61.8	**62.1**	60.3	62.8	62.9	64.4	**65.3**	64.2	67.6	67.8	68.2	**68.9**	67.3
T + V + A	77.9	78.1	78.6	**80.3**	78.1	66.1	66.4	67.3	**68.1**	67.0	72.5	73.3	74.2	**76.1**	73.5

Comparison with the Baseline and state of the art - Every LSTM network variant has outperformed the baseline *uni-SVM* on all the datasets by the margin of 2% to 5%(see Table 4.4). These results prove our initial hypothesis that modeling the contextual dependencies among utterances, which *uni-SVM* cannot do, improves the classification. The higher performance improvement on the IEMOCAP dataset indicates the necessity of modeling long-range dependencies among the utterances as continuous emotion recognition is a multiclass sequential problem where a person doesnt frequently change emotions [32].

We have implemented and compared with the current state-of-the-art approach proposed by Poria et al. [19]. In their method, they extracted features from each modality and fed to a multiple kernel learning (MKL) classifier. However, they did not conduct the experiment in speaker-independent manner and also did not consider the contextual relation among the utterances. Experimental results in Table 4.5 shows that the proposed

TABLE 4.5: Accuracy % on textual (T), visual (V), audio (A) modality and comparison with the state of the art. For fusion, hierarchical fusion framework was used (Section 4.3.3.2)

Modality	Sentiment (%)		Emotion on IEMOCAP (%)			
	MOSI	MOUD	*angry*	*happy*	*sad*	*neutral*
T	78.12	52.17	76.07	78.97	76.23	67.44
V	55.80	48.58	53.15	58.15	55.49	51.26
A	60.31	59.99	58.37	60.45	61.35	52.31
T + V	80.22	52.23	77.24	78.99	78.35	68.15
T + A	79.33	60.39	77.15	79.10	78.10	69.14
V + A	62.17	65.36	68.21	71.97	70.35	62.37
A + V + T	**80.30**	**68.11**	**77.98**	**79.31**	**78.30**	**69.92**
State-of -the-art	73.55[1]	63.25[1]	73.10 [2]	72.40[2]	61.90[2]	58.10[2]

[1]by [19],[2]by [15]

method has outperformed Poria et al. [19] by a significant margin. For the emotion recognition task, we have compared our method with the current state of the art [15], who extracted features in a similar fashion to [19] did. However, for fusion they used SVM trees.

4.4.3 Importance of the Modalities

As expected, in all kinds of experiments, bimodal and trimodal models have outperformed unimodal models. Overall, audio modality has performed better than visual on all the datasets. On MOSI and IEMOCAP datasets, textual classifier achieves the best performance over other unimodal classifiers. On IEMOCAP dataset, the unimodal and multimodal classifiers obtained poor performance to classify *neutral* utterances. Textual modality, combined with non-textual modes boosts the performance in IEMOCAP by a large margin. However, the margin is less in the other datasets.

On the MOUD dataset, textual modality performs worse than audio modality due to the noise introduced in translating Spanish utterances to English. Using Spanish word vectors[2] in *text-CNN* results in an improvement of 10% . Nonetheless, we report results using these translated utterances as opposed to utterances trained on Spanish word vectors, in order to make fair comparison with [19].

4.4.4 Generalization of the Models

To test the generalizability of the models, we have trained our framework on complete MOSI dataset and tested on MOUD dataset (Table 4.6). The performance was poor for audio and textual modality as the MOUD dataset is in Spanish while the model is

[2]http://crscardellino.me/SBWCE

TABLE 4.6: Cross-dataset comparison (classification accuracy).

Modality	MOSI → MOUD			
	uni-SVM	h-LSTM	sc-LSTM	bc-LSTM
T	46.5%	46.5%	46.6%	**46.9%**
V	43.3%	45.5%	48.3%	**49.6%**
A	42.9%	46.0%	46.4%	**47.2%**
T + V	49.8%	49.8%	49.8%	**49.8%**
T + A	50.4%	50.9%	51.1%	**51.3%**
V + A	46.0%	47.1%	49.3%	**49.6%**
T + V + A	51.1%	52.2%	52.5%	**52.7%**

trained on MOSI dataset which is in English language. However, notably visual modality performs better than other two modalities in this experiment which signifies that in cross-lingual scenarios facial expressions carry more generalized, robust information than audio and textual modalities. We could not carry out the similar experiment for emotion recognition as no other utterance-level dataset apart from the IEMOCAP was available at the time of our experiments.

4.4.5 Qualitative Analysis

In some cases the predictions of the proposed method are wrong given the difficulty in recognizing the face and noisy audio signal in the utterances. Also, cases where the sentiment is very weak and non contextual, the proposed approach shows some bias towards its surrounding utterances which further leads to wrong predictions. This can be solved by developing a context aware attention mechanism. In order to have a better understanding on roles of modalities for overall classification, we also have done some qualitative analysis. For example, this utterance - *"who doesn't have any presence or greatness at all."*, was classified as positive by the audio classifier ("doesn't" was spoken normally by the speaker, but "presence and greatness at all" was spoken with enthusiasm). However, textual modality caught the negation induced by "doesn't" and classified correctly. In another utterance "amazing special effects" as there was no jest of enthusiasm in speaker's voice and face audio-visual classifier failed to identify the positivity of this utterance. On the other textual classifier correctly detected the polarity as positive.

On the other hand, textual classifier classified this sentence - "that like to see comic book characters treated responsibly" as positive, possibly because of the presence of positive phrases such as "like to see", "responsibly". However, the high pitch of anger in the person's voice and the frowning face helps identify this to be a negative utterance.

4.5 Conclusions

Contextual relationship among the utterances is mostly ignored in the literature. In this chapter, we developed a LSTM-based network to extract contextual features from the

utterances of a video for multimodal sentiment analysis. The proposed method has outperformed the state of the art and showed significant performance improvement over the baseline. As a part of the future work, we plan to propose LSTM attention model to determine importance of the utterances and contribution of modalities in the sentiment classification.

Chapter 5

Contextual Multimodal Sentiment Analysis with Attention Mechanism

5.1 Introduction

With increasing and cheaper accessibility of smartphones and computers, resulting rapid growth of social media, people now tend to share their opinions about various topics on the social media platforms. Every day, a huge amount of these opinions are being shared in the form of videos on platforms like Facebook, YouTube, Vimeo. A video usually consists of three different information channels, also called modalities: visual (represents facial expressions and body gestures), audio (represents speech), and textual (represents spoken words). The need of mining opinions from such a large quantity of videos calls for multimodal sentiment analysis, a popular field of research [19].

An utterance is a segment of speech bounded by breaths or pauses. A review video often contains multiple utterances. The goal of utterance-level sentiment analysis is to label each utterance by its sentiment label. Utterance-level sentiment analysis facilitates us to understand the sentiment dynamics of the reviewer on multiple aspects of the review. Recently a number of approaches have been proposed in the field of multimodal sentiment analysis [14], [15], [19], [25], [26]. These approaches consider each utterances as independent entities, and thus ignore the relationship and dependencies among the utterances in a video. However, in a video, the utterances maintain a sequence and can be highly correlated due to the development of speaker's idea, co-reference etc. In particular, in classification of an utterance in a video, other utterances can provide useful contextual information. For example, in one video, the reviewer of the movie X-MEN Origin says: *"Now the title of the movie basically says it all"* which can carry both positive and negative sense. This ambiguity is resolved by other sentences in the video, like: *"That like to see comic book characters treated Responsibly!"* and *"Nothing Special"*, which indicate the disappointment in the reviewer towards the film and the target utterance. However, modeling such contextual relationship may not be enough. Identifying relevant and important information from the pool of utterances is necessary in order to make a model more robust and accurate. Thanks to attention mechanism, we propose an attention based Long Term Memory Network (LSTM) network which not only models the contextual relationship among the utterances, but also prioritizes more relevant utterances for classifying the target utterance.

It is already established in the literature [19], [26] that multimodal analysis outperforms unimodal analysis. However, the contribution of individual modality in the multimodal fusion has not been a major focus of the recent works. It is important to assess how a multimodal model dynamically determines the contribution of each modality in the decision making. Such a feat is difficult to achieve using simple concatenation based fusion methods as proposed in the literature [19]. In this work, we propose attention based fusion (also called AT-Fusion) network which dynamically determines and amplifies the modalities with major contribution for the sentiment classification. It is done by applying attention mechanism. Experimental results show that the proposed framework outperforms the state of the art on benchmark datasets by $6 - 8\%$. Below, we describe the major contributions of this chapter:

- We propose a deep attention based LSTM (CAT-LSTM) network to model the contextual relationship among utterances and prioritize the important contextual information for classification.
- We also introduce an attention based fusion mechanism called AT-Fusion, which amplifies the higher quality and informative modalities during fusion in multimodal classification.
- The experimental results indicate that our proposed method is able to over the performance of the state of the art and other baseline models.

This chapter is organized as follows: Section 5.2 provides a brief literature review on multimodal sentiment analysis; Section 5.3 describes the proposed method in detail; Experimental results and discussion are shown in Section 5.4; finally, Section 4.5 concludes the chapter.

5.2 Related Work

Recent works in text based sentiment analysis focus on developing and employing deep learning techniques such as Convolutional Neural Network (CNN) [33], Recursive Neural Tensor Network (RNTN) [7], Long Short Term Memory [34]. So far, these have shown significant improvement over the knowledge based approaches in sentiment analysis [5].

Since, sentiment analysis and emotion recognition are often treated as closely related problems, in this section we cover the existing works in both of these fields. Earlier works on facial expressions and emotion recognition were mainly attributed towards identifying facial characteristic points (FCP) and use them as features for emotion recognition [8]. To extract the acoustic features from speech data for emotion recognition and other tasks, Eyben et al. [35] developed a feature extraction toolkit called openSMILE. With the advancement of deep learning, Deep Belief Network (DBN) [36], CNN [37], LSTM are being used to automatically extract both audio and visual features for emotion recognition and sentiment analysis.

Two of the early works in audio-visual emotion recognition have been conducted by Chen et al. [11] and De Silva et al. [10]. Both of these approaches show that bimodal classifier performs better than unimodal in emotion recognition. More recently the approaches by Kessous et al. [12] and Schuller et al. [13] have fused audio and visual information in both feature and decision level. They have confirmed that audio modality is a better performer than visual modality.

Although, there are many research works conducted on audio-visual emotion recognition, only a few number of those works have been carried out using textual with visual and audio modalities. Poria et al. [19], [37] used CNN to extract features from the modalities and then employed Multiple Kernel Learning (MKL) for sentiment analysis and emotion recognition. Wollmer et al. [14] and Rozgic et al. [15], [28] fused audio, visual, and textual information in feature level. All of these approaches confirm that multimodal classifiers perform better than unimodal classifiers. As for fusing audio and textual clues, Metallinou et al. [16] and Eyben et al. [17] fused audio and textual modalities in feature level. On the other hand, Wu et al. [18] fused audio and textual cues at decision level.

5.3 Our Method

In the following subsections, we first discuss the problem definition, followed by the explanation of the proposed approach.

5.3.1 Problem Definition

Let us consider a video $V_j = [u_{j,1}\ u_{j,2}\ u_{j,3}, ..., u_{j,i}...u_{j,L_j}]$ where $u_{j,i}$ is the i^{th} utterance in video v_j and L_j is the number utterances in the video. The goal of this approach is to label each utterance $u_{j,i}$ by the sentiment expressed by the speaker of the utterance. We claim that in order to classify utterance $u_{j,i}$, the other utterances in the video, i.e., $[u_{j,k} \mid \forall k \leq L_j, k \neq i]$, serve as its context and can provide key information in the classification.

5.3.2 Overview of the Approach

The overview of the proposed approach is as follows:

1. **Unimodal feature extraction -** We first extract the utterance-level unimodal features from the respective unimodal classifiers. This phase does not consider the contextual relationship among the utterances.
2. **AT-Fusion – Multimodal fusion using attention mechanism -** In this step, the utterance-level unimodal features, extracted in Step 1, are fused using an attention network, called AT-Fusion, and the resulting output is used in the next step for sentiment classification.

3. **CAT-LSTM – Attention based LSTM model for sentiment classification -** CAT-LSTM is an attention based LSTM network which accepts the features (output of Step 2) of a sequence of utterances per video and generates a new representation of those utterances based on the surrounding utterances.

5.3.3 Unimodal Feature Extraction

In this step we extract unimodal features using dedicated unimodal feature extractors. The utterances are treated independently in this process.

5.3.3.1 Textual Features Extraction

We use a Convolutional Neural Network (CNN) for textual feature extraction. The CNN takes utterances represented as a matrix of Google `word2vec` [1] vectors; word2vec covers 87% of the vocabulary of MOSI dataset, the missing vectors are initialized randomly. The convolution filters are then applied to this matrix of word vectors.

The CNN has two convolutional layers; the first layer has two kernels of size 3 and 4, with 50 feature maps each and the second layer has a kernel of size 2 with 100 feature maps. The convolution layers are interleaved with max-pooling layers of window 2×2. This is followed by a fully connected layer of size 500 and softmax output. We use ReLU as the activation function. The activation values of the fully-connected layer are taken as the features of utterances for text modality.

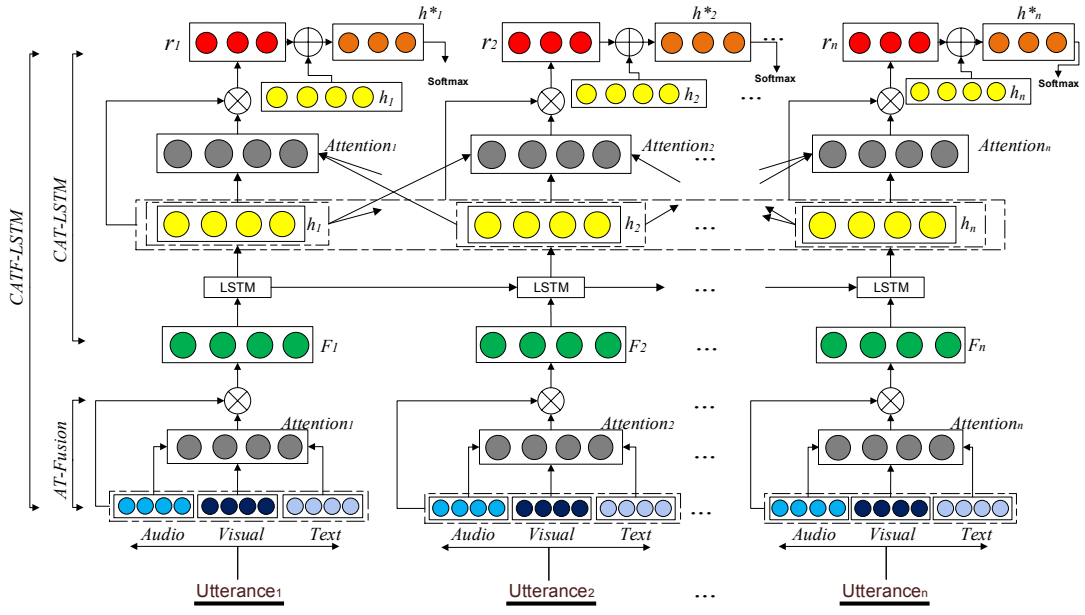


FIGURE 5.1: CATF-LSTM model takes input from multiple modalities;fuse them using AT-Fusion; send the output to CAT-LSTM for multimodal sentiment classification.

5.3.3.2 Audio Feature Extraction

Audio features are extracted with 30 Hz frame-rate and a sliding window of 100 ms using openSMILE toolkit. In order to identify samples with and without voice, voice normalization is performed using Z-standardization technique. The features extracted by openSMILE consist of several low-level descriptors (LLD) e.g., MFCC, voice intensity, pitch, and their statisticals, e.g., mean, root quadratic mean.

5.3.3.3 Visual Feature Extraction

There are various choices of deep networks specialized for image/video classification, e.g., cascaded CNN layers, recurrent networks such as RNN, LSTM, GRU. We chose 3D-CNN due to its proven ability to learn image representations (like 2D-CNN), along with the changes among the sequence of images (frames) in a video [23], [38].

Let $V \in \mathbb{R}^{c \times f \times h \times w}$ represents an utterance video, where c = number of channels in an image (in our experiments c = 3, since the constituent images are RGB), f = number of frames, h = height of each frame,and w = width of each frame. We apply 3D convolutional filter F to video V, where $F \in \mathbb{R}^{f_m \times c \times f_d \times f_h \times f_w}$, f_m = number of feature maps, c = number of channels, f_d = number of frames, f_h = height of the filter, and f_w = width of the filter (we chose $F \in \mathbb{R}^{32 \times 3 \times 5 \times 5 \times 5}$). Following the philosophy of 2D-CNN, 3D-CNN slides filter F across video V and produces output $cvout \in \mathbb{R}^{fm \times c \times (f - f_d + 1) \times (h - f_h + 1) \times (w - f_w + 1)}$. To discard irrelevant features we apply max-pooling of window $3 \times 3 \times 3$ on $cvout$. Output of pooling layer is fed to fully-connected layer of size 300, followed by a softmax layer for classification. The activations of the fully-connected layer is used as the features of video V.

5.3.4 AT-Fusion: Attention Based Network for Multimodal Fusion

Attention mechanism was first introduced in image classification [39] to focus on the most important parts of an object relevant to the classification, improving the performance of the deep neural networks. Attention mechanism has been successfully employed in NLP tasks, such as machine translation [40], sentiment analysis [41], summarization [42]. Since, not all modalities are equally relevant in classification of sentiment, we introduce an attention network, named as AT-Fusion, which takes input from audio, visual, and textual modalities and results an attention score for each modality.

We equalize the dimensions of the feature vectors of all three modalities prior to feeding them into attention network. This is done using a fully-connected layer of size d. Let $B = [B_a, B_v, B_t]$ be the feature set after dimensionality equalization to size d, where B_a = acoustic features, B_v = visual features, and B_t = textual features; following $B \in \mathbb{R}^{d \times 3}$. The attention weight vector α_{fuse} and the fused multimodal feature vector F

are computed as follows :

$$P_F = tanh(W_F.B) \tag{5.1}$$
$$\alpha_{fuse} = softmax(w_F^T.P_F) \tag{5.2}$$
$$F = B.\alpha_{fuse}^T \tag{5.3}$$

Here, $W_F \in \mathbb{R}^{d\times d}$, $w_F \in \mathbb{R}^d$, $\alpha_{fuse}^T \in \mathbb{R}^3$, and $F \in \mathbb{R}^d$. We then feed the output F to the CAT-LSTM (Section 5.3.5.1, Figure 5.1) for final multimodal sentiment classification of the utterance.

5.3.5 Classifier: Context-Dependent Sentiment Classification

A speaker usually tries to gradually develop his/her idea and opinion about a product in the review, which makes the utterances in a video sequential, temporally and contextually dependent. This phenomena motivates us to model inter-utterance relationship. To accomplish so, we use an LSTM layer, in combination with attention mechanism to amplify the important contextual evidences for sentiment classification of target utterance.

5.3.5.1 Our CAT-LSTM

LSTM [4] is a specialized Recurrent Neural Network (RNN), which models long range dependencies in a sequence. Specifically, LSTM solves the vanishing gradient problem of conventional RNN while modeling long range dependencies. Current research [30] indicates the benefit of using such networks to incorporate contextual information in classification process.

Let, $x \in \mathbb{R}^{d\times M}$ be input to the LSTM network, where M is the number of utterances in a video. The matrix x can be represented as $x = [x_1, x_2, ..., x_t, ...x_M]$, where $x_t \in \mathbb{R}^d$ for $t = 0$ to M.

Each cell in LSTM can be computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \tag{5.4}$$
$$f_t = \sigma(W_f.X + b_f) \tag{5.5}$$
$$i_t = \sigma(W_i.X + b_i) \tag{5.6}$$
$$o_t = \sigma(W_o.X + b_o) \tag{5.7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c.X + b_c) \tag{5.8}$$
$$h_t = o_t \odot tanh(c_t) \tag{5.9}$$

where $W_i, W_f, W_o \in \mathbb{R}^{d\times 2d}$ and $b_i, b_f, b_o \in \mathbb{R}^d$ are parameters to be learned during the training. σ is the sigmoid function and $\odot$ is element-wise multiplication. The output of this LSTM layer is represented as $H \in \mathbb{R}^{d\times M}$, where $H = [h_1, h_2, ..., h_t, ..., h_M$] and $h_i \in \mathbb{R}^d$.

We feed the sequence of M utterance-level features (fused features F, obtained in equation (5.3) or unimodal features) to LSTM and obtain contextually aware utterance representations H.

Attention Network All surrounding utterances are not equally relevant in sentiment classification of the target utterance. In order to amplify the contribution of context-rich utterances, we use an attention network.

Let A_t denote the t^{th} Attention network for utterance represented by h_t. The attention mechanism of A_t produces an attention vector α_t and a weighted hidden representation r_t as follows:

$$P_t = tanh(W_h[t].H) \tag{5.10}$$

$$\alpha_t = softmax(w[t]^T.P_t) \tag{5.11}$$

$$r_t = H.\alpha_t^T \tag{5.12}$$

where, $P_t \in \mathbb{R}^{d\times M}, \alpha_t \in \mathbb{R}^M, r_t \in \mathbb{R}^d$. And, $W_h \in \mathbb{R}^{M\times d\times d}, w \in \mathbb{R}^{M\times d}$ are projection parameters with $W_h[t]$ and $w[t]$ being used by the t^{th} attention model. Finally, the LSTM representation for t^{th} utterance is modified as:

$$h_t^* = tanh(W_p[t].r_t + W_x[t].h_t) \tag{5.13}$$

Similar to the results obtained by Rocktäschel et al. [43], addition of the term $W_x[t].h_t$ to $W_p[t].r_t$ gives better result in the experiments carried out. Here, $h_t^* \in \mathbb{R}^d$ and $W_p, W_x \in \mathbb{R}^{M\times d\times d}$ are weights to be learned while training.

In some experiments (e.g. Section 5.3.6.2) we use the output h_t^* as contextual features for further processing.

Classification Finally each modified LSTM cell output h_t^* is sent into a softmax layer for sentiment classification.

$$Z_t = softmax((h_t^*)^T.W_{soft}[t] + b_{soft}[t]) \tag{5.14}$$

$$\hat{y}_t = \arg\max_j(Z_t[j]), \quad \forall j \in class \tag{5.15}$$

where, $Z_t \in \mathbb{R}^{ydim}, W_{soft} \in \mathbb{R}^{M\times d\times ydim}, b_{soft} \in \mathbb{R}^{M\times ydim}, ydim$ = number of classes, and $\hat{y}_t$ is the predicted class.

5.3.6 Training

5.3.6.1 Unimodal Classification

In our work, we perform classification on two types of data – unimodal and multimodal. To classify the unimodal input, the extracted unimodal features (Section 5.3.3) are sent to the CAT-LSTM network as inputs.

5.3.6.2 Multimodal Classification

For multimodal classification, the extracted unimodal features are first fed to the AT-Fusion to produce fused multimodal features. Then these features are fed to the CAT-LSTM network for the sentiment classification. We call this multimodal sentiment classification model as Contextual Attentive Fusion LSTM, i.e., CATF-LSTM. The CATF-LSTM is shown in Figure 5.1 Multimodal classification be accomplished using two different frameworks:

Single-Level Framework In single-level framework, we fuse context-independent unimodal features as explained in Section 5.3.4 and feed those to CATF-LSTM network for multimodal fusion and classification.

Multi Level Framework Contextual unimodal features can further improve performance of the multimodal fusion framework explained in Section 5.3.6.2. In this fusion scheme, we first send context-independent unimodal features extracted from every modality to CAT-LSTM network. The contextual features yielded from the CAT-LSTM network are then fed to the CATF-LSTM network for the fusion and final classification.

Both the unimodal and multimodal classifiers are trained in an end-to-end manner using back propagation, with objective function being log-loss:

$$loss = -\sum_{i}\sum_{j} log(Z_t[y_i^j]) + \lambda||\theta||^2 \tag{5.16}$$

where, y = target class, Z_t = predicted distribution of j^{th} utterance from video V_i s.t. $i \in [0, N]$ and $j \in [0, L_i]$. λ is the L_2 regularization term and θ is the parameter set $\theta = \{W_i, b_i, W_f, b_f, W_o, b_o, W_F, w_F, W_h, w, W_p, W_x, W_{soft}, b_{soft}\}$.

In our experiments, we pad videos with dummy utterances to enable batch processing. Hence, we also use bit-masks to mitigate proliferation of noise in the network. The network is typically trained for $500 - 700$ epochs with an early-stopping patience of 20 epochs. As optimizer, we use AdaGrad [31] which is known to have improved robustness over SGD, given its ability to adapt the learning rate based on the parameters.

5.4 Experimental Results

In this section we present the experimental results on different network variants in contrast with various baselines.

5.4.1 Dataset Used

Since each individual has an unique way of expressing sentiments, finding generic and person-independent features for sentimental analysis is very tricky. In real-world applications, the model should be robust to person variance. However, it is very difficult to construct a generalized model from the behavior of a limited number of individuals.

To this end, we perform person-independent experiments to emulate unseen conditions. Our train/test splits of the dataset are completely disjoint with respect to speakers.

MOSI Dataset Zadeh et al. [25] constructed a multimodal sentiment analysis dataset called Multimodal Opinion-Level Sentiment Intensity (MOSI), consisting 2199 opinionated utterances, 93 videos by 89 speakers. The videos address a large array of topics, such as movies, books, and products. Videos were crawled from YouTube and segmented into utterances. Each of the 2199 utterances were labeled with its sentiment label, i.e. positive and negative. The train set comprises of the first 62 individuals in the dataset. So, the test set comprises of 31 videos by 27 speakers. In particular, we use 1447 utterances in the training and 752 utterances to test the models.

5.4.2 Different Models and Network Architectures

We have carried out experiments with both unidirectional and bi-directional LSTM with the later giving 0.3-0.7% better performance in all kinds of experiments. As this is an expected and non-critical outcome, we present all the results below using bi-directional LSTM variant. Additionally, we consider the following models in our experiments:

Poria et al. (2015) : We have implemented and compared our method with the current state of the art approach, proposed by Poria et al. [19]. In their approach, they extracted visual features using CLM-Z, audio features using openSMILE, and textual features using CNN. Multiple Kernel Learning (MKL) was then applied on the features obtained from the concatenation of the unimodal features. However, they did not conduct the speaker independent experiments.

Poria et al. (2016) : Extended approach over [19] which introduces a CNN-RNN feature extractor to extract visual features. We reimplemented their approach in our experiments.

Unimodal-SVM : We extract unimodal features (Section 5.3.3) and concatenate them to produce multimodal features. Followed by SVM being applied on this feature vector for the final sentiment classification.

Simple-LSTM : In this configuration, the extracted unimodal and multimodal features of the utterances are fed to an LSTM. In particular, this LSTM has no attention mechanism.

CAT-LSTM : This is the simple contextual attention based LSTM framework as described in Section 5.3.5.1. For multimodal setting, it accepts input generated by appending unimodal features.

CATF-LSTM : This model is used for multimodal classification. As explained in Section 5.3.6, it consists of AT-Fusion and CAT-LSTM, where the output of AT-Fusion is fed to CAT-LSTM.

ATS-Fusion : In this variant, instead of feeding the output of AT-Fusion to the cells of CAT-LSTM, we feed to softmax classifiers. The utterances are treated independently in this case.

Poria et al. (2015) + Our best model : In order to perform a fair comparison with Poria et al. (2015), we feed the features extracted by their method to our best performing model.

Poria et al. (2016) + Our best model : This model is similar to the model *Poria et al. (2015) + Our best model*, except it uses the features extraction process from Poria et al. (2016).

5.4.3 Single-Level vs Multilevel Framework :

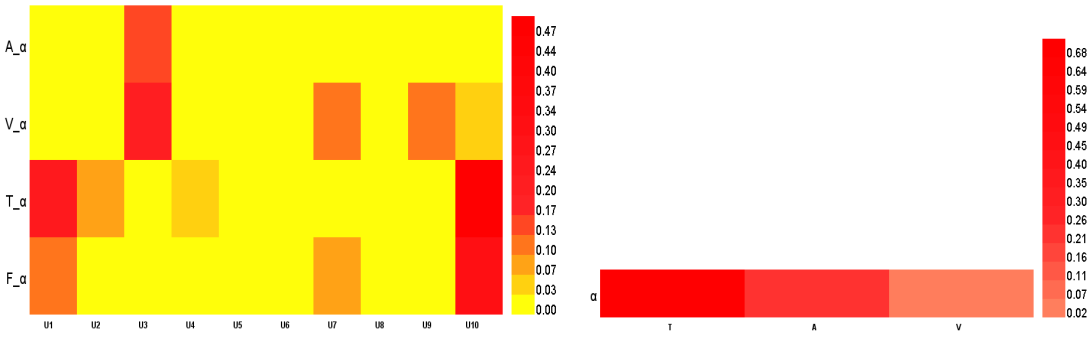
Multilevel framework outperforms single-level framework in our experiments given the presence of contextual unimodal features (see Table 5.2). Hence, for brevity, apart from Table 5.2, we present only the results of multilevel framework.

5.4.4 Performance of AT-Fusion

AT-Fusion employs attention mechanism to fuse multiple modalities. In order to assess the effectiveness of AT-Fusion, we compare it with a simple fusion technique where the feature vectors from different modalities are appended and fed to the sentiment classifier, i.e. CAT-LSTM. Table 5.2 presents the performance of CATF-LSTM: which utilizes AT-Fusion for feature fusion followed by CAT-LSTM for sentiment classification. Given AT-Fusion's ability to amplify the contribution of the important modalities during fusion, it unsurprisingly outperforms the simple fusion method. It should be noted that AT-Fusion can be integrated with the other network variants, i.e. Simple-LSTM (Table 5.3). Table 5.3 also shows that AT-Fusion with softmax output i.e., ATS-Fusion which outperforms the unimodal-SVM, thanks to the superiority of the AT-Fusion over simple feature append based fusion.

5.4.5 Comparison of the Models

Comparison with the state of the art : As shown in the Table 5.1, the proposed approach has outperformed the state of the art [19], [37] by 4.5%-6%. We use the same set of textual and audio features used in [19], [37]. Notably, apart from using a different fusion mechanism, our method also uses a different visual feature extraction method. On the MOSI dataset, the proposed visual feature extraction method has outperformed the CLM-Z (used in [19]) and CNN-RNN (proposed by [37]). When we employ our best classifier, i.e. CATF-LSTM, on the features extracted by [19] and [37], performance of those methods have improved. Using CATF-LSTM, we obtained better results than both of the state of the art results for audio-visual, visual-textual. According to [19], [37] trimodal classifier outperforms all unimodal and bimodal classifiers. Hence, we compare our proposed method with those works in the trimodal experiment. From these experimental results (Table 5.1), it is evident that the proposed contextual attention based LSTM network and fusion methodology are the key to outperform the state of the art.



(A) The visualization of the attention scores of unimodal CAT-LSTM and trimodal CATF-LSTM.

(B) Visualization of the attention weights of the AT-Fusion network in trimodal CATF-LSTM.

FIGURE 5.2: Target utterance for classification - U4 : You never know whats gonna happen. The input utterances are - U1 : This is the most engaging endeavor yet. U2 : And he puts them in very strange and unusual characters. U3 : Umm what really need about this movie is the chemical brothers did the soundtracks whats pulse pounding the entire way through. U4 : You never know whats gonna happen. U5 : Theres all these colorful characters. U6 : Now it isn't a great fantastic tell everybody about that kind of movie. U7 : But I think its one of those movies thats so unique. U8 : Its colourful. U9 : Its in you face. U10 : And something that I can't find anything else to compare it to.

TABLE 5.1: Comparison of state-of-the-art on multimodal classification with our network: CATF-LSTM. Metric used: macro-fscore. A=Audio; V=Visual; T=Textual.

Models	A+V+T
Poria et al. (2015)	73.55%
Poria et al. (2016)	75.13%
Poria et al. (2015)+ CATF-LSTM	79.40%
Poria et al. (2016) + CATF-LSTM	80.25%

unimodal-SVM : Our unimodal-SVM model yields comparable performance with the state of the art. However, simple-LSTM outperforms unimodal-SVM in all the experiments (Table 5.1) as the latter is incapable of grasping the context information while classifying an utterance.

CAT-LSTM vs Simple-LSTM : From Table 5.3, we can see that CAT-LSTM outperforms Simple-LSTM by 0.6-1.1% in unimodal experiments; 0.2-1% in bimodal experiments, and 0.9% in trimodal experiment. This again confirms that even though both of the networks have access to contextual information, CAT-LSTM outperforms Simple-LSTM because of its attention capability to capture the important contexts. As expected, CATF-LSTM further improves the performance of CAT-LSTM.

TABLE 5.2: Comparison between *single-level* and *multi level* fusion mentioned in Section 5.3.6.2 using CAT-LSTM network. Feat Append=Unimodal features are appended and sent to CAT-LSTM. AT-Fusion is used with CAT-LSTM network. The table reports the macro-fscore of classification. A=Audio; V=Visual; T=Textual.

Modality	Single-Level		Multi Level	
	Feat Append	AT-Fusion	Feat Append	AT-Fusion
A+V	61.0	**61.6**	62.4	**62.9**
A+T	78.5	**79.2**	79.5	**80.1**
V+T	**78.3**	78.3	79.6	**79.9**
A+V+T	78.9	**79.3**	81.0	**81.3**

TABLE 5.3: Comparison of models mentioned in Section 5.4.2. The table reports the macro-fscore of classification. Note: feat-append=fusion by concatenating unimodal features. Multilevel framework is employed (See Section 5.3.6.2). A=Audio; V=Visual; T=Textual.

Modalities	Sentiment, on MOSI					
	Uni-SVM	*Simple-LSTM*		*CAT-LSTM*		
	feat-app	feat-app	AT-Fusion	feat-app	AT-Fusion	ATS-Fusion
A	58.1	59.5	-	60.1	-	-
V	53.4	54.9	-	55.5	-	-
T	75.5	77.2	-	79.1	-	-
A + V	58.6	61.4	61.8	62.4	62.9	59.1
A + T	75.8	78.5	79.1	79.5	80.1	76.3
V + T	76.7	78.7	79.1	79.6	79.9	77.5
A + V + T	**77.9**	**80.1**	**80.6**	**81.0**	**81.3**	78.3

5.4.6 Importance of the Modalities :

As expected, bimodal classifiers dominate unimodal classifiers and trimodal classifiers perform the best among all. Across modalities, textual modality performs better than the other two, thus indicating the need for better feature extraction for audio and video modalities.

5.4.7 Qualitative Analysis and Case Studies

In this section we provide analysis and interesting observations on the learned attention parameters for both contextual attention (CAT-Fusion) and attention fusion (AT-Fusion). Below we enlist some of these observations:

The need for considering context dependency (see Section 5.1) is primal for utterance classification. The utterance: *Whoever wrote this isn't the writer definitely.* has the sentiment expressed implicitly and hence baseline unimodal-SVM and state of the art

fail to classify it correctly [1]. Information from neighboring utterances, like: *1) And the dialogue threw me off*; *2) The whole movie had a really dry dialogue* etc. indicate the negative context for the utterance. Such contextual relationships are prevalent throughout the dataset.

Considering the movie review utterance: *You never know whats gonna happen.*, the sentence doesn't provide explicit sentiment cues. In such cases, our context attention network attends to relevant utterances throughout the video to find contextual dependencies. Figure 5.2a shows the attention weights across the video for this mentioned utterance. While audio and visual provide decent attention vectors, text modality provides improved attention. It can be clearly seen that utterances like U_{10}, U_1 (Figure 5.2a) are the most relevant ones, which, multimodal attention has been able to capture, thus proving its effectiveness. Interestingly, in this case the most important utterance U_{10} is located far from the target utterance U_4, proving the effectiveness of LSTM in modeling long distance sequence. Figure 5.2b shows the contribution of each modality for the multimodal classification. Rightly, text has been given the highest weight by the attention fusion network, followed by audio and visual.

Although the context dependency among utterances can be modeled in a simple LSTM network, there are often cases where utterances with complementary contexts are sparsely distributed across the video. In such situations, neighboring irrelevant utterances may provide negative bias for the utterance to be classified. Our model, CAT-LSTM provides an attention framework which focuses only on the relevant utterances throughout the video, thus bypassing the unrelated ones. For an example, in one of the videos, the first utterance: *I am gonna give the reasons why I like him*, has its answers from the 7^{th} utterance onwards, with the intermediate utterances being irrelevant. In such situations, CAT-LSTM performs better than simple-LSTM model.

The effectiveness of the attention based fusion (AT-Fusion) network can also be seen in multiple cases. In one such instance, when a person while reviewing a movie, speaks the utterance: *Sigh it probably would have helped if I went with someone*, loud background noises affect the quality of the audio modality. In simple feature append based fusion models e.g., unimodal-SVM, this utterance is misclassified given the high noise presents in the fusion caused by the audio modality. However, the multimodal attention based fusion network (CATF-LSTM) correctly attends the video and text modality giving negligible attention on audio modality. This trend is also observed in many other cases.

We finally observe that in some cases textual CAT-LSTM classifier performs better than trimodal CATF-classifier because of the presence of noise in the audio modality and specially when the speaker does not look directly at the camera while speaking.

5.5 Conclusions

In multimodal sentiment analysis literature, utterances in a video are mostly considered independently. We discard this oversimplified assumption of utterance dependence by

[1] RNTN classifies it as neutral. It can be seen here http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

modeling relevant contextual information obtained from the other utterances in a video while classifying one target utterance. The proposed framework outperforms the state of the art and strong baselines.

Chapter 6

Personality Detection from Text using CNN

6.1 Introduction

Personality is a combination of behavior, emotion, motivation, and thought pattern characteristic of an individual [44]. Our personality has great impact on our lives. Our life choices, well-being, health, and numerous other preferences are affected by our personality.

Automatic detection of personality traits of a person has many important practical applications. In recommender systems, the products and services recommended to a person should be those that have been positively evaluated by other users with a similar personality type. In mental health diagnosis, certain diagnoses correlate with certain personality traits. In forensics, knowing personality traits helps reducing the circle of suspects. In human resources management, personality traits affect one's suitability for certain jobs, etc.

Personality is typically formally described in terms of the Big Five [45] personality traits, which are five binary (*yes* / *no*) values:

- Extroversion (EXT): Is the person outgoing, talkative, energetic vs. reserved, solitary?
- Neuroticism (NEU): Is the person sensitive, nervous vs. secure, confident?
- Agreeableness (AGR): Is the person trustworthy, straightforward, generous, modest vs. unreliable, complicated, meager, boastful?
- Conscientiousness (CON): Is the person efficient, organized vs. sloppy, careless?
- Openness (OPN): Is the person inventive, curious vs. dogmatic, cautious?

Texts often reflect various aspects of the author's personality. In this chapter, we present a method to extract personality traits from stream-of-consciousness essays (we experimented with Pennebaker and King's essay dataset [1]) using Convolutional Neural Network (CNN). We trained five different networks, all having the same architecture,

[1] http://mypersonality.org/wiki/doku.php?id=wcpr13

for the five personality traits. Each network was a binary classifier that predicted the corresponding trait to be positive or negative.

For this, we developed a novel document representation technique based on a CNN features extractor. Namely, we fed the sentences of the essays to convolution filters to obtain sentence representation in the form of n-gram feature vectors. Each individual essay was represented by aggregating the representations of its sentences. We concatenated the obtained vectors with the Mairesse features [46], which were extracted from the texts directly at the pre-processing stage; this improved performance of the method. Discarding emotionally neutral input sentences from the essays further improved the results.

For final classification, we fed this document representation into a fully-connected neural network with one hidden layer. The obtained results outperformed the current state of the art.

The chapter is organized as follows. In Section 6.2, we briefly discuss the previous work in the field. In Section 6.3, we give an overview of our method. In Section 6.3.1, we describe our network architecture. In Section 6.4, we discuss the experimental settings and obtained results. Finally, Section 6.5 concludes the chapter.

6.2 Previous Work

The Big Five, also known as Five Factor Model (FFM), is the most widely accepted model of personality. Initially, it was developed by several independent groups of researchers. However, it was advanced by Ernest Tupes and Raymond Christal [47]; J.M. Digman [45] made further advancements, and later perfected by Lewis Goldberg [48].

Some of the earlier work on automated personality detection from plain text was done by Pennebaker and King [49], who compiled the essay dataset used in our experiments [1]. For this, they collected consciousness-stream essays, written by volunteers in controlled environment, and then asked the authors of the essays to define their own Big Five personality traits. They used Liguistic Inquiry and Word Count (LIWC) features [50] to determine correlation between the essay and personality.

Mairesse et al. [46] used, in addition to LIWC, other features, such as imageability, to improve the performance. Mohammad et al. [51] performed a thorough study on this essays dataset, as well as the MyPersonality Facebook status dataset, by applying different combination of feature sets to outperform Mairesse's results, which they called the *Mairesse baseline*.

Recently, Liu et al. [52] developed a language-independent and compositional model for personality trait recognition for short tweets.

On the other hand, deep convolutional networks have been successfully used for related tasks such as sentiment analysis [53], opinion mining [54], and multimodal emotion recognition [37].

6.3 Our Method

Our method includes input data pre-processing and filtering, feature extraction, and classification. We use two types of features: (a) a fixed number of document-level stylistic features, and (b) per-word semantic features, which form a variable-length sentence representation, which in turn forms a variable-length part of document representation.

Pre-processing Pre-processing includes sentence splitting as well as data cleaning and unification, such as reduction to lowercase.

Document-level feature extraction As the document-level stylistic features, in our experiments we used the Mairesse baseline feature set, which includes such global features as the word count and average sentence length.

Filtering Some sentences in an essay may not carry any personality clues. There are two reasons to ignore such sentences in semantic feature extraction: first, they represent noise that reduces the performance of the classifier; secondly, removing those sentences considerably reduces the size of the input and thus the training time, without negatively affecting the results. So, we remove such sentences before the next step.

Word-level feature extraction We represent individual words by word embedding in a continuous vector space; specifically, we experimented with the wor2vec embeddings [55]. This gives a variable-length feature set for the document: the document is represented as a variable number of sentences, which are represented as a variable number of fixed-length word feature vectors.

Classification For classification, we use a deep CNN. Its initial layers process the text in a hierarchical manner. Each word is represented in the input as a fixed-length feature vector using `word2vec`, while sentences are represented as a variable number of word vectors. At some layer, this variable-length representation is reduced to fixed-length representation of each sentence, which is a kind of sentence embedding in a continuous vector space. At that level, documents are represented as a variable number of such fixed-length sentence embeddings. Finally, at a deeper layer this variable-length document representation is reduced to a fixed-length document embedding. This fixed-length feature vector is then concatenated with the document-level features, given a fixed-length document representation, which is then used for final classification.

When aggregating word vectors into sentence vectors, we use convolution to form word n-gram features. However, when aggregating sentence vectors into the document vector, we do not use convolution to form sentence n-gram features. We did try this arrangement; however, the network did not converge in 75 epochs, so we left this experiment to our future work.

6.3.1 Network Architecture

We train five separate neural classifiers, all having the same architecture, for five different personality traits: EXT, NEU, AGR, CON, and OPN.

The processing flow in our network consists of three main steps:

1. Word vectorization: We use fixed-length `word2vec` word embeddings as input data;
2. Sentence vectorization: From sequences of words in each sentence to fixed-length sentence vectors;
3. Document vectorization: From the sequence of sentence vectors to the document vector;
4. Classification: From the document vector to the classification result (*yes* / *no*).

Accordingly, the network consists of seven layers:

1. Input layer (word vectors);

Sentence vectorization:

2. Convolution layer;
3. Max pooling layer;

Document vectorization:

4. 1-max pooling layer;
5. Concatenation layer;

Classification:

6. Linear layer with Sigmoid activation;
7. Two-neuron softmax output layer.

Figure 6.1 depicts the end-to-end network for two sentences. In the following subsections, we discuss these steps and layers in detail.

6.3.1.1 Input

We represent the dataset as a set of documents: each document d is a sequence of sentences; each sentence s_i is a sequence of words, and each word w_j is a real-valued vector of fixed length known as word embedding; in our experiments we used Google's pre-trained `word2vec` embeddings [55].

Thus, our input layer is a 4-dimensional real-valued array from $\mathbb{R}^{D \times S \times W \times E}$, where D is the number of documents in the dataset, S is the maximum number of sentences in a document across all documents, W is the maximum number of words in a sentence across all documents, and E is the length of word embeddings.

In implementation, to force all documents to contain the same number of sentences, we padded shorter documents by dummy sentences. Similarly, we padded shorter sentences by dummy words.

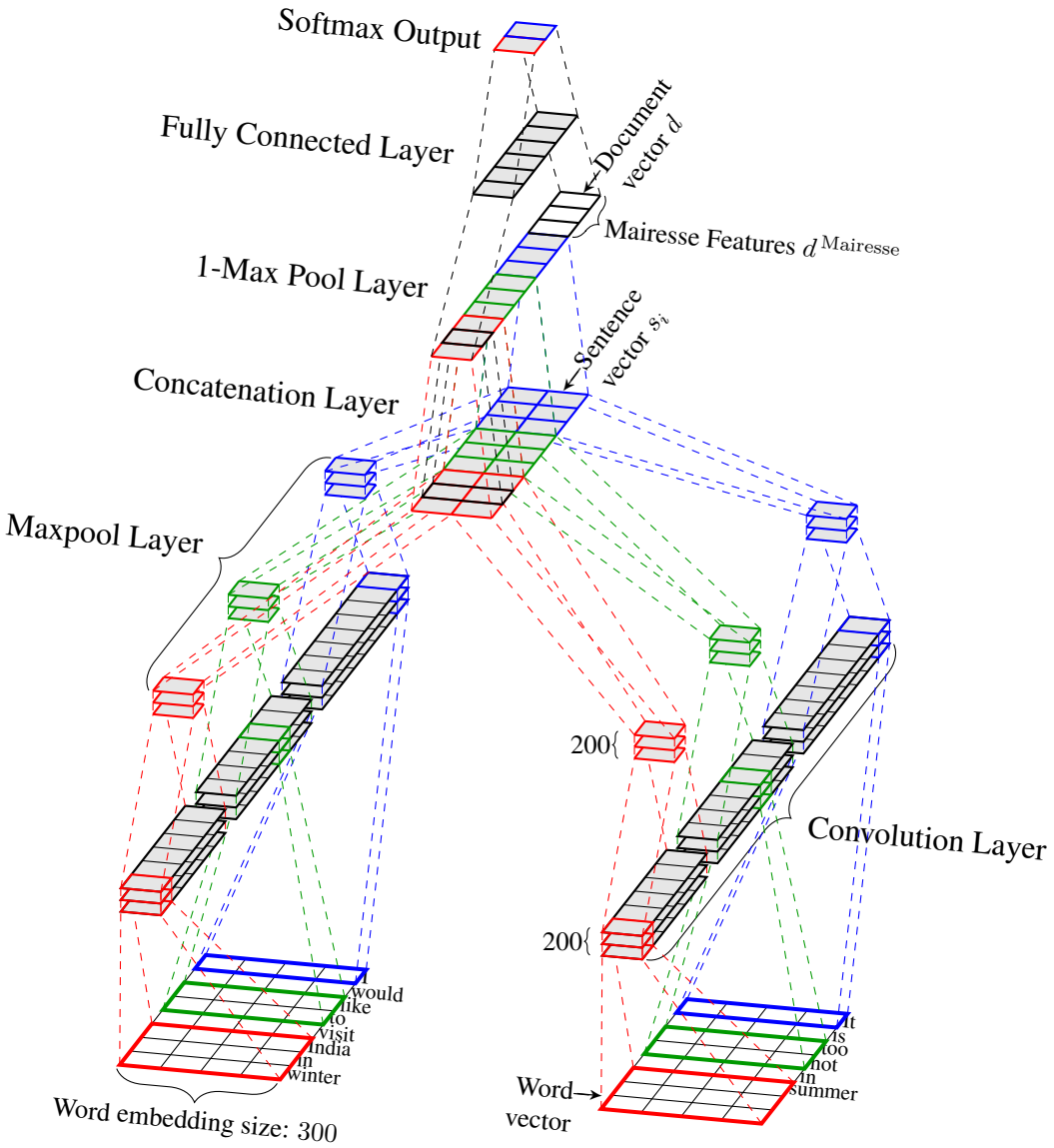


FIGURE 6.1: Architecture of our network

6.3.1.2 Aggregating Word Vectors into Sentence Vectors

We use three convolutional filters to extract unigram, bigram, and trigram features from each sentence. After max pooling, the sentence representation is a concatenation of the feature vectors obtained from these three convolutional filters.

Convolution In order to extract the n-gram features, we apply convolutional filter of size $n \times E$ on each sentence $s \in \mathbb{R}^{W \times E}$. We use 200 n-gram feature maps for each $n = 1, 2, 3$. So, for each n, our convolutional filter applied on the matrix s is $F_n^{\text{conv}} \in \mathbb{R}^{200 \times n \times E}$. We add a bias $B_n^{\text{conv}} \in \mathbb{R}^{200}$ to the output of the filter, which gives, for a given sentence, three feature maps $FM_n \in \mathbb{R}^{200 \times (W-n+1) \times 1}$, $n = 1, 2, 3$. To introduce non-linearity, we apply ReLU (Rectified Linear Unit) function to the feature maps FM_n.

Max pooling Then, we apply max pooling to each feature map FM_n, to further down-sample it to a feature map $DFM_n \in \mathbb{R}^{200 \times 1 \times 1}$, which we flatten to obtain feature-vector of size 200.

Finally, we concatenate the vectors obtained for the three types of n-gram to obtain a vector $s \in \mathbb{R}^{600}$ representing the sentence.

Convolution and max pooling are applied to each sentence in the document. The network parameters are shared between all sentences of the document. In particular, while we pad all sentences to a common size with dummy words, there is no need to pad all documents to a common size with dummy sentences.

6.3.1.3 Aggregating Sentence Vectors into Document Vector

After processing individual sentences, document representation is a variable-sized concatenation of all its sentence vectors.

1-max pooling: We assume that the document has some feature if at least one of its sentences has this feature. Each sentence is represented as a 600-dimensional vector. To obtain the document representation, for each of these 600 features we take the maximum across all the sentences of the document. This gives a 600-dimensional real-valued representation of the whole document: $d^{\text{network}} \in \mathbb{R}^{600}$.

6.3.1.4 Adding Document-Level Features to Document Vector

Mairesse et al. [46] developed a document-level feature set for personality detection. It consist of the LIWC features [49], Medical Research Council (MRC) features [56], utterance-type features, and prosodic features, such as word count, average number of words per sentence, total number of pronouns, past tense verbs, present tense verbs, and future tense verbs in the document, total number of letters, phonemes, syllables, questions, and assertions in the document, etc., 84 features in total.

Then we concatenated those 84 features d^{Mairesse} with the document representation d^{network} described in the previous section. This gave the final 684-dimensional document representation: $d = (d^{\text{network}}, d^{\text{Mairesse}}) \in \mathbb{R}^{684}$. We also use the feature set d^{Mairesse} as one of baselines in our evaluation.

6.3.1.5 Classification

For final classification, we use a two-layer perceptron consisting of a fully-connected layer of size 200 and the final softmax layer of size two, representing the *yes* and *no* classes.

Fully-connected layer We multiply the document $d \in \mathbb{R}^{684}$ by a matrix $W^{\text{fc}} \in \mathbb{R}^{684 \times 200}$ and add a bias $B^{\text{fc}} \in \mathbb{R}^{200}$ to obtain the representation $d^{\text{fc}} \in \mathbb{R}^{200}$. Introducing non-linearity with Sigmoid activation improved the results:

$$d^{\text{fc}} = \sigma(d\,W^{\text{fc}} + B^{\text{fc}}),$$

where

$$\sigma(x) = \frac{1}{1 + \exp(-x)}.$$

We have also experimented with ReLU and $tanh$ as activation functions, but they yielded lower results.

Softmax output We use softmax function to determine the probability of the document to belong to the class *yes* and *no*. For this, we build a vector

$$(x_{yes}, x_{no}) = d^{\text{fc}}\,W^{\text{sm}} + B^{\text{sm}},$$

where $W^{\text{sm}} \in \mathbb{R}^{200 \times 2}$ and the bias $B^{\text{sm}} \in \mathbb{R}^2$, and calculate the class probabilities as

$$P(i \mid \text{network parameters}) = \frac{\exp(x_i)}{\exp(x_{yes}) + \exp(x_{no})}$$

for $i \in \{yes, no\}$.

6.3.2 Training

We use Negative Log Likelihood as the objective function for training. We randomly initialize the network parameters F_1^{conv}, F_2^{conv}, F_3^{conv}, B_1^{conv}, B_2^{conv}, B_3^{conv}, W^{fc}, B^{fc}, W^{sm}, B^{sm}. We use Stochastic Gradient Descent (SGD) with Adadelta [24] update rules to tune the network parameters in order to minimize the error defined as Negative Log Likelihood. In our experiments, after 50 epochs the network converged, with 98% training accuracy.

6.4 Experimental Results

6.4.1 Dataset Used

We used Pennebaker and King's consciousness-stream essay dataset [49]. It contains 2468 essays written by anonymized authors, tagged with their personality traits: EXT,

NEU, AGR, CON, and OPN. We removed one essay from the dataset, which only contained the text "Err:508"; so we experimented with the remaining 2467 essays.

6.4.2 Experimental Setting

In all of our experiments, we used 10-fold cross-validation to evaluate the trained network.

Pre-processing We split the text into a sequence of sentences by the period and question mark characters. The we split each sentence into words by whitespace characters. We reduced all letters to lowercase and removed all characters other than ASCII letters, digits, exclamation mark, and single and double ASCII quotation marks.

Some essays in the dataset contain no periods, or some periods are missing. This resulted in absurdly long sentences. To handle such cases, we split each obtained "sentence" that was longer than 150 words into "sentences" of 20 words each (except the last piece that could happen to be shorter).

Extracting document-level features We used the [2] [46] library to extract the 84 Mairesse features from each document.

Sentence filtering We assumed that a relevant sentence would have at least one emotionally charged word. After extracting the document-level features but before extracting the `word2vec` features, we discarded all sentences that had no emotionally charged words.

We used the *NRC Emotion Lexicon*[3] [57], [58] to obtain emotionally charged words. This lexicon contains 14182 words tagged with ten attributes: *anger*, *anticipation*, *disgust*, *fear*, *joy*, *negative*, *positive*, *sadness*, *surprise*, and *trust*. We considered a word to be emotionally charged if it had at least one of these attributes; there are 6468 such words in lexicon (the majority of the words in this lexicon have no attributes).

So, if a sentence contained none of the 6468 words, we removed it before extracting the `word2vec` features from the text. In our dataset, all essays contained at least one emotionally-charged word.

We have also experimented with not removing any sentences and randomly removing half of each essay's sentences. Randomly removing half of the sentences improved the results as compared with no filtering at all; we do not have a plausible explanation of this fact. Removing emotionally-neutral sentences as described above further improved the results. Filtering also improved the training time by 33.3%.

[2] http://farm2.user.srcf.net/research/personality/recognizer.html
[3] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

Extracting word-level features We used the `word2vec`[4] embeddings [55] to convert words into 300-dimensional vectors. If a word was not found in the list, then we assigned all 300 coordinates randomly with a uniform distribution in $[-0.25, 0.25]$.

Word n-gram baseline As a baseline feature set, we used 30,000 features: 10,000 most frequent word unigrams, bigrams, and trigrams in our dataset. We used the Scikit-learn library [59] to extract these features from the documents.

Classification We experimented with three classification settings. In the variant marked MLP in Table 6.1, we used the network shown in Figure 6.1, which is a multiple-layer perceptron (MLP) with one hidden layer, trained together with the CNN. In the variant marked SVM in the table, we first trained the network shown in Figure 6.1 and then used the document vectors d to train a polynomial SVM of degree 3. In the variant marked sMLP in the table, in a similar manner we used the vectors d to train a standalone MLP (50 epochs) with the same configuration as the last two layers in Figure 6.1. Also, we experimented with a variation of sMLP, where we fed the output of hidden layer to the standalone MLP, marked by sMLP (Hi). For baseline experiments not involving the use of CNN, we used only a linear SVM.

6.4.3 Results and Discussion

Table 6.1 shows the obtained results. Using n-grams showed no improvement over the majority baseline: the classifier rejected all n-grams. Applying filtering and adding the document-level (Mairesse) features proved to be beneficial. In fact the CNN alone without the document-level features underperformed the Mairesse baseline; we attribute this to insufficient training data: our training corpus was only 1.9 million running words.

Contrary to our expectations, applying SVM to the document representation d built with the CNN did not improve the results. Rather surprisingly, applying a standalone MLP to d did improve the results. We cannot attribute this to the fact that the system was thus received additional 50 epochs of training, since the network used to build the document representation d has converged in its 50 epochs of initial training.

Increasing the window size for convolution filters did not seem to consistently improve the results; while the best result for the NEU trait was obtained with 2-, 3-, and 4-grams, even for this trait the sizes 1, 2, and 3 outperformed the current state of the art.

We have also tried a number of configurations not shown Table 6.1, as well as some variations of the network architecture. In particular, in addition to using convolution filters to obtain vector representation of each sentence, we tried using convolution filters to obtain document vector d from the sequence of sentence vectors s_i. However, training did not converge after 75 epochs; so we used 1-Max pool layer on the array of sentence vectors to obtain a the document vector.

[4] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

TABLE 6.1: Accuracy obtained with different configurations. For each trait, underlined are the best results obtained in our experiments but below the state of the art; shown in bold are the best results overall.

N	Document representation d	Filter	Classifier	Conv. filter	Personality Traits				
					EXT	NEU	AGR	CON	OPN
1.	—		Majority		51.72	50.02	53.10	50.79	51.52
2.	Word n-grams		SVM		51.72	50.26	53.10	50.79	51.52
3.	Mairesse (as reported in [51])		SVM		55.13	58.09	55.35	55.28	59.57
4.	Mairesse (our experiments)		SVM		55.82	58.74	55.70	55.25	60.40
5.	Published state of the art per trait (see [51, Table 8])				56.45	58.33	56.03	56.73	60.68
6.	CNN		MLP	1,2,3	55.43	55.08	54.51	54.28	61.03
7.	CNN		MLP	2,3,4	55.73	55.80	55.36	55.69	61.73
8.	CNN		SVM	2,3,4	54.42	55.47	55.13	54.60	59.15
9.	CNN + Mairesse		MLP	1,2,3	54.15	57.58	54.64	55.73	61.79
10.	CNN + Mairesse		SVM	1,2,3	55.06	56.74	53.56	56.05	59.51
11.	CNN + Mairesse	+	sMLP/FC	1,2,3	54.61	57.81	55.84	**57.30**	62.13
12.	CNN + Mairesse	+	sMLP/MP	1,2,3	**58.09**	57.33	**56.71**	56.71	61.13
13.	CNN + Mairesse	+	MLP	1,2,3	55.54	58.42	55.40	56.30	**62.68**
14.	CNN + Mairesse	+	SVM	1,2,3	55.65	55.57	52.40	55.05	58.92
15.	CNN + Mairesse	+	MLP	2,3,4	55.07	**59.38**	55.08	55.14	60.51
16.	CNN + Mairesse	+	SVM	2,3,4	56.41	55.61	54.79	55.69	61.52
17.	CNN + Mairesse	+	MLP	3,4,5	55.38	58.04	55.39	56.49	61.14
18.	CNN + Mairesse	+	SVM	3,4,5	56.06	55.96	54.16	55.47	60.67

6.5 Conclusions

We have presented a novel document representation technique based on CNN and have demonstrated its effectiveness on the personality detection task. A distinctive feature of our network architecture is per-sentence processing with later merging individual sentence vectors into the document representation. In addition, we introduced a document representation based on both CNN-based n-gram features and document-level stylistic features. Yet another contribution is the filtering out emotionally-neutral sentences.

Even without hand-crafted Mairesse features, our CNN outperformed the state of the art on the OPN trait and showed competitive results for other traits. Merging our CNN-generated features with the document-level Mairesse features allowed us to outperform the state of the art on all but one traits, with the result obtained for the latter trait being practically equal to the current state of the art.

In the future, we plan to incorporate more features and pre-processing. We plan to apply the Long Short-Term Memory (LSTM) recurrent network to build both the sentence vector from a sequence of word vectors and the document vector from a sequence of sentence vectors. In addition, we plan to apply our document representation to other emotion-related tasks such as sentiment analysis [60] or mood classification [61].

Chapter 7

Conclusions

In this chapter, we conclude this thesis with summarizing our contributions, listing our publications derived from this thesis, and discussing our future research plans.

7.1 Contributions

In this thesis, we developed novel techniques for sentiment analysis and opinion mining that outperform the existing approaches, introduced novel machine-learning models and deep learning architectures required for these techniques, and implemented a set of software modules for the application and evaluation of these techniques.

Opinion mining and sentiment analysis are booming research and application areas that have recently attracted much attention from industry, government, and academia. They have numerous applications in economy, security, and healthcare, to name only a few.

Methodological Contributions

The main methodological contributions of this thesis consist in the development of novel method for opinion mining and sentiment analysis applications as follows:

- **Multimodal Sentiment Analysis.** We developed a range of novel techniques for multimodal sentiment analysis—a task that consists in determining, from a video clip showing a person giving his or her opinion on some subject, whether the person's opinion is positive or negative, or whether he or she experiences positive or negative emotions while speaking. Specifically, our techniques include the following improvements:

  - Correct modeling of the text as a hierarchy of structures: words, sentences, paragraphs, etc. Before, in machine-learning applications text was usually modeled as a plain sequence of words or characters with no structure.
  - Taking into account the context of each utterance. Current published state-of-the-art methods classify the utterances separately, completely losing their relationship with the previous discourse.

  - Modeling of attention in natural language communication for applications of machine-learning techniques.

  We have provided empirical evidence for that each of these techniques significantly improves the accuracy of detection of sentiment and emotions in multimodal data.

- **Personality Detection from Text.** We developed a novel technique for detection of five personality traits from the standard five-factor model corresponding to the author of a given text. This is an important part of author profiling task, required for numerous applications of opinion mining such as recommender systems.

Theoretical Contributions

The main theoretical contributions of this thesis consist in introduction of novel machine-learning techniques and deep learning architectures as follows:

- **Improved Feature Fusion.** We proposed two different feature fusion methods more effective than current published state-of-the-art concatenation-based fusion methods:

  - *Hierarchical Fusion.* We developed a novel fusion mechanism that fuses features from different modalities in a hierarchical manner. For bimodal scenario, it fuses two feature vectors using a series of perceptrons. In case of trimodal fusion, it first fuses all bimodal combinations and the merges all three combinations using a series of perceptrons. This filters out redundant features. Also, it is capable of prioritizing a specific modality for a specific feature.

    Paper currently under review.

  - *Attention-Based Fusion.* We introduced the use of attention mechanism to fuse feature vectors from different modalities into an unified vector space. This method is capable of amplifying important modalities, while impeding less informative modalities.

    Paper currently under review.

- **Inter-Utterance Dependency.** We applied the Long Short-Term Memory (LSTM) deep learning architecture to model inter-utterance dependencies in text, speech, or video stream. We have shown that this captures the context of a particular utterance, leading to more accurate sentiment polarity assignment of the utterance. Inclusion of attention mechanism on top of LSTM amplifies more important utterances which are relevant to the context.

  Paper accepted for ACL 2017, the largest and **most prestigious** international conference on Natural Language Processing.

- **Hierarchical Document Modeling.** We applied the Convolutional Neural Network (CNN) deep learning architecture to model the sentences in the text and then merged them using 1-max-pooling. Also, we introduced a technique based on filtering out emotionally-neutral sentences using an emotion lexicon, which further improved performance.

  Paper published in a JCR-indexed journal, **impact factor 3.532**.

Technical Contributions

We used Theano [62] to implement the discussed neural network architectures. The implementation of Personality Detection algorithm (discussed in Chapter 6) is shared in the following link: `https://github.com/SenticNet/personality-detection`. The implementations for rest of the methods will be released upon acceptance of their corresponding papers.

7.2 Publications

Published or Accepted

1. N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning based document modeling for personality detection from text", IEEE Intelligent Systems, vol. 32, no. 2, pp. 74–79, 2017.

   JCR impact factor 2015: **3.532**.

2. S. Poria, D. Hazarika, N. Majumder, A. Zadeh, L.-P. Morency, and E. Cambria, "Modeling Contextual Relationships Among Utterances for Multimodal Sentiment Analysis", in ACL, 2017.

   ACL is the **largest** and **most prestigious** conference of Natural Language Processing.
   Acceptance rate 2016: **28%**.

Under Evaluation

Due to double blind review policies of the corresponding journals and conferences, we cannot disclose here the specific titles and author lists of the submitted papers; we will only specify the general topics of the currently submitted papers.

3. Paper on application of LSTM to aspect-based sentiment analysis: application of the techniques developed in Chapter 4; submitted.

4. Paper on application of attention mechanism to multimodal sentiment analysis: results of Chapter 5; submitted.

5. Paper on application of hierarchical fusion mechanism to multimodal sentiment analysis: results of Chapter 3; submitted.

6. Paper on application of modified LSTM architecture to multimodal sentiment analysis: results of Chapters 4 and 5; submitted.

7. Paper on application of hierarchical fusion mechanism and LSTM to multimodal deception detection: results of Chapter 3; submitted.

7.3 Future Work

In the future, we plan to apply our hierarchical fusion (Chapter 3) with LSTM. Particularly, we want to replace AT-Fusion (Section 5.3.4) with hierarchical fusion in CATF-LSTM in Figure 5.1. Also, we plan to improve the features for audio and video modalities, since both of them contributes very little to the classification process. We also want to apply LSTM for personality detection which will build both sentence representation and document representation. We mean to include a social media component to sentiment analysis. It will help extract sentiment of a specific demographic about a particular topic.

Bibliography

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *ArXiv preprint arXiv:1301.3781*, 2013.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation.", in *EMNLP*, vol. 14, 2014, pp. 1532–43.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch", *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

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

[5] E. Cambria, "Affective computing and sentiment analysis", *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques", in *Proceedings of ACL*, Association for Computational Linguistics, 2002, pp. 79–86.

[7] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank", in *Proceedings of EMNLP*, 2013, pp. 1631–1642.

[8] P. Ekman, "Universal facial expressions of emotion", *Culture and Personality: Contemporary Readings/Chicago*, 1974.

[9] D. Datcu and L Rothkrantz, "Semantic audio-visual data fusion for automatic emotion recognition", *Euromedia'2008*, 2008.

[10] L. C. De Silva, T. Miyasato, and R. Nakatsu, "Facial emotion recognition using multi-modal information", in *Proceedings of ICICS*, IEEE, vol. 1, 1997, pp. 397–401.

[11] L. S. Chen, T. S. Huang, T. Miyasato, and R. Nakatsu, "Multimodal human emotion/expression recognition", in *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE, 1998, pp. 366–371.

[12] L. Kessous, G. Castellano, and G. Caridakis, "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis", *Journal on Multimodal User Interfaces*, vol. 3, no. 1-2, pp. 33–48, 2010.

[13] B. Schuller, "Recognizing affect from linguistic information in 3d continuous space", *IEEE Transactions on Affective Computing*, vol. 2, no. 4, pp. 192–205, 2011.

[14] M. Wollmer, F. Weninger, T. Knaup, B. Schuller, C. Sun, K. Sagae, and L.-P. Morency, "Youtube movie reviews: Sentiment analysis in an audio-visual context", *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 46–53, 2013.

[15] V. Rozgic, S. Ananthakrishnan, S. Saleem, R. Kumar, and R. Prasad, "Ensemble of svm trees for multimodal emotion recognition", in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, IEEE, 2012, pp. 1–4.

[16] A. Metallinou, S. Lee, and S. Narayanan, "Audio-visual emotion recognition using gaussian mixture models for face and voice", in *Tenth IEEE International Symposium on ISM 2008*, IEEE, 2008, pp. 250–257.

[17] F. Eyben, M. Wöllmer, A. Graves, B. Schuller, E. Douglas-Cowie, and R. Cowie, "On-line emotion recognition in a 3-d activation-valence-time continuum using acoustic and linguistic cues", *Journal on Multimodal User Interfaces*, vol. 3, no. 1-2, pp. 7–19, 2010.

[18] C.-H. Wu and W.-B. Liang, "Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels", *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 10–21, 2011.

[19] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis", in *Proceedings of EMNLP*, 2015, pp. 2539–2544.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[21] V. Teh and G. E. Hinton, "Rate-coded restricted boltzmann machines for face recognition", in *Advances in neural information processing system*, T Leen, T Dietterich, and V Tresp, Eds., vol. 13, 2001, pp. 908–914.

[22] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor", in *Proceedings of the 18th ACM international conference on Multimedia*, ACM, 2010, pp. 1459–1462.

[23] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[24] M. D. Zeiler, "ADADELTA: an adaptive learning rate method", *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701.

[25] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, "Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages", *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 82–88, 2016.

[26] V. Pérez-Rosas, R. Mihalcea, and L.-P. Morency, "Utterance-level multimodal sentiment analysis", in *ACL (1)*, 2013, pp. 973–982.

[27] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion", *Information Fusion*, 2017.

[28] V. Rozgic, S. Ananthakrishnan, S. Saleem, R. Kumar, and R. Prasad, "Ensemble of SVM trees for multimodal emotion recognition", in *Proceedings of APSIPA ASC*, IEEE, 2012, pp. 1–4.

[29] F. Gers, "Long short-term memory in recurrent neural networks", PhD thesis, Universität Hannover, 2001.

[30] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification", in *The 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 207–213.

[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization", *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[32] M. Wöllmer, F. Eyben, S. Reiter, B. W. Schuller, C. Cox, E. Douglas-Cowie, R. Cowie, *et al.*, "Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies.", in *Interspeech*, vol. 2008, 2008, pp. 597–600.

[33] Y. Kim, "Convolutional neural networks for sentence classification", *ArXiv preprint arXiv:1408.5882*, 2014.

[34] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models", *ArXiv preprint arXiv:1508.06615*, 2015.

[35] F. Eyben and B. Schuller, "Opensmile:): The munich open-source large-scale multimedia feature extractor", *ACM SIGMultimedia Records*, vol. 6, no. 4, pp. 4–13, 2015.

[36] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition", in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 3687–3691.

[37] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional mkl based multimodal emotion recognition and sentiment analysis", in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, IEEE, 2016, pp. 439–448.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[39] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention.", in *ICML*, vol. 14, 2015, pp. 77–81.

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

[41] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network", *ArXiv preprint arXiv:1605.08900*, 2016.

[42] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization", *ArXiv preprint arXiv:1509.00685*, 2015.

[43] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom, "Reasoning about entailment with neural attention", *ArXiv preprint arXiv:1509.06664*, 2015.

[44] Boundless, *Boundless Psychology*. Boundless.com, May 2016. [Online]. Available: https://www.boundless.com/psychology/textbooks/boundless-psychology-textbook/personality-16/introduction-to-personality-76/defining-personality-303-12838/.

[45] J. Digman, "Personality structure: Emergence of the five-factor model", *Annual Review of Psychology*, vol. 41, pp. 417–440, 1990.

[46] F. Mairesse, M. Walker, M. Mehl, and R. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text", *Journal of Artificial Intelligence Research (JAIR)*, vol. 30, pp. 457–500, 2007.

[47] E. Tupes and R. Christal, "Recurrent personality factors based on trait ratings", Lackland Air Force Base, TX: Personnel Laboratory, Air Force Systems Command, Tech. Rep. ASD-TR-61-97, 1961.

[48] L. Goldberg, "The structure of phenotypic personality traits", *American Psychologist*, vol. 48, pp. 26–34, 1993.

[49] J. W. Pennebaker and L. A. King, "Linguistic styles: Language use as an individual difference", *Journal of Personality and Social Psychology*, vol. 77, pp. 1296–1312, 1999.

[50] J. W. Pennebaker, R. J. Booth, and M. E. Francis, *Linguistic Inquiry and Word Count: LIWC2007, Operator's manual.* Austin, TX: LIWC.net, 2007.

[51] S. M. Mohammad and S. Kiritchenko, "Using hashtags to capture fine emotion categories from tweets", *Comput. Intell.*, vol. 31, no. 2, pp. 301–326, May 2015, ISSN: 0824-7935. DOI: 10.1111/coin.12024. [Online]. Available: http://dx.doi.org/10.1111/coin.12024.

[52] F. Liu, J. Perez, and S. Nowson, "A language-independent and compositional model for personality trait recognition from short texts", *CoRR*, vol. abs/1610.04345, 2016. [Online]. Available: http://arxiv.org/abs/1610.04345.

[53] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks", in *Proceedings of the 26th International Conference on Computational Linguistics (COLING), Osaka*, COLING, 2016.

[54] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network", *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.

[55] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781.

[56] M. Coltheart, "The mrc psycholinguistic database.", *Quarterly Journal of Experimental Psychology*, vol. 33A, pp. 497–505, 1981.

[57] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon", *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

[58] S. Mohammad and P. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon", in *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, LA, California, Jun. 2010.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[60] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives", in *Proceedings of the 26th International Conference on Computational Linguistics (COLING), Osaka*, 2016.

[61] B. G. Patra, D. Das, and S. Bandyopadhyay, "Multimodal mood classification framework for Hindi songs", *Computación y Sistemas*, vol. 20, pp. 515–526, 3 2016.

[62] Theano Development Team, "Theano: a Python framework for fast computation of mathematical expressions", *ArXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688.