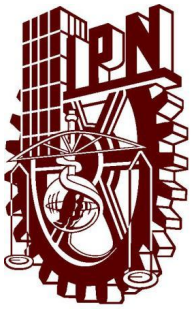


Instituto Politécnico Nacional

Centro de Investigación en Computación

TESIS:

Latent Memory-Based Neural Models for Sentiment Analysis of Multimodal Multi-Party Conversations

QUE PARA OBTENER EL GRADO DE:
Doctorado en Ciencias de la Computación

PRESENTA:
M. en C. Navonil Majumder

DIRECTORES DE TESIS:
Dr. Alexander Gelbukh
Dr. Soujanya Poria

México, CDMX Mayo 2020

SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 14:00 horas del día 02 del mes de marzo del 2020 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: Centro de Investigación en Computación para examinar la tesis titulada:

"Latent Memory-Based Neural Models for Sentiment Analysis of Multimodal Multi-Party Conversations"

del (la) alumno (a):

Apellido Paterno:	Majumder	Apellido Materno:	----------	Nombre (s):	Navonil

Número de registro: B 1 7 0 2 6 2

Aspirante del Programa Académico de Posgrado: Doctorado en Ciencias de la Computación

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 15 % de similitud. **Se adjunta reporte de software utilizado**.

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** [ ] **NO** [x] **SE CONSTITUYE UN POSIBLE PLAGIO.**

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*

En el análisis, fueron excluidos como fuentes 1) los artículos donde el alumno es coautor. 2) citas. 3) la lista de las referencias. Con esto el sistema mostró 15% de similitud, lo que es completamente aceptable. Cabe mencionar que la fuente de mayor similitud tenía el valor de similitud de 2%, lo que es normal por el uso de los términos en común.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** [x] **SUSPENDER** [ ] **NO APROBAR** [ ] la tesis por **UNANIMIDAD** [ ] o **MAYORÍA** [ ] en virtud de los motivos siguientes:

La tesis refleja un estudio original del tema con suficiente grado de completitud, ampliamente confirmado a través de publicaciones relevantes del alumno en revistas especializadas con alto factor de impacto y en congresos especializados del mayor prestigio en el área.

COMISIÓN REVISORA DE TESIS

Dr. Alexander Gelbukh

Dr. Ildar Batyrshin

Dra. Olga Kolesnikova

Soujanya Poria

Dr. Soujanya Poria

Dr. Grigori Sidorov

Dr. Marco Antonio Moreno Ibarra
PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de **México** el día **01** del mes **Mayo** del año **2020**, el que suscribe **M. en C. Navonil Majumder** alumno del Programa de **Doctorado en Ciencias de la Computación** con número de registro **B170262**, adscrito a **Centro de Investigación en Computación**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de **Dr. Alexander Gelbukh y Dr. Soujanya Poria** y cede los derechos del trabajo intitulado **Latent Memory-Based Neural Models for Sentiment Analysis of Multimodal Multi-Party Conversations**, 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

Debido a la rápida expansión de Internet y la proliferación de dispositivos inteligentes, el uso compartido y el consumo del contenido del usuario a través de estos medios han explotado a un nivel sin precedentes. Una parte importante de este contenido compartido comprende opiniones sobre diversos temas, como revisiones de productos, comentarios políticos. Muchas grandes empresas están interesadas en aprovechar esta información abiertamente disponible para su beneficio. Específicamente, tienen la intención de construir sistemas que recopilen automáticamente los comentarios de los usuarios al examinar un gran volumen de contenido del usuario. Dicha información podría ayudar a tomar decisiones comerciales, asignación de recursos, evaluación de riesgos, estudios de mercado, por mencionar algunos. Esta recopilación de comentarios suele ser conveniente y completa en términos de *sentimiento y emoción*. Con este fin, en esta tesis, presentamos métodos basados en redes neuronales que se adaptan al análisis de sentimientos y emociones en diferentes escenarios.

Recientemente ha habido un aumento en el intercambio de opiniones a través de videos debido a la mayor accesibilidad de la cámara de teléfono inteligente de buena calidad. Como los videos a menudo contienen tres modalidades: textual, acústica y visual, el análisis de sentimientos y emociones de estos videos requiere algoritmos de análisis de emociones y sentimientos multimodales. Un componente clave de cualquier algoritmo multimodal es la fusión multimodal. Como tal, proponemos un algoritmo de fusión basado en codificador automático variacional no supervisado cuya representación latente se utiliza como representación multimodal.

A menudo, el sentimiento del usuario sobre aspectos específicos de un objeto es más útil que la impresión general. El análisis de sentimientos basado en aspectos (ASBA) se vuelve relevante en tales escenarios. Sin embargo, la mayoría de los trabajos existentes sobre ASBA no consideran la coexistencia de múltiples aspectos en una sola oración. Presentamos un método que ajusta las representaciones de aspecto comparándolo con los aspectos vecinos usando la red de memoria.

Como consecuencia de las personas que interactúan y discuten en plataformas como Facebook, YouTube, Reddit, la estructura general del contenido termina siendo conversacional. Estas conversaciones a menudo contienen más de dos partes. El análisis de sentimientos y emociones de tales conversaciones multipartitas requiere algoritmos conscientes de la parte. Por lo tanto, presentamos un modelo basado en redes neuronales recurrentes (RNR) para el reconocimiento de emociones en la conversación (REC) que es capaz de una clasificación de emociones de nivel de expresión específica del hablante. A diferencia de los enfoques existentes, nuestro método no está limitado por el número de hablantes definidos por la arquitectura del modelo o el conjunto de entrenamiento. Esto se logra mediante el perfil dinámico de las partes a lo largo de la conversación utilizando la estructura similar a RNR. Como tal, obtenemos un rendimiento de vanguardia en conjuntos de datos REC diádicos y multipartitos.

Abstract

Owing to the quick expansion of internet and proliferation of smart-devices, sharing and consumption of user content through these means have exploded to unprecedented level. A significant portion of this shared content comprises of opinion on various topics, such as, product reviews, political commentary. Many large enterprises are keen on leveraging this openly available data to their benefit. Specifically, they intend to build systems that would automatically gather user feedback by sifting through huge volume of user content. Such information could aid in making business decisions, resource allocation, risk assessment, market survey to mention a few. This feedback gathering is often convenient and comprehensive in terms of *sentiment and emotion*. To this end, in this thesis, we present neural network-based methods that cater to sentiment and emotion analysis in different scenarios.

There has been a recent surge in opinion sharing via videos due to increased accessibility of good quality smartphone camera. As videos often contain three modalities — textual, acoustic, and visual — sentiment and emotion analysis of these videos calls for multimodal sentiment and emotion analysis algorithms. A key component of any multimodal algorithm is multimodal fusion. As such, we propose an unsupervised variational auto-encoder-based fusion algorithm whose latent representation is used as multimodal representation. We gain improvement over the state-of-the-art multimodal sentiment and emotion analysis algorithms with this method.

Often user sentiment on specific aspects of an object is more useful than overall impression. Aspect-based sentiment analysis (ABSA) becomes relevant in such scenarios. However, most existing works on ABSA do not consider co-existence of multiple aspects in a single sentence. We present a method that fine-tunes the aspect representations by comparing with the neighboring aspects using memory network. We empirically show that this approach beats the state of the art on multiple domains.

As a consequence of people interacting and arguing on platforms like Facebook, YouTube, Reddit, the overall content structure ends up conversational. These conversations often contain more than two parties. Sentiment and emotion analysis of such multi-party conversations requires party-aware algorithms. Hence, we present a recurrent neural network- (RNN) based model for emotion recognition in conversation (ERC) that is capable of speaker-specific utterance-level emotion classification. Unlike the existing approaches, our method is not bound by the number of speakers defined by model architecture or training set. This is achieved by dynamic profiling of the parties along the conversation using the RNN-like structure. As such, we obtain state-of-the-art performance on both dyadic and multi-party ERC datasets.

Acknowledgements

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

Stephen Hawkings

My tenure as doctoral candidate at *Centro de Investigación en Computación* (CIC) of *Instituto Politécnico Nacional* (IPN) has been one of the productive and rewarding periods of my life. I enjoyed the classes given by the respected professors of this institute. 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 advisers *Dr. Alexander Gelbukh* and *Dr. Soujanya Poria* for their endless support through expert guidance and motivation that have swiftly steered me to the end of my PhD. Specifically, I am greatly indebted to the priceless years worth of wisdom and experience in academia they shared with me. At the same time, their technical acumen has helped navigate the vastly complex area of NLP research.

I greatly appreciate the administration of CIC for their unrelenting support and administrative guidance. 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) and *BEIFI* for their recurrent financial support, due to which I could devote myself full-time to research and study. Also, I would like to express my gratitude towards *Coordinación de Cooperación Académica* (CCA) for sponsoring my travel to the venue of my internship.

A lot of thanks to my friends and fellow students of CIC for their unyielding support. Also, I greatly appreciate the discussions we had in our laboratory seminars that inspired several research ideas within me. 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 immense gratitude towards my parents for great upbringing.

Contents

List of Figures

List of Tables

List of Abbreviations

ABSA	**A**spect-**B**ased **S**entiment **A**nalysis
AE	**A**uto **E**ncoder
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
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
ELMo	**E**mbedding from **L**anguage **Mo**dels
ERC	**E**motion **R**ecognition in **C**onversation
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
MAE	**M**ean **A**bsolute **E**rror
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
OOV	**O**ut **O**f **V**ocabulary
RNN	**R**ecurrent **N**eural **N**etworks
RNTN	**R**ecursive **N**eural **T**ensor **N**etwork
ReLU	**Re**ctified **L**inear **U**nit
SGD	**S**tochastic **G**radient **D**escent
SLP	**S**ingle **L**ayer **P**erceptron
SOTA	**S**tate **O**f **T**he **A**rt
SVM	**S**upport **V**ector **M**achine
TF-IDF	**T**erm **F**requency-**I**nverse **D**ocument **F**requency
VAE	**V**ariational **A**uto **E**ncoder

Chapter 1

Introduction

This chapter deals with introducing the readers to the problem at hand. We deconstruct the problem into multiple subproblems and discuss each subproblem individually. We also provide necessary background knowledge as to its relevance in modern society. Further, we describe our solutions to these problems at conceptual level, along with their novelties. In the end, we point out our contributions to science.

1.1 Background

It's a foregone conclusion that internet is one of the linchpins of modern society. Various aspects of our daily activities rely on internet — daily commute (Uber), shopping (Amazon), entertainment (Netflix, YouTube), education (Coursera), finance (net-banking), and so on.

E-commerce services, like Amazon, have enabled people with access to internet to make purchases with the press of a button, saving numerous hours for productivity and leisure. Again, the proliferation of smart devices has enabled social-media platforms, like Facebook, Instagram, YouTube, to have great influence on public opinion. Users post their opinion on various topics, such as politics, sports, food, products, etc, on these platforms. These opinions are shared in huge quantity each day, in usually textual or audio-visual form.

This huge quantity of data is being leveraged by various large enterprises to boost their sales and revenue. E-commerce services, like Amazon, make recommendations to their patrons based on their prior purchases and sentiment on those purchases. On the other hand, product designers often want to get user feedback on various aspects of their products to narrow down drawbacks of their products and plan improvements. This aids companies to make critical business decisions based on user sentiment towards their products.

On platforms like Facebook, Twitter, users often interact with each other as they express their opinions. This leads to conversation among various parties. Hence, processing all these conversational or otherwise multimodal or textual opinionated data and extracting sentimental or emotional information from them warrant scalable algorithms. To this end, in this thesis, we tackle a few types of chosen sentiment and emotion extraction problems (Section 1.2) with neural network-based algorithms.

1.2 Problems

The principal problems that we deal with in this thesis are sentiment analysis and emotion detection. The methods we discuss are applicable to both of sentiment and emotion detection as they are closely related problems, emotion detection being more fine-grained than sentiment detection. Further, we explore the multimodal extension of both of these problems.

1.2.1 Sentiment and Emotion Classification

1.2.1.1 Sentiment Analysis

The task is to assign appropriate sentiment label to a snippet of text, be it an utterance, paragraph, or document. In this work, we perform sentiment analysis at utterance level. Usually, the set of labels consists of three labels — *positive*, *negative*, and *neutral*. Table 1.1 illustrates these labels.

Sentiment	Example
Positive	*The display is gorgeous.*
Negative	*The phone is too bulky.*
Neutral	*It is powered by an Intel Core i5 processor.*

TABLE 1.1: Different Sentiment Polarities.

Chapters 3 and 4 deal with sentiment analysis.

1.2.1.2 Emotion Classification

Similar to sentiment classification, emotion classification is the assignment of appropriate emotion label to a piece of text, be it an utterance, paragraph, or document. In this thesis, we deal with utterance-level emotion classification. The emotion labels that we used in our work are illustrated in Table 1.2.

Emotion	Sample
Happy	*Today, I got a big raise along with promotion.*
Sad	*It's a shame that John hurt his ankle before the expedition.*
Neutral	*DHL is a courier service.*
Angry	*Don't ever call here again!*
Excited	*I have to finish this thesis as soon as possible.*
Frustrated	*Finishing this draft is taking longer than I expected.*

TABLE 1.2: Illustration of utterance-level emotion classification.

Chapters 3 and 5 deal with emotion classification.

1.2.2 Aspect-Based Sentiment Analysis

Often, a single utterance or sentence holds of multiple sentiments on multiple objects (or aspects). As such, extraction of those aspects and their corresponding sentiment are two different tasks that emerge. In this thesis, we deal with the latter task of sentiment classification of given aspects in the sentence. Table 1.3 illustrates aspect-based sentiment analysis.

Sample	Aspects (Sentiment)
Frenchie's have mediocre *food*, but great *service*.	food (*negative*), service (*positive*)

TABLE 1.3: Illustration of Aspect-Level Sentiment.

Chapter 4 deals with aspect-based sentiment analysis.

1.2.3 Multimodal Sentiment and Emotion Classification

Unlike regular sentiment/emotion classification, multimodal sentiment/emotion classification also deals with visual and acoustic information, apart from textual information. This particularly comes into play for the extraction of sentiment/emotion in videos or audios, where the extra modalities might lead to better classification result.

The main challenge of any multimodal task is the fusion of modalities. Chapter 3 deals with multimodal fusion.

1.2.4 Emotion and Sentiment Classification in Conversations

Conversational emotion and sentiment classification operates on conversation level in addition to utterance level. The annotation is performed at utterance level like regular utterance-level emotion and sentiment classification. However, speaker and turn information are provided as well to aid the classification. Fig. 1.1 illustrates one such conversation. Chapter 5 deals with emotion recognition in conversations.

1.3 Relevance

Due to the massive growth of opinionated data on various topics, publicly available over the internet, automated sentiment and emotion classification have risen to great relevance. Major reviewers publish their opinions on newly released products on the internet, discussing various facets of the target products. Even numerous general end users chime in with their opinion. As such, the product manufacturers are keen on user reception of many attributes of their product. They use this valuable information to make critical business decisions, as to resource allocation — aspects with poor reception require more resource for improvement; design decisions — design choices appreciated in a rival product should be in incorporated. For instance, a company releasing a new

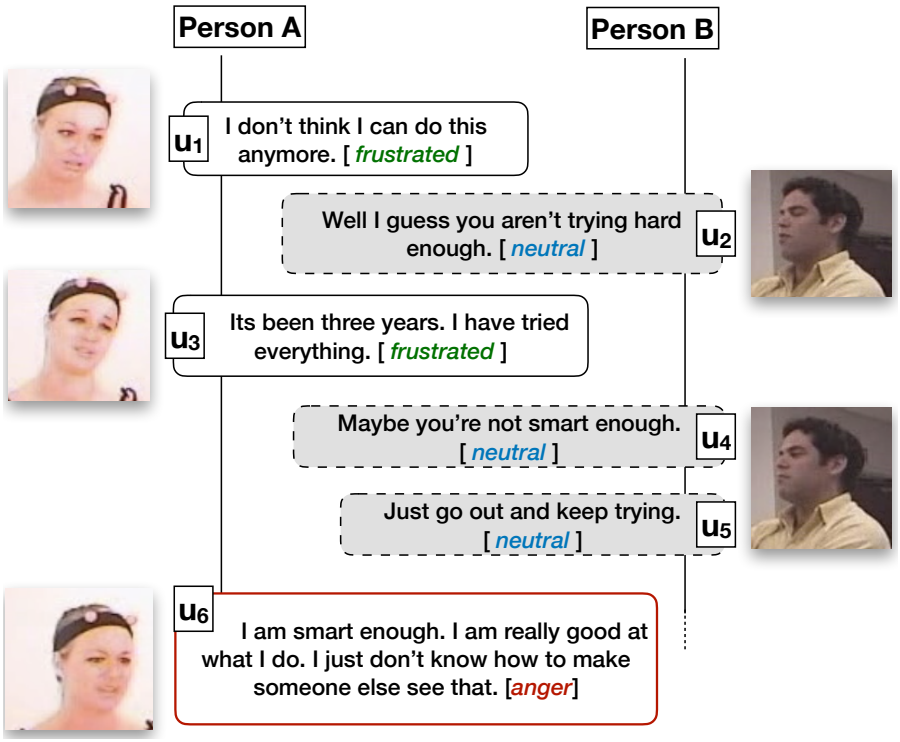


FIGURE 1.1: Illustration of a conversation where the constituent utterances are labeled with corresponding emotion labels.

smart-watch into the market would be interested in the near universally denounced poor battery life, so that they could fix it in the next iteration.

Moreover, recently, there has been a trend of releasing opinion content in video form to maximize reachability and quick information conveyance. Since, videos have multiple channels of information, emotion and sentiment information tend to be more accurate than solely textual channel. Hence, companies are more and more interested in opinions expressed in videos.

E-commerce services can and do leverage the users' experience and complaints about past purchases to make relevant recommendations.

Also, this opinion information can contribute to risk assessment by conveying public sentiment about specific choices that are being considered to be implemented.

Again, conversational emotion recognition systems can be used in empathetic dialogue generation systems in healthcare domain or customer relationship management, where the response emotion and content of the dialogue system should be determined by the user input (emotion and content), as illustrated in Fig. 1.2.

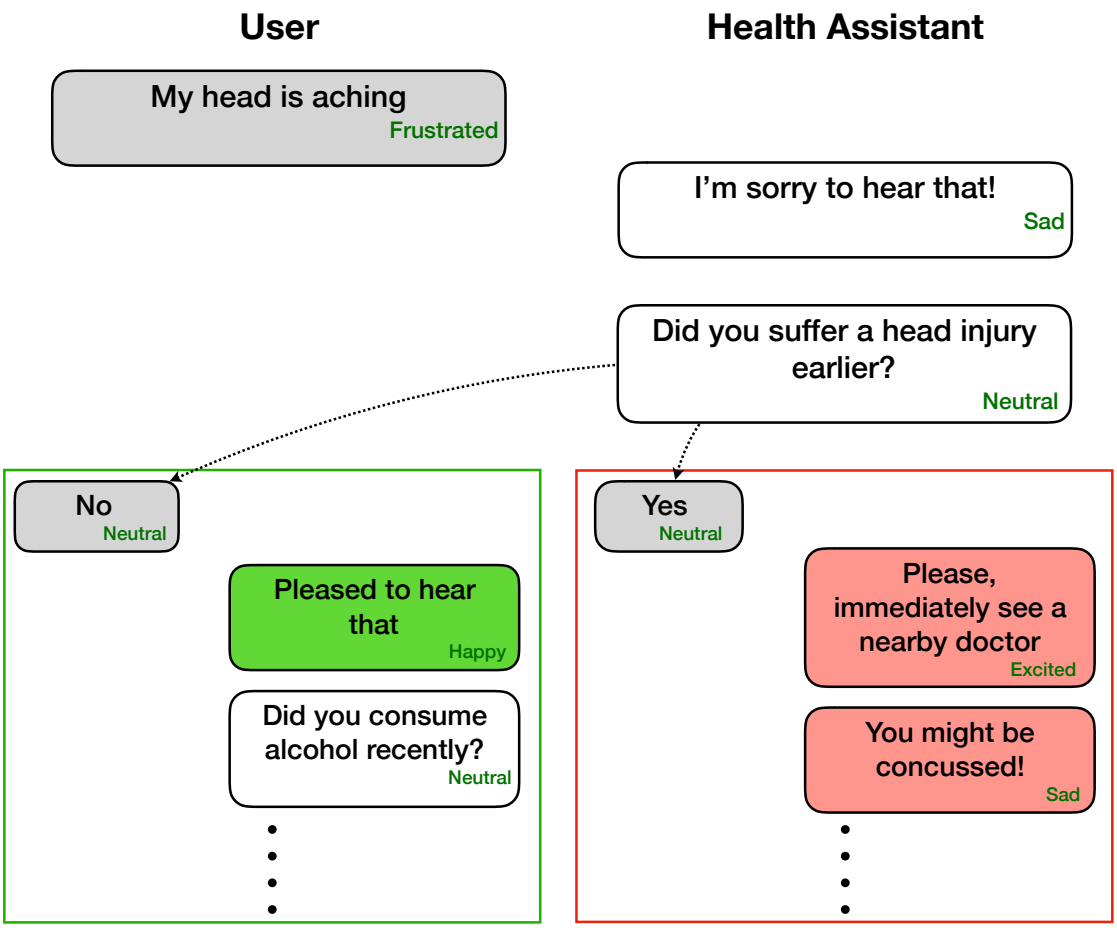


FIGURE 1.2: Example of empathetic dialogue generation based on user input.

1.4 Novelty

1.4.1 Multimodal Sentiment Analysis

Fusion of information from multiple modalities is the most crucial hurdle of multimodal sentiment analysis or any multimodal task for that matter. Recently, myriad of multimodal fusion schemes (Majumder et al., 2018; Zadeh et al., 2017, 2018a,c) have been proposed. These methods only fuse (encoding) the modality features into a unified feature. However, we go one step further by mapping (decoding) the fused feature back to the original unimodal features to improve information retention. We reckon this strategy forces the fusion model to retain both modality specific and invariant features at the same time, resulting in fused feature vector with distinct features. We discuss this method in details in Chapter 3.

Further, since, this strategy has distinct encoder and decoder parts, any new fusion strategy can be used as encoder to boost the performance of newer fusion strategies.

1.4.2 Aspect-Based Sentiment Analysis (ABSA)

Most ABSA algorithms treat the aspects in the same sentence independently. However, it is not always the case. For example, "*Their food was much better than their service*" has two aspects *food* and *service*. Due to the usage of 'better', the sentiment of *food* is dependent on the sentiment of *service* and vice versa.

Hence, we employ memory networks (Sukhbaatar et al., 2015) to model the dependency among the aspects within the same sentence, which resulted in improved performance over SOTA. This is discussed in details in Chapter 4.

1.4.3 Emotion Recognition in Conversation (ERC)

Most of the existing works on emotion recognition in conversation do not utilize speaker information effectively and in a scalable fashion. We, however, present a RNN-based method that profiles individual speaker as the conversation proceeds. As such, it is capable of handling multi-party conversations in a scalable way. Moreover, it is also capable of handling arbitrary number of speakers in a conversation without retraining the model. In Chapter 5, we discuss this in details and show that our method outperforms the SOTA.

1.5 Contributions

The followings are the contributions of this thesis:

- **Improved Multimodal Feature Fusion —** Development of feature fusion scheme that is capable of boosting performance of existing feature fusion schemes;
- **Improved Unsupervised Multimodal Feature Fusion —** Development of feature fusion method without labeled data;
- **Inter-Aspect Dependency Modeling —** Modeling of inter-aspect dependency for aspect-based sentiment analysis, leading to more precise aspect-level sentiment classification;
- **Improved Incorporation of Speaker Information in ERC —** Usage of on-the-fly speaker profiling for ERC for improved classification performance;
- **Scalable Multi-Party ERC —** Capable of handling multi-party conversations without the increase of number of model parameters with speaker count;
- **Improved Real-time Multi-Party ERC —** Capable of generating emotion predictions in realtime with a small modification to the model.

1.6 Structure of this Document

- **Chapter 1** (the current chapter) introduces the problems we strive to solve in this thesis and their importance;
- **Chapter 2** briefly introduces the reader to the theoretical elements employed to solve the problems at hand;
- **Chapter 3** describes our novel multimodal fusion method;
- **Chapter 4** discusses inter-aspect dependency-based aspect-level sentiment classification model;
- **Chapter 5** presents our RNN-based method for emotion recognition in conversation that performs speaker profiling in realtime;
- **Chapter 6** concludes this thesis by elaborating our contributions, along with mentioning the publications born out of this thesis and the future works projected.

Chapter 2

Theoretical Framework

We provide necessary theoretical background as to different methods, tools, metrics to facilitate reader understanding of the subsequent chapters. Firstly, we briefly discuss different emotion and sentiment frameworks. Further, we discuss a few text representation methods, which is followed by classification techniques. Then, we delve into some elementary neural-network architectures and training neural networks. Finally, we close off this chapter by explaining the methods for validating and evaluating classification methods on a given dataset.

2.1 Emotion and Sentiment Framework

Emotion Framework. There are two basic types of emotion categorization models — categorical and dimensional. Categorical models define emotion as a finite set of discrete categories. On the other hand, dimensional models define emotion as a continuous multi-dimensional space, where each point in the space corresponds to some emotional state.

One of the most popular categorical models, Plutchik (1982)'s wheel of emotion (Fig. 2.1) posits eight distinct primary emotion types. Each primary type is defined as a collection related subtypes. On the other hand, Ekman (1993) only defines six basic emotion types — anger, disgust, fear, happiness, sadness, and surprise.

On the dimensional categorization front, most such models (Mehrabian, 1996; Russell, 1980) define two dimensions — valence and arousal. Valence denotes the degree of positive emotion and arousal represents the intensity of given emotion.

Compared to categorical models, dimensional models map emotion onto a continuous space. It facilitates simple and intuitive comparison of two emotional states using vector operations and vector similarity metrics. In contrast, comparison between categorical states is non-trivial.

It can be challenging to choose categorization model for annotation, due to the availability of several taxonomies. For example, a simple model like Ekman's model has limited capability of grounding complex emotions. However, complex models like Plutchik's model make annotation very difficult due to the existence of subtly-different and related emotion categories, e.g., anger and rage. Besides, this can also lead to low annotator agreement.

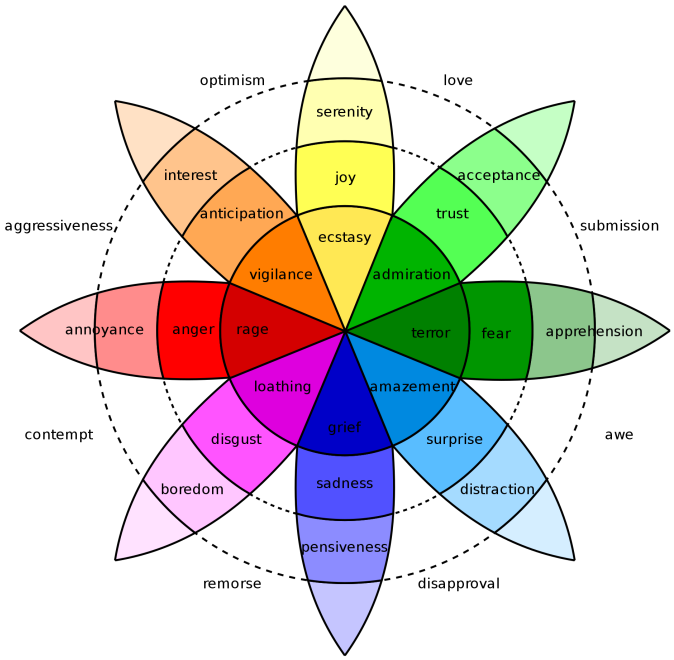


FIGURE 2.1: Plutchik's wheel of emotion (Plutchik, 1982).

Sentiment Framework. Sentiment framework, however, is usually much simpler and unambiguous, consisting of three sentiment types — positive, negative, and neutral.

2.2 Text Representations

2.2.1 Bag of Words (BOW)

BOW is arguably the simplest form of text representation, which only considers two factors — vocabulary of text and presence of constituent words. In the simplest form, words in BOW are represented as one-hot vectors of length of the size of vocabulary, where each element represents an unique word in the vocabulary. However, term frequency-inverse document frequency (TF-IDF) (Ramos, 2003; Robertson, 2004) based vectors are more popular and effective. However, the major two drawbacks of BOW are:

1. **sparsity —** word representation vectors are as large as the size of the vocabulary (which can be millions) which is not scalable to deep neural networks or even large volume of data,

2. **loss of word-order —** word order is not captured in document representations, leading to loss of syntactic information. Bag of n-grams is often used to alleviate this, but it does not scale well to large volume of data.

As such, distributed word representations become necessary, as discussed in the following sections.

2.2.2 Word2vec Embeddings

Mikolov et al. (2013) catapulted the use of deep learning in NLP through word2vec distributed word embeddings. These embeddings are build from large corpus in such a way that relationships among words are preserved in the vector space where the embeddings lie in. The two variants of word2vec are continuous-bag-of-words (CBOW) and SkipGram.

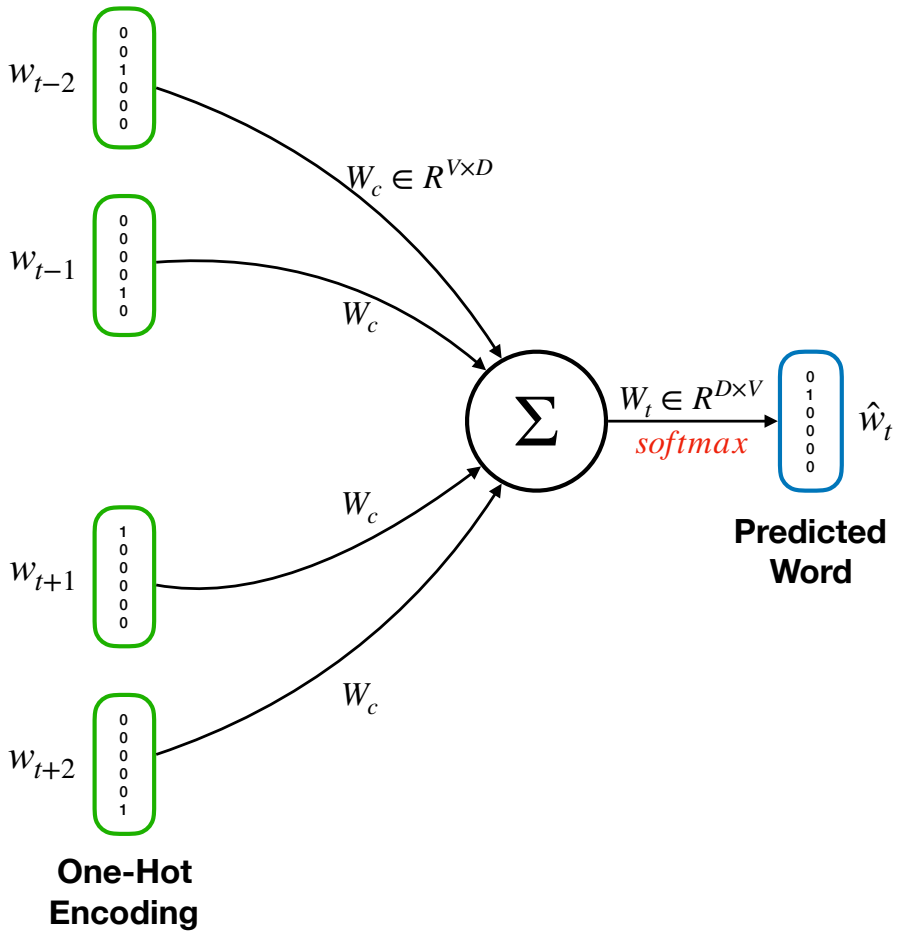


FIGURE 2.2: Word2vec CBOW model.

CBOW — Given a fixed number of context words, within fixed context window, surrounding the target word, the model predicts the target word using a simple neural network depicted in Fig. 2.2. Here, the rows of the weight matrix represent the word embeddings that are trained. The final softmax layer computes probability over the entirety of the vocabulary, which is often computationally infeasible. To circumvent this issue negative sampling strategy is adopted. This reduces the task to a binary classification problem. Instead of predicting the target word directly, the task is now to predict if a series of *(target word, context word)* pairs belong to the corpus. Alternatively, instead

of negative sampling, the softmax function can be approximated using hierarchical softmax (Morin and Bengio, 2005), differentiated softmax (W. Chen et al., 2016).

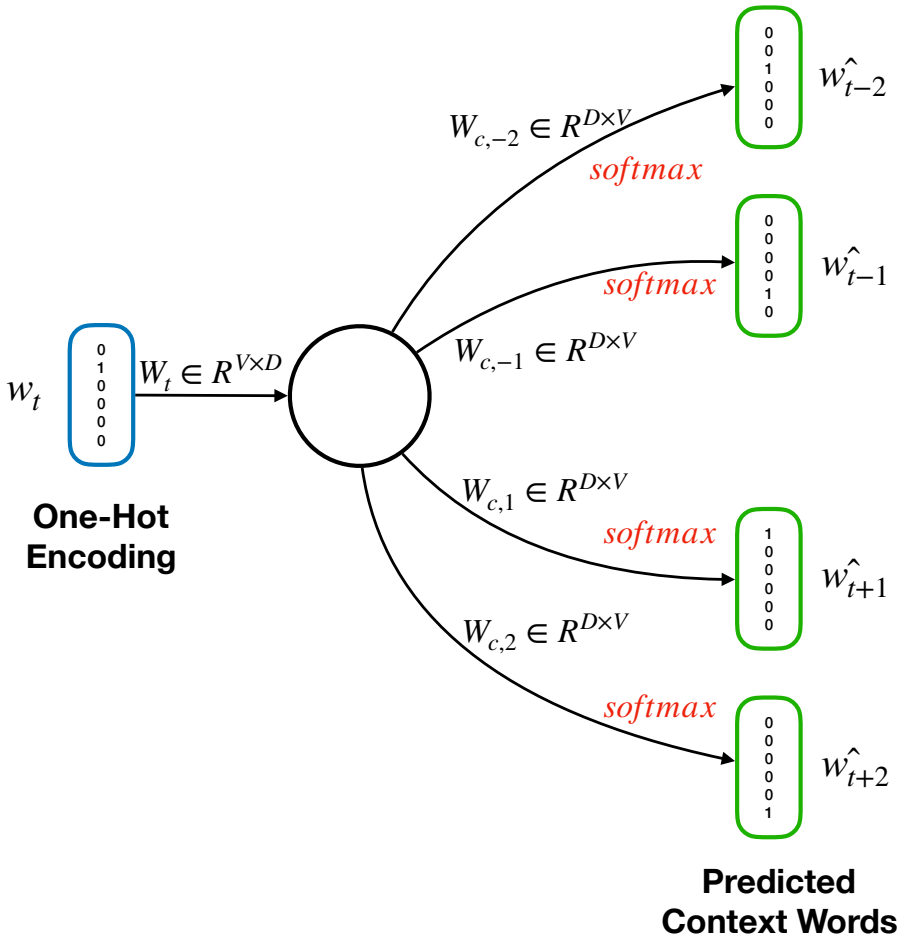


FIGURE 2.3: Word2vec SkipGram model.

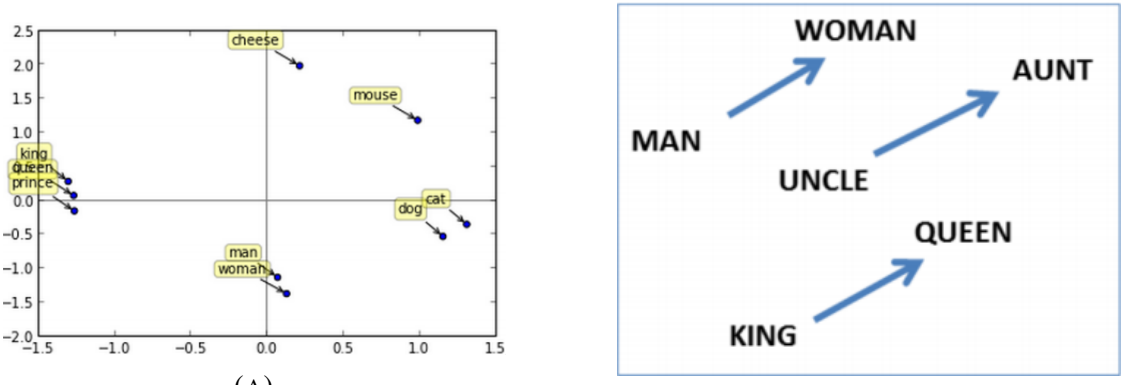
SkipGram — This model is mirror image of CBOW. Here, given a single word, the model predicts the context words. Fig. 2.3 shows this model.

Word2vec vectors are trained to have much smaller dimensionality (pretrained Google word2vec vectors are of size 300). One important property of word2vec is — similar words are closer in the vector space. Fig. 2.4a shows 2D projection of word2vec embeddings where similar words are clustered together.

Further, Fig. 2.4b shows that pairs of words having similar relationship between them have similar distance vectors. As such, semantics is built into the vector space.

2.2.3 GloVe Embeddings

GloVe (Pennington et al., 2014) model adopts similar principal as word2vec (Mikolov et al., 2013), by considering the co-occurrence of target and context word within corpus.



(A) (B)

FIGURE 2.4: (A) 2D projection of word2vec embeddings where semantically similar words are closer; (B) Vector differences between words signifying *gender*.

However, unlike word2vec that only considers local statistics, GloVe also uses global statistics by incorporating co-occurrence counts between words into the training objective (Eq. (2.1)). Moreover, GloVe deliberately infuses semantics into vector space (as in Fig. 2.4b), unlike word2vec where the semantics emerges as a side effect of the training process.

The following is the training objective of GloVe:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(\mathbf{w_i}^T \mathbf{w_j} + b_i + b_j - \log X_{ij}), \tag{2.1}$$

$$f(x) = \begin{cases} (x/x_{\max})^\alpha & \text{if } x < x_{\max}, \\ 1 & \text{otherwise,} \end{cases}$$

where $\mathbf{w_i}, \mathbf{w_j} \in \mathbb{R}^D$ are embeddings of i^{th} and j^{th} word in vocabulary, respectively; b_i, b_j are biases for i^{th} and j^{th} word, respectively; X_{ij} is the co-occurrence count between i^{th} and j^{th} word; V is the vocabulary size; f is the weighting function, where usually $\alpha = 3/4$ and $x_{\max} = 100$.

2.2.4 Contextual Embeddings

1. The bank blocked Jason's credit card.
2. Segun sat by the river bank, pondering the future.
3. Jason sat sulking by the river bank, contemplating suing the bank.

Financial Institution **Shore**

FIGURE 2.5: Two different semantics of the word *bank*, depending on the context.

One drawback of word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) embeddings is those remain static, regardless of the sentence they appear in (in other words, context). Fig. 2.5 illustrates a scenario of *polysemy*, where two different semantics of the word *bank* are appear. Word2vec and GloVe treat both *bank*s as the same word, hence, same embedding and semantics, which is wrong. Recently, pre-training models like Embedding from Language Models (ELMo) (Peters et al., 2018) and Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) have been proposed that generates embeddings specific to the sentences. Employing these embeddings has significantly improved performance of many tasks, such as question answering, natural language inference, co-reference resolution, etc. These models also allow fine-tuning specific to task at hand to obtain even better embeddings.

ELMo and BERT both are trained by solving some sentence-level problems, auto-generated from the corpus. ELMo is composed of two language models that are trained on large corpus. BERT, on the other hand, predicts randomly masked tokens in sentences (masked language model) and if a given sentence follow the other in the corpus. ELMo uses bidirectional long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) (Section 2.4.4) to construct the language models. BERT, however, uses transformer networks (Vaswani et al., 2017) which operates faster than LSTMs. Unlike static embeddings, ELMo is immune to out-of-vocabulary (OOV) issue due to character-level input encoding. BERT, on the other hand, uses WordPiece (Wu et al., 2016) encoding that somewhat alleviates OOV problem, but not fully. Overall however, BERT outperforms ELMo on most tasks, due to harder training objective that captures more intricate linguistic features.

In our work, we do not employ these contextual embeddings for the sake of fair comparison with the SOTA methods that do not use contextual embeddings.

2.3 Classification Techniques

2.3.1 Perceptron

Single-Layer Perceptron (SLP) is the simplest form of classifier that is capable of obtaining a hyperplane separating data-points of two distinct classes. Fig. 2.6 illustrates such a case there two classes (red and blue) are separated by two among infinitely many possible straight line (hyperplane in 2D space).

The general equation of hyperplane in nD space is

$$\mathbf{w}^T\mathbf{x} + b = 0, \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a point in nD space. We optimize parameters $\mathbf{w} \in \mathbb{R}^n$ and b of Eq. (2.2) such that the resulting hyperplane linearly separates data-points $\mathbf{x}$. In other words, the decision function is

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

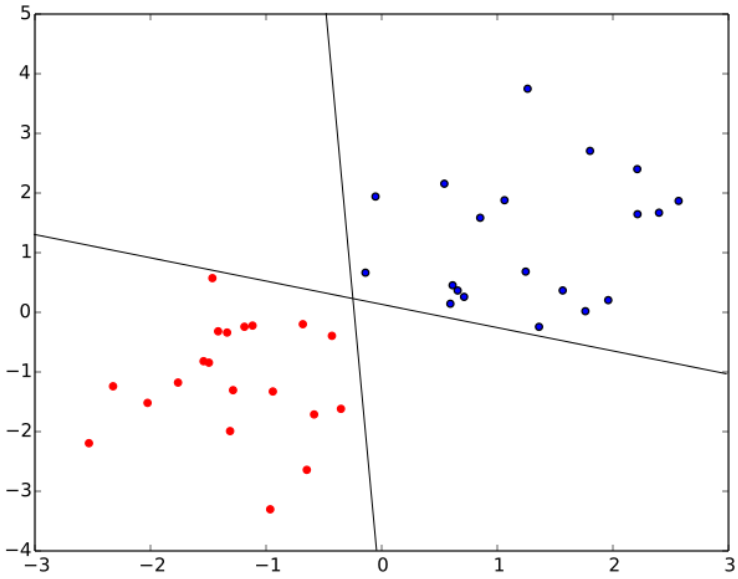


FIGURE 2.6: Two hyperplanes separating linearly separable data-points with two distinct classes.

The following steps are performed to optimize parameters $\mathbf{w}$ and b:

1. Initialize parameters $\mathbf{w}$ and b with small random values close to zero. Often values sampled from standard normal distribution are used, i.e., $\mathbf{w(0)} \sim \mathcal{N}(0, I)$ and $b(0) \sim \mathcal{N}(0, 1)$.

2. For each sample $(\mathbf{x_j}, y_j)$ in the training set at iteration t, we perform

$$\hat{y_j}(t) = f(\mathbf{x_j}), \tag{2.4}$$
$$\mathbf{w(t+1)} = \mathbf{w(t)} + C(y_j - \hat{y_j}(t))\mathbf{x_j}, \tag{2.5}$$
$$b(t+1) = b(t) + C(y_j - \hat{y_j}(t)), \tag{2.6}$$

   where C is some learning rate (usually set to 0.001 or 0.0001).

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

2.3.2 Logistic Regression / Single-Layer Perceptron

The drawback of simple perceptron is that it is ineffective against linearly inseparable data-points, as shown in Fig. 2.7. To introduce non-linearity, we feed the output of

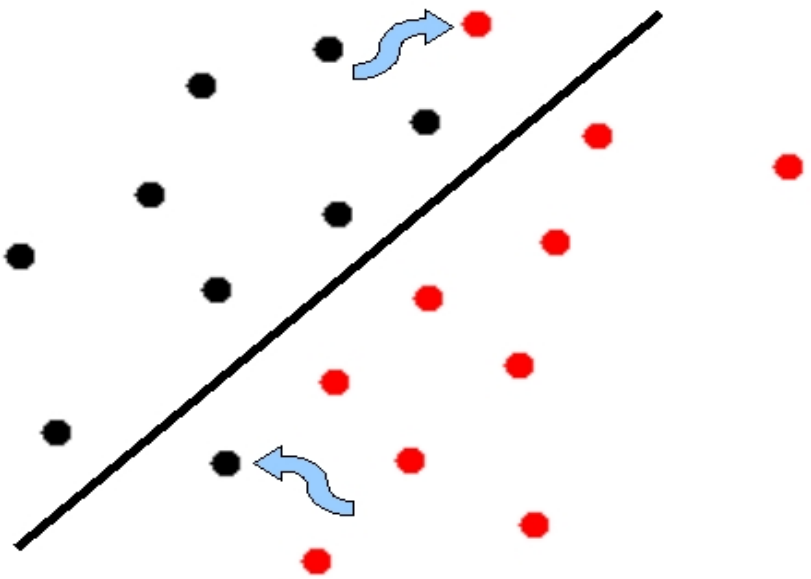


FIGURE 2.7: Linearly non-separable data.

$\mathbf{w}^T.\mathbf{x} + b$ through sigmoid activation (σ), where

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{2.7}$$

Fig. 2.8 depicts the shape of Sigmoid function. Since, the mid part of the curve is linear, manipulating parameters $\mathbf{w}$ and b can make the decision boundary linear of the data be linearly separable. Henceforth, we rewrite the decision-function as

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \sigma(\mathbf{w}^T\mathbf{x} + b) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

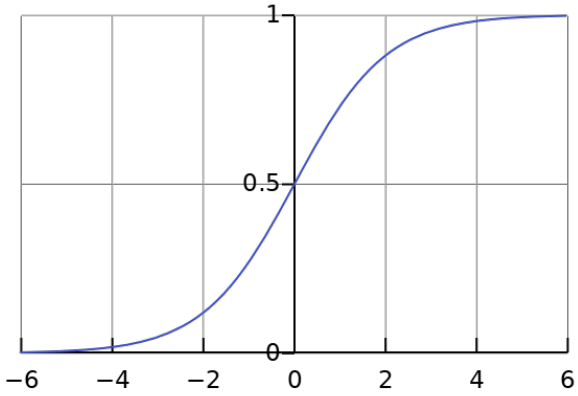


FIGURE 2.8: Sigmoid function (σ).

Since, the range of sigmoid is $[0, 1]$, its output can be interpreted as probability. Thus, the probabilities of $\mathbf{x}$ belonging to class 1 and 0 are $\mathcal{P}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$ and $1 - \mathcal{P}(\mathbf{x})$, respectively. Further, we write the joint class probability of all training samples as

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

where $y_j \in \{0, 1\}$ is the expected label of sample j. Naturally, our goal is to maximize the value of P. To this end, we use stochastic gradient descent (SGD). However, we need to transform this maximization problem to minimization problem.

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

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

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

$$\log P = \sum_{j=1}^{s} [y_j \log \mathcal{P}(\mathbf{x_j}) + (1 - y_j) \log (1 - \mathcal{P}(\mathbf{x_j}))] \tag{2.11}$$

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

$$J = -\frac{1}{s} \sum_{j=1}^{s} [y_j \log \mathcal{P}(\mathbf{x_j}) + (1 - y_j) \log (1 - \mathcal{P}(\mathbf{x_j}))] \tag{2.12}$$

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

2.3.3 Multi-Layer Perceptron

Logistic regression is effective in various tasks. However, certain arrangements of data-points that cannot be reasonably separated using logistic hypersurfaces. For instance, Fig. 2.9 shows four points with two classes (XOR function), which cannot be separated using a single straight-line (even with logistic-curve). Clearly, it requires two straight-lines to separate those.

To solve this, we employ multiple perceptrons. We introduce two perceptrons connected to the input, which constitute the hidden layer. Such connections are called dense connections (or full connections). Each perceptron stands for a straight-line (or hyper-plane in general). We feed the output of those two perceptrons to another perceptron, which is the overall output. This network is depicted in Fig. 2.10.

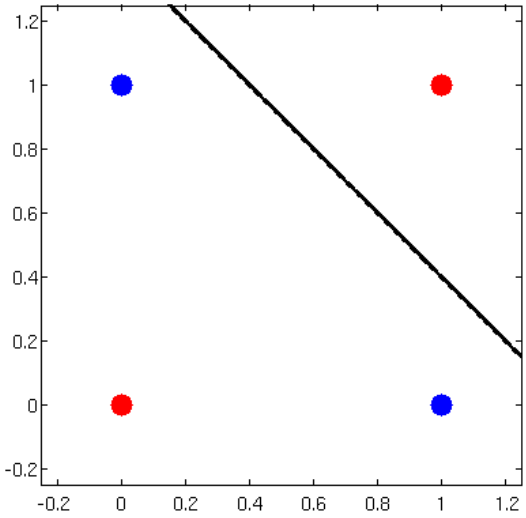


FIGURE 2.9: XOR function: Linearly non-separable.

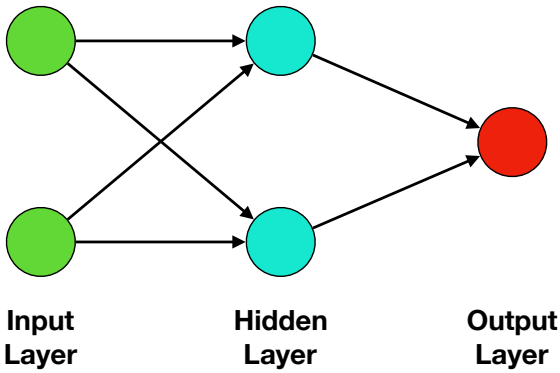


FIGURE 2.10: Multi-Layer Perceptron for XOR function.

The structure of XOR network is trivial, since the arrangement of the data-points is known and simple. However, for most applications the distribution of the data-points is unknown. As such, the networks are usually built with much higher number of perceptrons with various activation functions and multitude of layers (these are called hyper-parameters). As such, this is called multi-layer perceptron (MLP).

We perform the optimization of the parameters using SGD or one of its variants, as discussed in Section 2.5. Hyper-parameters are fine tuned based on performance on validation split, manually or using grid-search. Although, bayesian hyper-parameter optimization (Snoek et al., 2015) is often used.

2.3.4 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, support vector machine (SVM), etc

are still very relevant techniques in data-poor scenarios. Moreover, naive bayes, random forest, decision tree are widely used where interpretability of the model is critical.

2.4 Neural Network Architectures

Dense or fully-connected layer discussed in Section 2.3.3 is very important building block of neural networks. However, it alone rarely suffices in practice, without manual feature engineering. Since, manual feature engineering is often limited in coverage and not well scalable, specialized architecture like convolutional neural networks (CNN), recurrent neural networks (RNN) are employed to process and filter relevant information from raw input data directly, as per the task.

2.4.1 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are inspired by arrangement of cells (neurons) in visual cortex of animals. Individual cells are only receptive of certain sub-area of the visual field, acting as filter on the corresponding sub-area. However, as a whole these cells cover the entire visual field. CNN mimics this strategy over the input space, which is often an image or a sentence.

2.4.1.1 Convolution Filter

CNN applies convolution filter on a sub-region of the image at a time to yield a scalar in the feature map. This filter is slid across the image to form the complete feature map, as illustrated in Fig. 2.11.

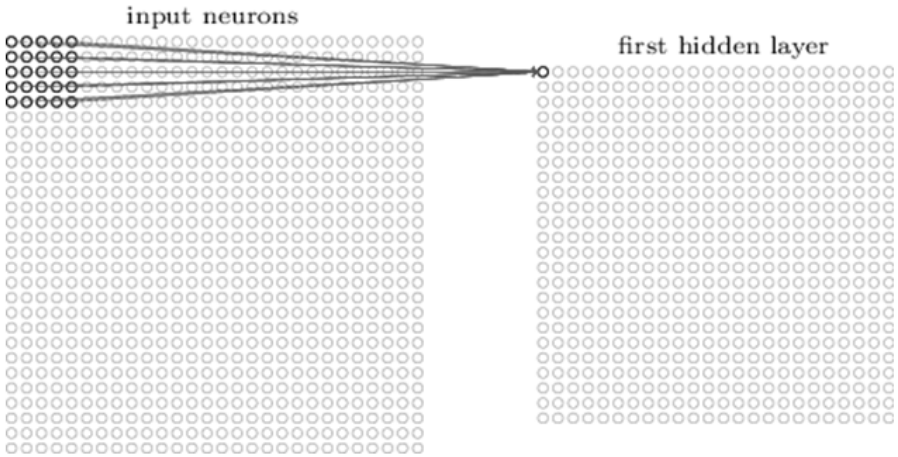


FIGURE 2.11: Feature map derived from original image; 5×5 filter sliding along input volume, yielding activation.

Multiple of such feature maps can be produced, using different filter parameters, from a single image. Therefore, k-th feature map is represented as f^k, where

$$f_{ij}^k = ReLU((W^k * x)_{ij} + b^k), \tag{2.13}$$

$$ReLU(x) = \max(0, x), \tag{2.14}$$

$W^k \in \mathbb{R}^{n \times m}$ and b^k are the weight and bias, respectively, of convolutional filter of size $n \times m$, i and j are row and column index of a neuron in feature map. Activation function, rectified linear unit ($ReLU$), is widely used in deep networks, to mitigate vanishing gradient problem. In general, feature maps production can be represented as depicted in Fig. 2.12.

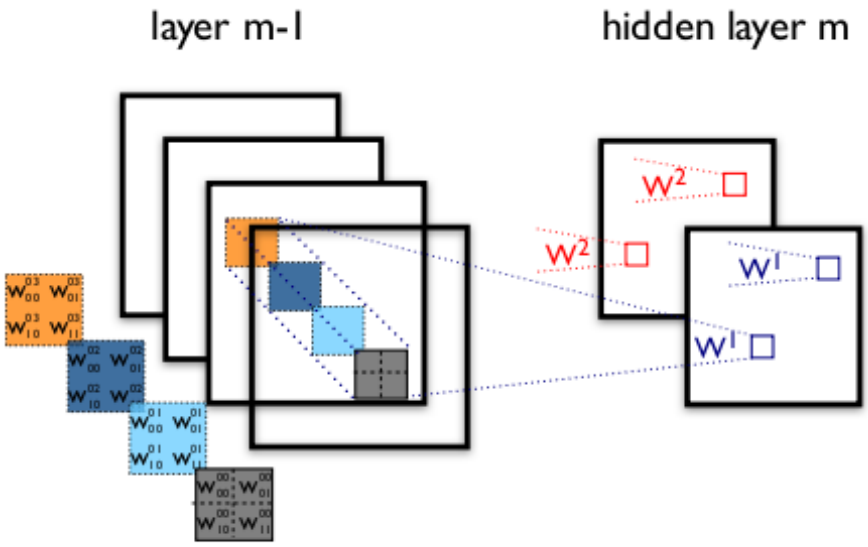


FIGURE 2.12: New feature maps derived from previous feature maps.

2.4.1.2 Pooling

Pooling operation is often an important part of CNN-based networks. It only passes a single value obtained from sliding window, of stipulated size, over the feature map. This filters out less relevant features within given region, that enhances task performance. Also, this reduces downstream computation cost per feature map, allowing higher number of subsequent feature maps. Max pooling is the most frequently used form of pooling, where the maximum value within a window is passed. However, there exists average pooling, min pooling.

2.4.1.3 Classification

Such layers of CNN and max pooling are often stacked and the output of the final layer is flattened to a vector. This vector is represents the input image with all the relevant information for classification. This vector is fed to a MLP for final classification. Fig. 2.13 illustrates a full scale CNN for image classification.

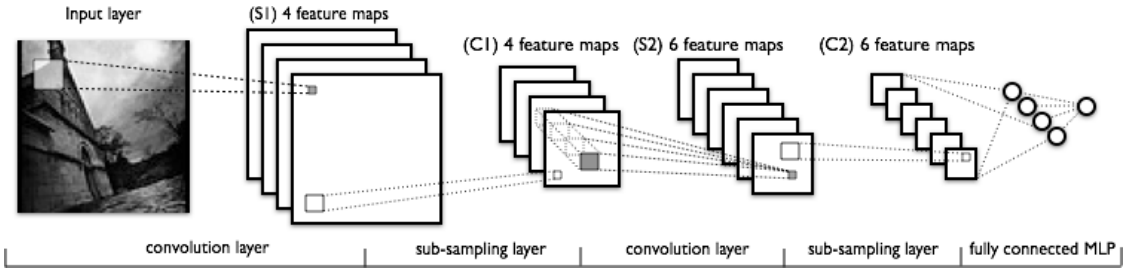


FIGURE 2.13: CNN for image classification.

2.4.1.4 Applications of CNN in NLP

CNN is applied to solve various NLP tasks, including generating sentence representation from word embeddings (Section 2.2). The constituent word embeddings of the sentence are stacked in order into a matrix, resembling an image. CNN of filter size $n \times d$ is applied to this matrix, where d is embedding size and n represents n-gram, to obtain n-gram feature maps. This is further fed through a series of pooling and CNN layers to finally classify it. Chapter 3 discusses the application of CNN for utterance representation generation.

2.4.2 3D Convolutional Neural Network (3D-CNN)

3D-CNN is 3D extension of regular 2D-CNN (Section 2.4.1). The convolution is performed on a window of frames 2D frames at a time. Fig. 2.14 shows the architecture of a 3D-CNN-based network with 3D-max-pooling and a classifier at the end.

Let $V \in \mathbb{R}^{c \times f \times h \times w}$ be a video, where c is the number of channels in an image ($c = 3$ for RGB images), f is the number of frames, and $h \times w$ is the size of a frame. Again, we consider the 3D convolutional filter $F \in \mathbb{R}^{fm \times c \times fl \times fh \times fw}$, where fm is the number of feature maps, c is the number of channels, fd is the number of frames (in other words depth of the filter), and $fh \times fw$ is the size 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.

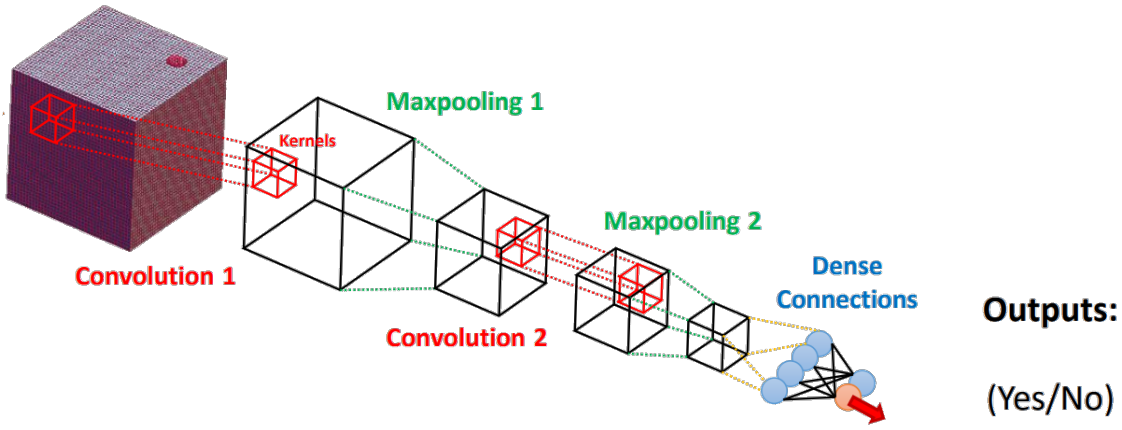


FIGURE 2.14: 3D-CNN.

2.4.3 Recurrent Neural Network (RNN)

Recurrent neural networks (RNN) are a class of neural networks where the connection among the nodes form at least one directed cycle or loop.

RNNs usually process one input per cell and retain that information in memory for the next cell (that share parameters) where it takes another input and so forth. As such, RNNs are specially apt in handling sequential information. Therefore, these are frequently used in many NLP task to encode and decode sentences, that are intrinsically sequential.

In theory, RNNs are capable of retaining every relevant information it has encountered at some point. However, in practice these only retains information from a few steps back. To circumvent this critical issue RNNs with specific structure, like long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) (Section 2.4.4), are used, that are much less inclined to memory loss.

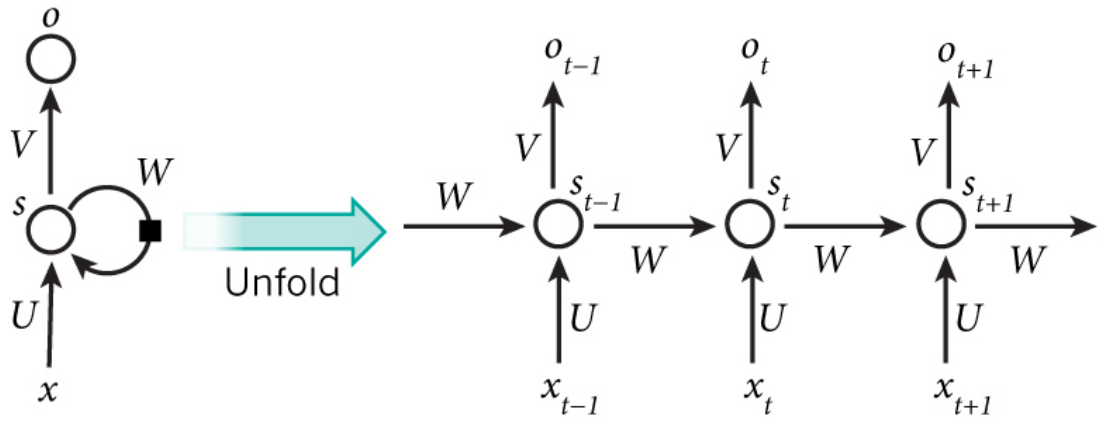


FIGURE 2.15: Recurrent Neural Network (RNN).

Fig. 2.15 shows an RNN loop, unfolded into a sequence, where W, U, and V are network parameters. x_t is the input vector at time-step t, which could be an word embedding (Section 2.2); s_t is hidden state at time-step t:

$$s_t = \mathcal{N}(U\ x_t + W\ s_{t-1}), \tag{2.15}$$

where $\mathcal{N}$ is non-linearity, such as sigmoid or $\tanh$. The initial state s_{-1} is often initialized to null vector. o_t is the output at step t:

$$o_t = \text{softmax}(V\ s_t), \tag{2.16}$$

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

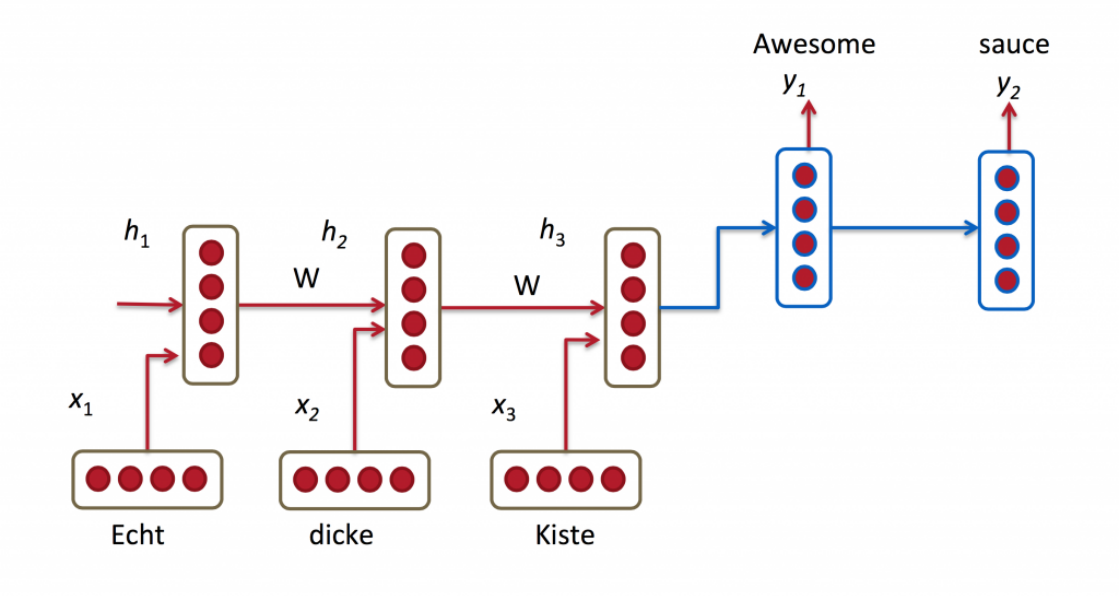


FIGURE 2.16: RNN for machine translation.

Fig. 2.16 shows an RNN for translating German to English.

2.4.4 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is purposely designed to be less prone to vanishing gradient problem. Also, it is more adept at retaining long-term memories in a sequence. These make it perfect fit for NLP applications due to intrinsically sequential nature of sentences and documents.

LSTM has three gates, namely input, output, and forget, that decide the content to incorporate, to output, and to forget, based on current input and memory state.

Fig. 2.17 depicts architecture of single LSTM cell. The same structure reiterates over all the time steps, sharing the parameters. LSTM is described by the following

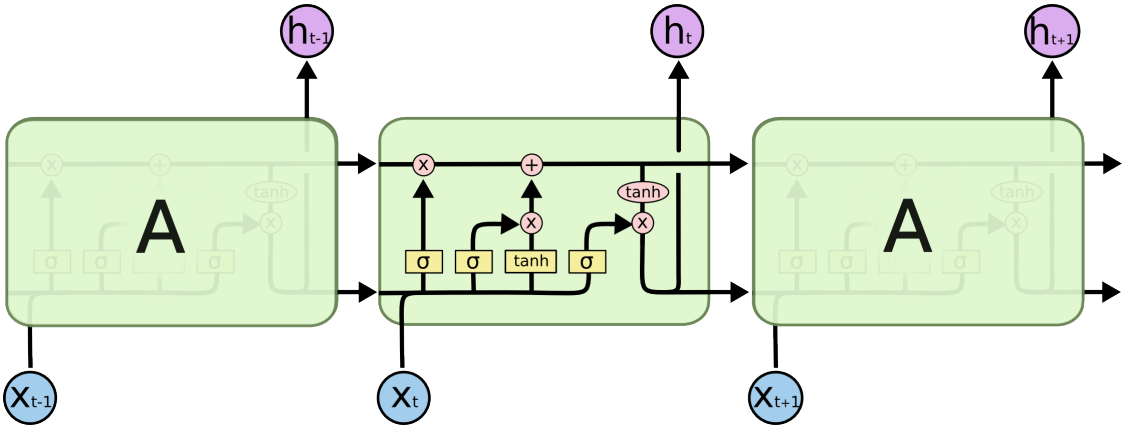


FIGURE 2.17: Long Short-Term Memory (LSTM).

equations:

$$i_t = \sigma(W_i\, x_t + U_i\, h_{t-1} + b_i), \tag{2.17}$$
$$o_t = \sigma(W_o\, x_t + U_o\, h_{t-1} + b_o), \tag{2.18}$$
$$f_t = \sigma(W_f\, x_t + U_f\, h_{t-1} + b_f), \tag{2.19}$$
$$C_{in} = \tanh(W_c\, x_t + U_c\, h_{t-1} + b_c), \tag{2.20}$$
$$C_t = i_t * C_{in} + f_t * C_{t-1}, \tag{2.21}$$
$$h_t = o_t * \tanh(C_t), \tag{2.22}$$

where i_t, o_t, and f_t represent input, output, and forget gate, respectively, at time-step t; C_{in} is the candidate state at time-step t; C_t is cell state at time-step t (memory); h_t is hidden output of LSTM cell at time-step t. The gates depend on the current input and output of previous cell, which means that it considers prior information with current information to make decision retention and incorporation. In Eq. (2.21), new memory is formed by purging part of existing memory and integrating part of new input x_t. In Eq. (2.22), LSTM cell passes part of the new memory as output. The output of final cell is often considered representation of the whole sequence. However, often all the outputs are pooled to obtain the sequence representation.

2.4.5 Gated Recurrent Unit (GRU)

Gated recurrent unit (GRU) (Chung et al., 2014), as depicted in Fig. 2.18, employs similar ideas as LSTM. However, it works with fewer parameters than LSTM. Also, it has only two gates as compared to three gates in LSTM. GRU can be described with the

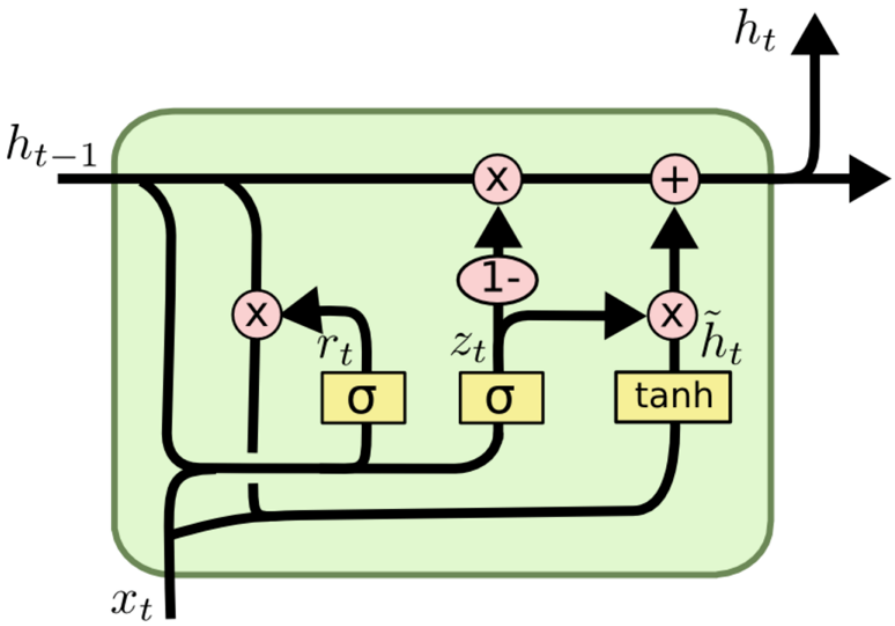


FIGURE 2.18: Single Gated Recurrent Unit (GRU) cell.

following:

$$z_t = \sigma(W_z\, x_t + U_z\, h_{t-1} + b_z), \tag{2.23}$$
$$r_t = \sigma(W_r\, x_t + U_r\, h_{t-1} + b_r), \tag{2.24}$$
$$\tilde{h}_t = \tanh(W_h\, x_t + U_h\, (r_t * h_{t-1}) + b_h), \tag{2.25}$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \tag{2.26}$$

where z_t and r_t are update and refresh gates, respectively; $\tilde{h}_t$ is candidate input information; h_t is the output at time-step t.

GRU is an approximation of LSTM. Compared to LSTM, GRU often performs better than LSTM on small datasets. However, with sufficient data LSTM performs better or comparable to GRU.

2.5 Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is an optimization algorithm that minimizes a differentiable function by iteratively updating its parameters by a controlled fraction of its gradients until a state of saturation is reached. Fig. 2.19 depicts the gradual descent to the absolute minima of a convex function. Algorithm 1 summarizes SGD.

Hyper-parameters like η and ϵ are to be supplied by the user. ϵ is usually quite small, like 0.0001. η (or learning-rate) is chosen with care to a small value like 0.001 and is refined across many SGD runs. Too small a learning-rate may not lead to a

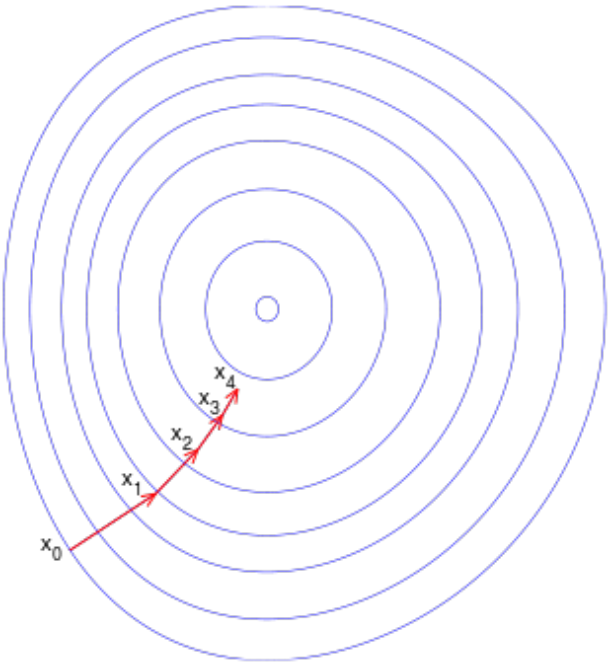


FIGURE 2.19: Gradient Descent for a convex function.

Algorithm 1 Stochastic Gradient Descent algorithm.

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

convergent state in a feasible amount of time. On the other hand, too large a η often leads to oscillation of J or worse yet missing the global minima.

For large neural networks, the gradients ($\nabla J(w^{(t-1)})$) are calculated using back-propagation algorithm, that employs chain rule to propagate error from loss (J) towards input.

2.6 Model Validation Techniques

2.6.1 Cross Validation

Cross-validation is a method of validating predictive models by generating one or more splits of the dataset containing a training and a test partition.

Stratified k-fold cross-validation is the most popular variant of cross-validation. Here, the dataset is randomly partitioned in to k nearly equal sized subsets, where it is equally

likely for a sample to belong to any of the k subsets. Each of the k subset is used exactly once for evaluation, while the remaining $k-1$ subsets are used for training. The results from each k test sets are aggregated for final evaluation.

Other variants include *leave one group out* where the test set contains samples with some feature exclusive to the test set.

2.7 Model Evaluation Techniques

The following methods evaluate the quality of the model predictions against expected output.

2.7.1 Evaluating Regression Quality

Mean Absolute Error — Mean absolute error (MAE) is defined as the average of absolute difference between expected (Y_i) and predicted ($\hat{Y}_i$) model output:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|, \tag{2.27}$$

where n is the number of samples.

Mean Squared Error — Mean squared error (MSE) is defined as the average of squared difference between expected (Y_i) and predicted ($\hat{Y}_i$) model output:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{2.28}$$

where n is the number of samples. MSE is also used as loss for training neural networks for regression problems due to being differentiable.

Pearson Correlation Coefficient — Pearson correlation coefficient (ρ) is computed to measure linear correlation between expected and predicted output, that ranges within $[-1, +1]$:

$$\rho_{Y,\hat{Y}} = \frac{\text{COV}(Y,\hat{Y})}{\sigma_Y \sigma_{\hat{Y}}}, \tag{2.29}$$

$$\text{COV}(Y,\hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y_i})(\hat{Y}_i - \bar{\hat{Y}_i}), \tag{2.30}$$

$$\sigma_Y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y_i})^2}, \sigma_{\hat{Y}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - \bar{\hat{Y}_i})^2}. \tag{2.31}$$

2.7.2 Evaluating Classification Techniques

Precision — Precision is a class-specific metric (say class c) which is defined as — the fraction of the samples predicted as class c that are predicted correctly as c. In terms of formula, it is

$$\text{Pr}_c = \frac{T_c}{T_c + F_c}, \tag{2.32}$$

where T_c and F_c are the number of samples that are correctly and falsely predicted as class c, respectively.

Recall — Recall is also a class-specific metric (say class c) which is defined as — the fraction of the samples known to be class c that are predicted correctly as c. The formula is

$$\text{Re}_c = \frac{T_c}{T_c + F_{\neg c}}, \tag{2.33}$$

where T_c is the number of samples correctly predicted as class c and $F_{\neg c}$ is the number of samples incorrectly predicted as other classes that are known to be c.

F-Score — A measure that combines precision and recall with harmonic mean:

$$\text{F}_c = \frac{2 \cdot \text{Pr}_c \cdot \text{Re}_c}{\text{Pr}_c + \text{Re}_c}. \tag{2.34}$$

This measure is especially relevant for imbalanced data where precision and recall significantly deviate from each other.

To determine overall F-Score macro, micro, or weighted average is taken over F-Scores of all the classes.

Accuracy — Accuracy is the fraction of all the samples that are predicted correctly:

$$\text{Accuracy} = \frac{\sum_{c \in C} T_c}{\sum_{c \in C} T_c + \sum_{c \in C} F_c}, \tag{2.35}$$

where C is the set of all classes.

Chapter 3

Variational Fusion for Multimodal Sentiment Analysis

3.1 Introduction

Multimodal sentiment analysis or affect detection is a fast-growing research area due to its strong and increasing demand in the industry. Thanks to the lucrative video journalism, promulgated by the likes of Facebook, YouTube, many professional reviewers are sharing their product reviews on video platforms. Even nowadays, major news channels are publishing their content on these platforms. Since, videos contain three channels or modalities of information — textual, visual, and acoustic — predictions made by using information from all these modalities tend to be more accurate compared to using single channel. As such, major organizations are interested in utilizing these freely available videos for market research, feedback gathering, customer relationship management, and more.

Multimodal fusion is a major component of any multimodal task, including multimodal sentiment analysis. Recent works on multimodal fusion (Poria et al., 2017; Zadeh et al., 2017, 2018c) have focused on encoding extracted unimodal representations into a single unified multimodal representation. However, our approach takes this one step further by reconstructing original unimodal representations. Our motivation is that, since, different modalities are manifestation of the state of mind, then we can assume that the fused representation should be representation of the state of mind as well. As such, if we can ensure mapping between fused representation and unimodal representations, then improvement of the quality of fused representation is reasonable to assume. We empirically show the validity of this supposition in this chapter.

We implement our strategy using variational autoencoder (VAE) (Kingma and Welling, 2014), which is composed of an encoder and a decoder. The encoder network produces a latent multimodal representation from the unimodal representations. Again, the decoder network decodes the latent multimodal representation into the original unimodal representations.

The rest of the chapter is organized as follows — Section 3.2 briefly discusses the recent works, Section 3.3 describes our approach, Section 3.4 states the experimental setup, Section 3.5 reports and interprets the results of our experiments, and finally Section 3.6 concludes this chapter.

3.2 Related Works

Lately, sentiment analysis (Cambria et al., 2017) has become a prevalent tool for extracting affective content out of large volume of social media content residing on Facebook, YouTube, blogs, and various other online platforms. There is growing interest in scientific community, leading to many exciting open challenges, as well as in the business world, due to the remarkable benefits to be had from financial forecasting (Xing et al., 2017) and political forecasting (Ebrahimi et al., 2017), user profiling (Mihalcea and Garimella, 2016), and more.

In emotion recognition, early works by De Silva et al. (1997) and L. S. Chen et al. (1998) demonstrated the superiority of fusion of acoustic and visual modalities over unimodal approaches. Both feature level (Kessous et al., 2010) and decision level (Schuller, 2011) fusion have been investigated.

As for textual modality, Rozgic et al. (2012) and Wollmer et al. (2013) were one of the first few who fused acoustic, visual, and text modalities for sentiment and emotion detection. Poria et al. (2015) employed CNN and multi-kernel learning for multimodal sentiment analysis. Further, Poria et al. (2017) used long short-term memory (LSTM) to achieve context-dependent multimodal fusion, where the surrounding utterances are taken into account for context.

Zadeh et al. (2017) used tensor outer-products to model intra- and inter-modal interactions for utterance-level fusion. Again, Zadeh et al. (2018a) used multi-view learning for utterance-level multimodal fusion. Further, Zadeh et al. (2018c) employed hybrid LSTM memory components to model intra-modal and cross-modal interactions.

3.3 Method

Humans have three major channels/modalities at their disposal to communicate their thoughts — textual (written text or speech), acoustic (pitch and other vocal properties), and visual (facial expression). Communicating through speech often leads to the usage of all three modalities, one substantiating the others. To fuse relevant information from all these multiple modalities existing methods employ some encoder to generate unified fused representation. However, in this chapter we present a method that further decodes the fused representation back to the original unimodal representations.

Firstly, utterance-level unimodal features are extracted independently (Section 3.3.1). These unimodal features are passed to an encoder network (Section 3.3.2) to generate fused representation. Further, the fused representation is decoded back to the unimodal representations to maximize the fidelity of the fused representation, with respect to the unimodal representations. This is basically an autoencoder setting. Specifically, we employ a variational autoencoder (VAE) (Kingma and Welling, 2014), as described in Fig. 3.1, where the sampled latent representation is used as the multimodal representation.

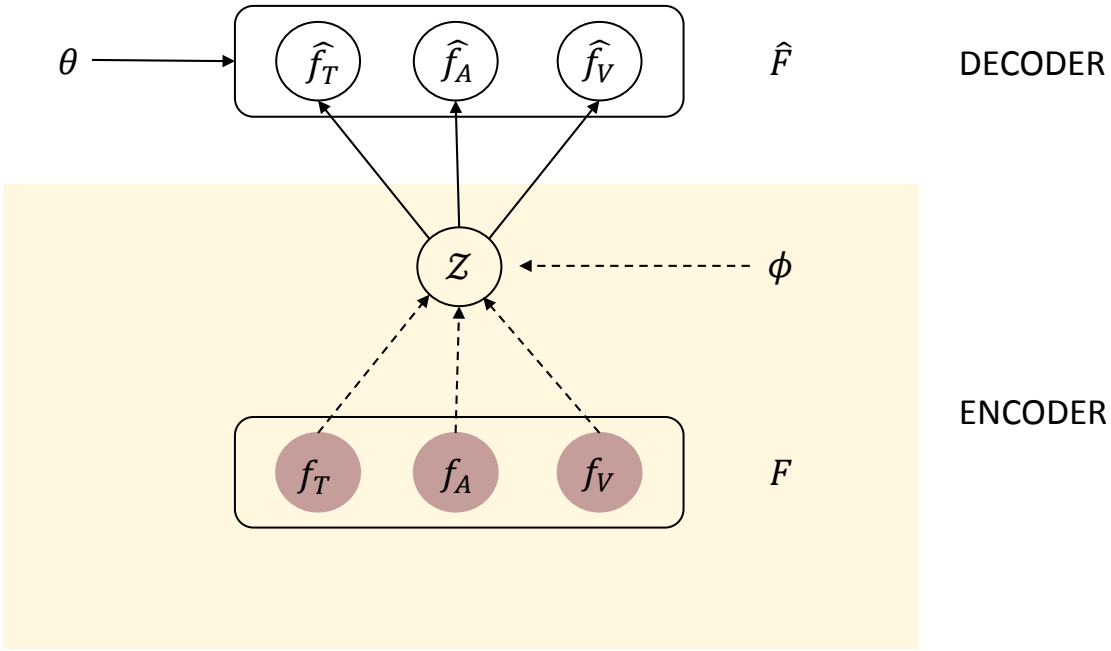


FIGURE 3.1: Graphical model of our multimodal fusion scheme.

3.3.1 Unimodal Feature Extraction

Textual (f_t), acoustic (f_a), and visual (f_v) features are extracted using CNN, 3D-CNN (Tran et al., 2015), and OpenSmile (Eyben and Schuller, 2015), respectively.

3.3.1.1 Textual Feature Extraction

Firstly, transcripts are obtained from the videos. We represent the words in each utterance with 300-dimensional word2vec vectors (Mikolov et al., 2013) (Section 2.2.2). Each utterance is padded or truncated to have exactly 50 words. Such utterances are fed to CNN- (Karpathy et al., 2014) based network (Fig. 3.2) to predict their sentiment labels.

The network consists of two consecutive convolution layers — the first layer has two filters of sizes 3 and 4, each having 50 feature maps; the second one has a filter of size 2 with 100 feature maps. Each convolution is followed by a max-pooling layer of size 2×2. The transition of the sentence through the convolution filters forces the network to learn abstract features. Moreover, with each subsequent layer the scope of the individual feature values expands further.

The output of the final max-pooling layer is fed to a dense layer of size 500 with ReLU (Teh and Hinton, 2001) activation and dropout, followed by another dense layer with softmax activation for classification. The output of the penultimate fully-connected layer is taken as the textual representation of the corresponding utterance.

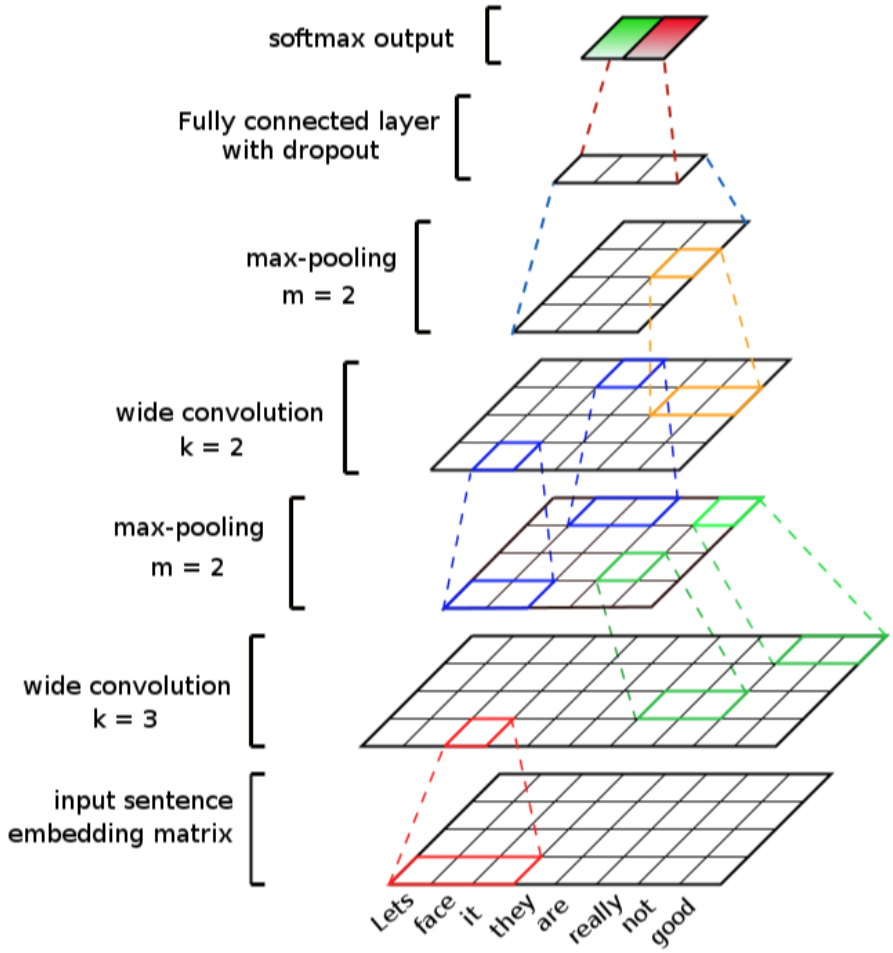


FIGURE 3.2: Text CNN for textual feature extraction.

3.3.1.2 Acoustic Feature Extraction

Using openSMILE (Eyben et al., 2010), we extract low level descriptors (LLD) — like pitch, voice intensity — and various statistical functionals — amplitude mean, arithmetic mean, root quadratic mean, standard deviation, flatness, skewness, kurtosis, quartiles, inter-quartile ranges, and linear regression slope — of the utterance-level audio clips.

Prior to extracting the aforementioned features, voiceless audio segments are removed using voice intensity threshold. Also, Z-standardization is applied for voice normalization.

OpenSMILE is utilized for both feature extraction and audio pre-processing, where input audio is sampled at 30 Hz frame rate, with 100 ms sliding window. "IS13-ComParE" configuration file is used for feature extraction, that results in total 6392 acoustic features per utterance.

3.3.1.3 Visual Feature Extraction

We employ 3D-CNN (Section 2.4.2) for utterance-level visual feature extraction. As a 3D extension of regular CNN, 3D-CNN not only captures frame-level information, but also inter-frame temporal dependencies that exist in a video clip.

Each video $V \in \mathbb{R}^{3\times f\times h\times w}$, is composed of f number of frames of size $h \times w$ with RGB pixels with three channels. We apply 3D convolutional filter $F_{conv} \in \mathbb{R}^{32\times 3\times 5\times 5\times 5}$, followed by max-pooling of size $3 \times 3 \times 3$ on each V. This is flattened and fed to a dense layer with 300 output size, which is fed to another dense layer with softmax activation for sentiment classification. The 300-dimensional output of the penultimate layer is taken as visual feature after training.

3.3.2 Encoder

The encoder network accepts utterance-level unimodal representations — $f_t \in \mathbb{R}^{D_t}$, $f_a \in \mathbb{R}^{D_a}$, and $f_v \in \mathbb{R}^{D_v}$ — and generates the latent multimodal representation $z \in \mathbb{R}^{D_z}$ by sampling from posterior distribution $p_\theta(z|F)$:

$$p_\theta(z|F) = \frac{p_\theta(F|z)p(z)}{p(F)}, \tag{3.1}$$

$$F = f_t \oplus f_a \oplus f_v, \tag{3.2}$$

$$p(F) = \int p_\theta(F|z)p(z)dz, \tag{3.3}$$

$$p(z) = \mathcal{N}(0, I), \tag{3.4}$$

where $p_\theta(F|z)$ represents the decoder network with parameter θ given standard normal prior of z.

Unfortunately, the true posterior $p_\theta(z|F)$ is analytically intractable due to RHS (right-hand side) of Eq. (3.3) being the same. As such, we approximate the posterior distribution by feeding F to a neural network of two dense layers that generates mean and covariance matrix of the approximate posterior distribution $q_\phi(z|F)$:

$$q_\phi(z|F) = \mathcal{N}(\mu_{enc}, \sigma_{enc}), \tag{3.5}$$

$$h_1 = \text{ReLU}(W_{h_1}F + b_{h_1}), \tag{3.6}$$

$$\mu_{enc} = W_\mu h_1 + b_\mu, \tag{3.7}$$

$$\sigma_{enc} = \text{softplus}(W_\sigma h_1 + b_\sigma), \tag{3.8}$$

where $F \in \mathbb{R}^{D_t+D_a+D_v}$, $W_{h_1} \in \mathbb{R}^{D_h\times(D_t+D_a+D_v)}$, $b_{h_1} \in \mathbb{R}^{D_h}$, $h_1 \in \mathbb{R}^{D_h}$, $W_{\{\mu,\sigma\}} \in \mathbb{R}^{D_z\times D_h}$, $b_{\{\mu,\sigma\}} \in \mathbb{R}^{D_z}$, $\mu_{enc} \in \mathbb{R}^{D_z}$, and $\sigma_{enc} \in \mathbb{R}^{D_z}$. It is point out that the non-diagonal elements of the covariance matrix of are set to zero to force the feature values in z be independent of each other. Hence, σ_{enc} is a vector instead of matrix. Eq. (3.20) in Section 3.3.5, approximates $p_\theta(z|F)$ into $q_\phi(z|F)$ during training.

A more sophisticated encoder, like TFN (Zadeh et al., 2017) and MFN (Zadeh et al., 2018a), could be adopted instead for posterior approximation. However, encoding the

unimodal representations is not the focus of this chapter, rather it is the reconstruction of the unimodal features from the multimodal representation.

Sampling Latent (Multimodal) Representation — The latent representation $z \sim q_\phi(z|F)$ is sampled from the approximate posterior using the reparameterization trick (Kingma and Welling, 2014) to allow backpropagation during training:

$$z = \mu_{enc} + \epsilon \odot \sigma_{enc}, \tag{3.9}$$
$$\epsilon \sim \mathcal{N}(0, I), \tag{3.10}$$

where $z \in \mathbb{R}^{D_z}$, $\epsilon \in \mathbb{R}^{D_z}$, and $\odot$ represents elementwise product. z is the multimodal representation.

3.3.3 Decoder

Unimodal representations ($\hat{F}$) are reconstructed by the decoder network from z (Eq. (3.9)):

$$h_3 = \text{softplus}(W_{h_3} z + b_{h_3}), \tag{3.11}$$
$$\hat{F} = W_{rec} h_3 + b_{rec}, \tag{3.12}$$

where $W_{h_3} \in \mathbb{R}^{D_h \times D_z}$, $b_{h_3} \in \mathbb{R}^{D_h}$, $W_{rec} \in \mathbb{R}^{(D_t+D_a+D_v)\times D_h}$, $b_{rec} \in \mathbb{R}^{(D_t+D_a+D_v)}$, $h_3 \in \mathbb{R}^{D_h}$, and $\hat{F} \in \mathbb{R}^{(D_t+D_a+D_v)}$.

Similar to the encoder network, decoder network construction is not the focus of this chapter. A more sophisticated decoder can be used.

3.3.4 Classification

We employed two different types of classification networks:

Context-Free Classifier (Logistic Regression (LR)) — The multimodal representation z is fed to a dense layer with softmax activation:

$$\mathcal{P} = \text{softmax}(W_{cls} z + b_{cls}), \tag{3.13}$$
$$\hat{y} = \underset{i}{\text{argmax}}\, \mathcal{P}[i], \tag{3.14}$$

where $W_{cls} \in \mathbb{R}^{C \times D_z}$, $b_{cls} \in \mathbb{R}^{C}$, $\mathcal{P} \in \mathbb{R}^{C}$ contains the class-probabilities, $\hat{y}$ is the predicted class, and C is the number of classes ($C = 2$ for MOSI dataset (Section 3.4.1)).

The output $\hat{y}$ does not depend on the neighboring utterances, ergo context-free classifier.

Context-Dependent Classifier (bc-LSTM (Poria et al., 2017)) — In order to infuse contextual information into the multimodal utterance representations (z_j; j is the index

of an utterance) within a video, z_j are fed to a bidirectional-LSTM (bi-LSTM) (Hochreiter and Schmidhuber, 1997), as per Poria et al. (2017). The output of the bi-LSTM is fed to a dense layer with softmax activation for sentiment classification:

$$Z = [z_1, z_2, \ldots, z_n], \tag{3.15}$$
$$H = \text{bi-LSTM}(Z), \tag{3.16}$$
$$H = [h_1, h_2, \ldots, h_n], \tag{3.17}$$
$$\mathcal{P}_j = \text{softmax}(W_{cls} h_j + b_{cls}), \tag{3.18}$$
$$\hat{y}_j = \underset{i}{\text{argmax}}\, \mathcal{P}_j[i], \tag{3.19}$$

where Z and H contain n constituent multimodal utterance representations without and with context information, respectively; $h_i \in \mathbb{R}^{2D_l}$, $W_{cls} \in \mathbb{R}^{C \times 2D_l}$, $b_{cls} \in \mathbb{R}^C$, $\mathcal{P}_j \in \mathbb{R}^C$ contains class-probabilities for utterance j, $\hat{y}_j$ is the predicted class for utterance j, and C is the number of classes (e.g. $C = 2$ for MOSI dataset (Section 3.4.1)).

3.3.5 Training

Latent Multimodal Representation Inference — As per Kingma and Welling (2014), the true posterior $p_\theta(z|F)$ is approximated into $q_\phi(z|F)$ by maximizing the evidence lower bound (ELBO):

$$\log p(F) \geq \underbrace{\mathbb{E}_{q_\phi(z|F)}[\log p_\theta(F|z)] - \text{KL}[q_\phi(z|F)||p(z)]}_{\text{ELBO}}, \tag{3.20}$$

The first term of ELBO, $\mathbb{E}_{q_\phi(z|F)}[\log p_\theta(F|z)]$, captures the reconstruction loss of input F. The second term, $\text{KL}[q_\phi(z|F)||p(z)]$, pushes the approximate posterior $q_\phi(z|F)$ close to the prior $p(z) = \mathcal{N}(0, I)$ by minimizing the KL-divergence between them.

Classification — The sentiment classifiers (Section 3.3.4) were trained by minimizing categorical cross-entropy (E) between expected and estimated class probabilities:

$$E = -\frac{1}{N} \sum_{i=1}^{N} \log \mathcal{P}_i[y_i], \tag{3.21}$$

where N is the number of samples, $\mathcal{P}_i$ contains the class probabilities for sample i, and y_i is the target class for sample i.

We used SGD-based Adam (Kingma and Ba, 2015) algorithm to train the parameters. The hyper-parameters $\{D_h, D_l\}$ and learning-rate are tuned using grid-search. The latent representation size D_z is set to 100.

3.4 Experimental Settings

We assess the multimodal representations (Eq. (3.9)), sampled from VAE, on two distinct classification scenarios (Section 3.3.4). Therefore, the two variants are named VAE+LR and VAE+bc-LSTM in Table 3.2.

3.4.1 Datasets

We assess our method on three distinct datasets (splits are shown on Table 3.1):

Dataset	Train	Test
CMU-MOSEI	16188	4614
IEMOCAP	5810	1623
CMU-MOSI	1447	752

TABLE 3.1: Utterance count in the train and test sets.

CMU-MOSI (Zadeh et al., 2016) dataset consists of English review videos on various topics by 89 people. Each constituent utterance in the videos are annotated with sentiment label (*positive* or *negative*). Ideally, our model should be able to generalize well, irrespective of the speaker. To this end, we keep the train and test split mutually exclusive in terms of speakers. Specifically, training and test split consists of 1,447 and 752 utterances, respectively.

CMU-MOSEI (Zadeh et al., 2018b) dataset consists of 22,676 utterances spread across 3,229 YouTube product and movie review videos by 1,000 unique creators. The training, validation, and test split contain 16,188, 1,874, and 4,614 utterances, respectively. Each utterance is annotated with *positive, negative*, or *neutral* sentiment label.

IEMOCAP (Busso et al., 2008) consists of dyadic conversations among 10 unique speakers. The constituent utterances of each conversation are annotated with one of the six emotion labels — *anger, happy, sad, neutral, excited,* and *frustrated*. The first 8 speakers from sessions one to four are exclusive to training set and the remaining 2 belong to the test set.

3.4.2 Baseline Methods

Logistic Regression (LR) — The concatenation of unimodal representations are classified using logistic regression, as in Section 3.3.4. Context from the neighboring utterances is not considered.

bc-LSTM (Poria et al., 2017) — The concatenation of unimodal representations is sequentially fed to the bc-LSTM sentiment classifier, as in Section 3.3.4. This is the state-of-the-art method.

TFN (Zadeh et al., 2017) — Intra- and inter-modal interactions are modeled using vector outer product. It does not use neighboring context information.

MFN (Zadeh et al., 2018a) — Multi-view learning is employed for modality fusion with memory content. It also does not use neighboring context information.

MARN (Zadeh et al., 2018c) — In this model the intra- and cross-modal interactions are modeled with hybrid LSTM memory component.

3.5 Results and Discussion

	Method	CMU-MOSI	CMU-MOSEI	IEMOCAP
	TFN	74.8	53.7	56.8
	MARN	74.5	53.2	54.2
	MFN	74.2	54.1	53.5
CF	LR	74.6	56.6	53.9
CF	VAE+LR	77.8	57.4	54.4
CD	bc-LSTM	75.1	56.8	57.7
CD	VAE+bc-LSTM	**80.4***	**58.8***	**59.6***

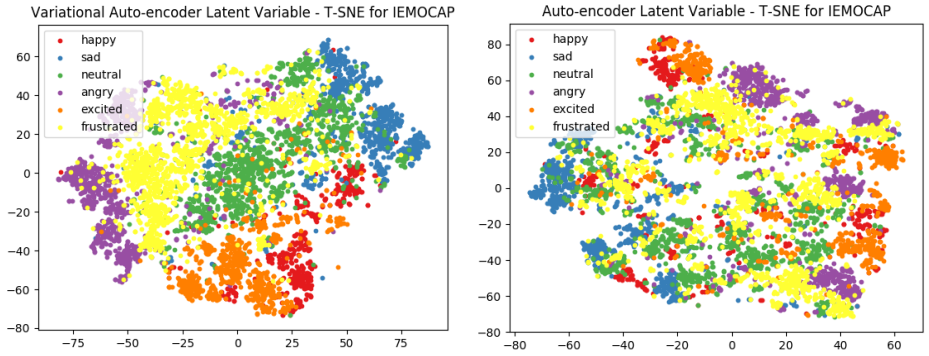
TABLE 3.2: Trimodal (acoustic, visual, and textual) performance (F1) of our method against the baselines (results on MOSI and IEMOCAP are based on the dataset split from Poria et al. (2017)); CF and CD stand for context-free and context-dependent models, respectively; * signifies statistically significant improvement ($p < 0.05$ with paired t-test) over bc-LSTM.

Following Table 3.2, our VAE-based methods — VAE+LR and VAE+bc-LSTM — surpass the corresponding concatenation-based fusions, with LR and bc-LSTM, consistently on all three datasets. In particular, our context-dependent model, namely VAE+bc-LSTM, outperforms the context-dependent state-of-the-art method bc-LSTM on all the datasets, by 3.1% on average. Similarly, our context-free model VAE+LR outperforms the other context-free models — MFN, MARN, TFN, and LR — on all datasets, by 1.5% on average. Further, VAE+bc-LSTM outperforms VAE+LR by 3.1% on average — this shows the importance of context in sentiment analysis.

Overall, it is reasonable to conclude that the superior fused representation generated by VAE, that retains enough unique information, leads to boosted classification performance. Also, it is important to point out that the fusion representation generator, VAE, is unsupervised. As such, the multimodal representation is label invariant.

3.5.1 VAE vs. AE Fusion

As opposed to VAE, we also used auto-encoder (AE) for multimodal fusion. This yielded inferior performance. We surmise this is due to relatively rigid nature of the latent states of AE due to no stochasticity. This is demonstrated in Fig. 3.3, where the individual classes in t-SNE scatter plot of AE are more fractured compared the same of VAE.



(A) (B)

FIGURE 3.3: (a) and (b) show t-SNE scatter-plots of VAE and AE multimodal representations, respectively, for IEMOCAP.

3.5.2 Case Study

Matching the predictions of our model to the baseline models have highlighted the fact that our model is more capable where non-textual cues are necessary for correct sentiment classification. For example, the utterance "*I still can't live on in six seven and five. It's not possible in Los Angeles. Housing is too expensive.*" is misclassified by bc-LSTM as *excited*, as compared to VAE+bc-LSTM that correctly classifies it as *angry*. We surmise that bc-LSTM was unable to distinguish the nuance between *angry* and *excited* from textual modality. However, due to informative and non-redundant multimodal representation from VAE, VAE+bc-LSTM could leverage information from visual modality. We observed similar trend overall, where our VAE-based methods correctly classify based on non-verbal cues as opposed to non-VAE baselines.

3.5.3 Error Analysis

We observed overall inability to recognize sarcasm by our models. For instance, *"No. I am just making myself fascinating for you."* is a sarcastic response to a question *"you going out somewhere, dear?"*. Unfortunately, VAE+bc-LSTM fails to correctly classify the emotion as *excited*, in contrast to the ground truth *angry*.

3.6 Conclusion

This chapter presents a VAE-based unsupervised multimodal fusion strategy. We empirically show that our method outperforms the SOTA by significant margin. We used simple encoder and decoder models, however. In the future, we mean to develop more sophisticated encoder (MFN, TFN) and decoder networks for improved performance.

Chapter 4

IARM: Inter-Aspect Relation Modeling for Aspect-Based Sentiment Analysis

4.1 Introduction

Sentiment analysis has become a very active research area due to its myriad of applications in e-commerce, customer relationship management (CRM), recommender systems, user-feedback gathering, etc. Companies are often interested in user opinion on various aspects of their products, rather than the entirety of it. This enables them to plan and focus on specific areas for improvement. Aspect-based sentiment analysis (ABSA) enables this aspect-specific analysis.

Reviewers usually express their opinion on various aspects of the products individually. For instance, "*Everything is so easy to use, Mac software is just so much simpler than Microsoft software.*" discusses three aspects, "*use*", "*Mac software*", and "*Microsoft software*", where the sentiment behind them are *positive*, *positive*, and *negative*, respectively. Naturally, there are two subtasks at hand — aspect extraction (Shu et al., 2017) and aspect sentiment polarity detection (Wang et al., 2016). In this chapter, we focus on the second subtask where each pre-extracted aspect is to be assigned appropriate sentiment label (*positive*, *negative*, or *neutral*).

Existing works on aspect polarity classification does not account for the effect of neighboring aspects on the target aspect. For example,"*The menu is very limited - I think we counted 4 or 5 entries.*" has two aspects — "*menu*" and "*entries*". Here,"*I think ... entries*" phrase does not completely convey the sentiment behind aspect "*entries*". However, when we consider the phrase connected to "*menu*", the sentiment behind "*entries*" become clear, to be *negative*. So, in retrospect, we can observe that aspect "*menu*" imposes its sentiment on aspect "*entries*". To account for such cases, our presented model considers the neighboring aspects during target-aspect classification.

Aspect classification in a sentence containing multiple aspects is inherently more difficult than a sentence containing only one aspect. This is because the ABSA classifier has to correctly associate every aspect with their corresponding sentiment-bearing phrase or word. For instance, the sentence "*Coffee is a better deal than overpriced cosi sandwiches*" contains two aspects — *"coffee"* and *"cosi sandwiches"*. Here, *"coffee"* and *"cosi sandwiches"* are opinionated by *"better"* and *"overpriced"*, respectively. We

empirically show that our model is able to make this associations better that the existing methods.

Often, the aspects in sentences containing conjunctions — and, not only, also, but, however, though, etc — influence each other by sharing their sentiment. For example, let's consider the sentence *"Food is usually very good, though I wonder about freshness of raw vegetables"* containing aspects *"food"* and *"raw vegetables"*. Clearly, *"raw vegetables"* does not have any explicit sentiment marker. However, in contrast to *"food"*, which has a clear sentiment marker "*good*", one can easily infer the sentiment of *"raw vegetables"* to be *negative*. This connection is made by the existence of "*though*" in the sentence. Hence, we hypothesize our model can capture such interaction among the aspects.

We perform the following steps to model inter-aspect dependencies:

1. As in Wang et al. (2016), we derive aspect-specific sentence representations for all constituent aspects by feeding aspect-concatenated word representations to gated recurrent unit (GRU) (Chung et al., 2014) and attention mechanism (Luong et al., 2015);

2. Then, the dependencies among the aspects are modeled using memory networks (Sukhbaatar et al., 2015), where the target-aspect-aware sentence representation is compared to the rest of the aspect-aware sentence representations;

3. The output of the memory networks is fed to a softmax classifier for final aspect sentiment classification.

In this chapter, we empirically show that these steps outperforms the state of the art (Ma et al., 2017) by 1.6% on average on two distinct domains — restaurant and laptop.

The rest of the chapter is organized as follows — Section 4.2 discusses related works; Section 4.3 describes the proposed method; Section 4.4 discusses the mentions the dataset, baselines, and experimental settings to evaluate our method; Section 4.5 shows the results of our experiments and interprets them; and Section 4.6 makes a concluding remark by mentioning the contributions and planned future work.

4.2 Related Works

Due to the current advent of sharing opinionated textual pieces over blogs, wikis, editorials on social media platforms, sentiment analysis has gained a massive traction from the research community. Targeted sentiment analysis requires solving two subtasks — aspect extraction (Poria et al., 2016; Shu et al., 2017) and aspect-based sentiment analysis (ABSA) (Ma et al., 2017; Wang et al., 2016).

Due to the performance and scalability of deep learning-based methods, recently it has been experiencing much interest and progress within NLP (Young et al., 2018). This has led to significant performance improvement in ABSA.

One of the first deep learning-based ABSA approach (Wang et al., 2016) generated aspect-aware sentence representation with aspect-concatenated word embeddings. This

aspect-aware representation is fed to softmax classifier for aspect-level sentiment classification. Tay et al. (2017) improved this by utilizing word-aspect association with circular correlation. Recently, X. Li et al. (2018) employed transformer networks for ABSA.

The state-of-the-art method Ma et al. (2017) leverages attention mechanism to model interaction between aspect and sentence.

From question-answering point of view C. Li et al. (2017) and Tang et al. (2016b) employed memory networks to solve ABSA. However, none of these methods try to model inter-aspect interactions.

4.3 Method

4.3.1 Problem Definition

Input — The input consists of a sentence S composed of L words w_i — $S = [w_1, w_2, \ldots, w_L]$ — and constituent M aspect-terms $A_1, A_2, \ldots, A_M$, where $A_i = [w_k, \ldots, w_{k+m-1}]$, $1 \leq k \leq L$, $0 < m \leq L - k + 1$.

Output — Sentiment label (*positive*, *negative*, and *neutral*) per aspect-term A_i.

4.3.2 Model

The main novelty of our method over the exiting methods in the literature is the usage of neighboring aspects within a sentence for target aspect classification. Our supposition is that the presented inter-aspect relation modeling (IARM) method[1] (Fig. 4.1) is able to find the dependencies between target and neighboring aspects, that filters out irrelevant information to the target aspect, resulting in improved classification performance.

4.3.2.1 Overview

IARM consists in the following three stages:

Input Representation — The constituent words in the input sentence and aspect terms are represented and replaced with GloVe embeddings (Pennington et al., 2014) (Section 2.2.3). Multi-word aspect-terms are represented with the mean of their containing word representations.

Aspect-Aware Sentence Representation (AASR) — To obtain Aspect-Aware Sentence Representation (AASR), we follow Wang et al. (2016), where the words within the given sentence are concatenated with the given aspect representation. This sentence with aspect information is fed to a gated recurrent unit (GRU)[2] to infuse context and

[1] Implementation available at http://github.com/senticnet/IARM

[2] LSTM (Hochreiter and Schmidhuber, 1997) gives similar performance with more parameters

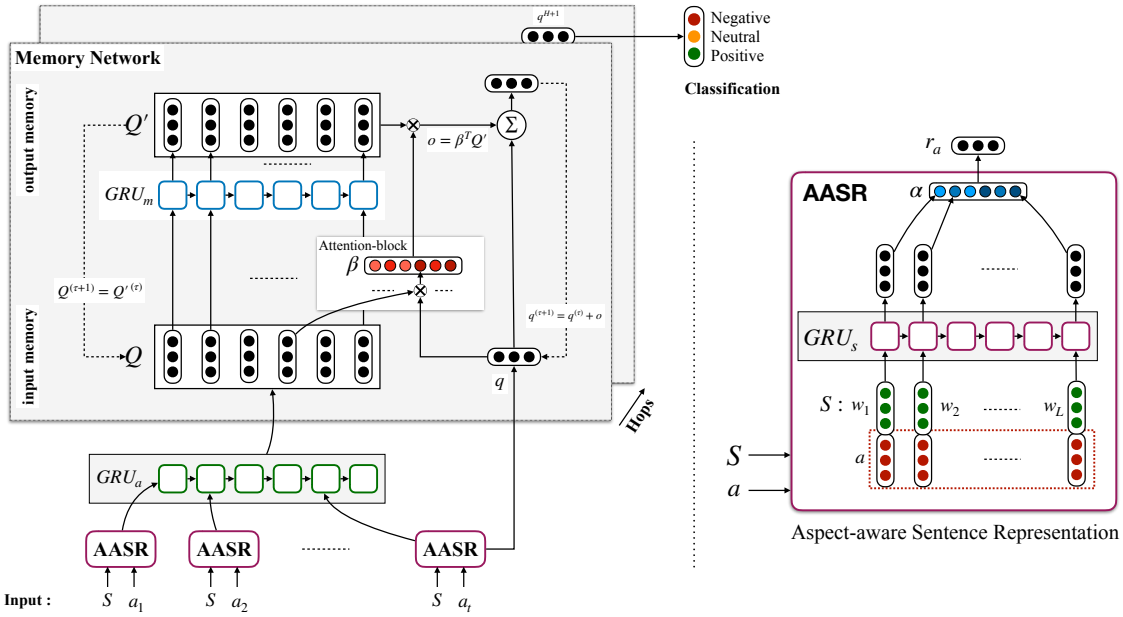


FIGURE 4.1: IARM architecture; AASR stands for *Aspect-Aware Sentence Representation*.

aspect information into the constituent words. Attention mechanism is applied to this context-rich word representations (output of GRU) to obtain a fixed-size AASR vector. This attention mechanism amplifies words and information relevant to the given aspect.

Inter-Aspect Dependency Modeling — To model inter-aspect dependency, we use memory networks (MemNet) (Sukhbaatar et al., 2015). The target AASR is passed as the query to the memory networks, where all AASRs within the sentence are stored in the memory slots. MemNet matches the target AASR with the rest of the AASRs and through attention mechanism inter-aspect dependency information is augmented to the target AASR. We assume that following several of this hops the refined target AASR would contain information only relevant to sentiment classification. Hence, this refined target AASR is fed to a dense layer with softmax activation for final sentiment classification.

4.3.2.2 Input Representation

The constituent words in sentence S, w_i are represented and replaced with GloVe embeddings (Pennington et al., 2014) (Section 2.2.3). That is $S \in \mathbb{R}^{L \times D}$, where $D = 300$ is the GloVe embedding size.

Likewise, aspect terms are represented as the mean of the constituent word embeddings — $a_i \in \mathbb{R}^D$ represents the i^{th} aspect term.

4.3.2.3 Aspect-Aware Sentence Representation

It is a reasonable assumption that sentiment of a specific aspect does not depend on all the words or phrases in the sentence. The stop words would be few of such words. As such, it is necessary to construct a sentence representation that is indicative of the aspect sentiment. To this end, following (Wang et al., 2016), we first concatenate aspect representation a_i to all the constituent words in sentence S:

$$S_{a_i} = [w_1 \oplus a_i, w_2 \oplus a_i, \ldots, w_L \oplus a_i] \in \mathbb{R}^{L \times 2D}. \tag{4.1}$$

Now, to infuse contextual information into the words, we pass S_{a_i} through a GRU, named GRU_s, of size D_s (Table 4.1) and obtain R_{a_i}. Generally, $GRU_{\cdot}$ is described as

$$z = \sigma(x_t U_{\cdot}^z + s_{t-1} W_{\cdot}^z), \tag{4.2}$$
$$r = \sigma(x_t U_{\cdot}^r + s_{t-1} W_{\cdot}^r), \tag{4.3}$$
$$h_t = \tanh(x_t U_{\cdot}^h + (s_{t-1} * r) W_{\cdot}^h), \tag{4.4}$$
$$s_t = (1 - z) * h_t + z * s_{t-1}, \tag{4.5}$$

where h_t and s_t are hidden output and cell state, respectively, at time t. We use shorthand for GRU operation as — $R_{a_i} = GRU_s(S_{a_i})$, where $R_{a_i} \in \mathbb{R}^{L \times D_s}$ and the GRU_s has trainable parameters — $U_s^z \in \mathbb{R}^{2D \times D_s}$, $W_s^z \in \mathbb{R}^{D_s \times D_s}$, $U_s^r \in \mathbb{R}^{2D \times D_s}$, $W_s^r \in \mathbb{R}^{D_s \times D_s}$, $U_s^h \in \mathbb{R}^{2D \times D_s}$, $W_s^h \in \mathbb{R}^{D_s \times D_s}$.

Finally, we utilize attention mechanism to summarize information relevant to the sentiment of aspect a_i, from context- and aspect-aware word representations in R_{a_i}:

$$z = R_{a_i} W_s + b_s, \tag{4.6}$$
$$\alpha = \text{softmax}(z), \tag{4.7}$$
$$r_{a_i} = \alpha^T R_{a_i}, \tag{4.8}$$

where $z = [z_1, z_2, \ldots, z_L] \in \mathbb{R}^{L \times 1}$, $\text{softmax}(x) = [e^{x_1} / \sum_j e^{x_j}, e^{x_2} / \sum_j e^{x_j}, \ldots]$, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_L] \in \mathbb{R}^{L \times 1}$, $r_{a_i} \in \mathbb{R}^{D_s}$, $W_s \in \mathbb{R}^{D_s \times 1}$, and b_s is a scalar. r_{a_i} is the final aspect-aware sentence representation (AASR).

4.3.2.4 Inter-Aspect Dependency Modeling

We employ a GRU, namely GRU_a, of size D_o (Table 4.1) to infuse information on neighboring aspects into all the AASRs r_{a_i}, which aids in dependency modeling among the aspects:

$$R = [r_{a_1}, r_{a_2}, \ldots, r_{a_M}] \in \mathbb{R}^{M \times D_s}, \tag{4.9}$$
$$Q = GRU_a(R), \tag{4.10}$$

where $Q \in \mathbb{R}^{M \times D_o}$ and GRU_a has the parameters — $U_a^z \in \mathbb{R}^{D_s \times D_o}$, $W_a^z \in \mathbb{R}^{D_o \times D_o}$, $U_a^r \in \mathbb{R}^{D_s \times D_o}$, $W_a^r \in \mathbb{R}^{D_o \times D_o}$, $U_a^h \in \mathbb{R}^{D_s \times D_o}$, $W_a^h \in \mathbb{R}^{D_o \times D_o}$.

However, to refine target-aspect-aware sentence representation, namely r_{a_t} (t is the index of target aspect that is to be classified), memory networks (MemNet) (Sukhbaatar et al., 2015) is employed. Thus, we transform r_{a_t} into query (q) of the MemNet using a dense layer:

$$q = \tanh(r_{a_t} W_T + b_T), \tag{4.11}$$

where $q \in \mathbb{R}^{D_o}$, $W_T \in \mathbb{R}^{D_s \times D_o}$, and $b_T \in \mathbb{R}^{D_o}$.

Input Memory Representation — All individual AASRs are contained within the rows of input memory Q. Following Weston et al. (2014), we use attention mechanism to read from Q, where each memory slot (row) in Q is compared to the query q using inner product:

$$z = qQ^T, \tag{4.12}$$
$$\beta = \textit{softmax}(z), \tag{4.13}$$

where $z = [z_1, z_2, \ldots, z_M] \in \mathbb{R}^{M \times 1}$, $\beta = [\beta_1, \beta_2, \ldots, \beta_M] \in \mathbb{R}^{M \times 1}$. Attention score β_i represents the dependency between target aspect a_t and aspect a_i.

Output Memory Representation — The output memory Q' is a refinement of the input memory Q. Q is passed through GRU_m of size D_o, which further improves inter-aspect dependencies:

$$Q' = GRU_m(Q), \tag{4.14}$$

where GRU_m has parameters — $U_m^z \in \mathbb{R}^{D_o \times D_o}$, $W_m^z \in \mathbb{R}^{D_o \times D_o}$, $U_m^r \in \mathbb{R}^{D_o \times D_o}$, $W_m^r \in \mathbb{R}^{D_o \times D_o}$, $U_m^h \in \mathbb{R}^{D_o \times D_o}$, $W_m^h \in \mathbb{R}^{D_o \times D_o}$.

The target AASR is refined with response vector o, which contains the refined inter-aspect dependency information related to the target aspect. o is constructed by pooling over output memory Q' with inter-aspect dependency measure β:

$$o = \beta^T Q', \tag{4.15}$$

where $o \in \mathbb{R}^{D_o}$.

Final Classification (Single Hop) — For single MemNet hop, update information o is added to the query q and fed to a dense layer with softmax activation for aspect-sentiment classification:

$$\mathcal{P} = \text{softmax}((q + o)W_{smax} + b_{smax}), \tag{4.16}$$
$$\hat{y} = \underset{i}{\text{argmax}}(\mathcal{P}[i]), \tag{4.17}$$

where $W_{smax} \in \mathbb{R}^{D_o \times C}$, $b_{smax} \in \mathbb{R}^C$, $C = 3$ is the number of sentiment labels, and $\hat{y}$ is the predicted sentiment label (0 for *negative*, 1 for *positive*, and 2 for *neutral*).

Final Classification (Multiple Hops) — Multiple passes of target AASR q is achieved with multiple hops of MemNet. Each of total H (Table 4.1) hops is defined as follows:

- Target ASSR $q^{(\tau)}$ at the end of hop τ is updated as

$$q^{(\tau+1)} = q^{(\tau)} + o. \tag{4.18}$$

- Output memory of hop τ, namely $Q'^{(\tau)}$, is used as the input memory of the next hop $\tau + 1$:

$$Q^{(\tau+1)} = Q'^{(\tau)}. \tag{4.19}$$

$q^{(H)}$ is the final refined target AASR after H hops. This is fed to a dense layer with softmax activation for sentiment classification:

$$\mathcal{P} = \text{softmax}(q^{(H+1)} W_{smax} + b_{smax}), \tag{4.20}$$

$$\hat{y} = \underset{i}{\text{argmax}}(\mathcal{P}[i]), \tag{4.21}$$

where $W_{smax} \in \mathbb{R}^{D_o \times C}$, $b_{smax} \in \mathbb{R}^C$, and $\hat{y}$ is the predicted sentiment label (0 for *negative*, 1 for *positive*, and 2 for *neutral*). Algorithm 2 summarizes this method.

4.3.3 Training

The presented network is trained for 30 epochs on average with categorical cross-entropy added to L2-regularizer as loss (L):

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=0}^{C-1} y_{ik} \log \mathcal{P}[k] + \lambda \left\| \theta \right\|_2, \tag{4.22}$$

where N represents the number of samples, λ is the regularization weight (we set it to 10^{-4}),

$$y_{ik} = \begin{cases} 1, & \text{label of sample } i \text{ is } k, \\ 0, & \text{otherwise}, \end{cases} \tag{4.23}$$

and θ contains the trainable parameters, where

$$\theta = \{U^z_{\{s,a,m\}}, W^z_{\{s,a,m\}}, U^r_{\{s,a,m\}}, W^r_{\{s,a,m\}}, U^h_{\{s,a,m\}}, W^h_{\{s,a,m\}}, W_s, b_s, W_T, b_T, W_{smax}, b_{smax}\}.$$

We use SGD-based ADAM algorithm (Kingma and Ba, 2015) with learning-rate 0.001 for training.

Algorithm 2 IARM algorithm

procedure TRAINANDTESTMODEL(U, V) ▷ U = train set, V = test set

Aspect-aware sentence representation (AASR) extraction:
for `i:[1,`M`]` **do** ▷ generate AASR for all the aspects in the sentence
$r_{a_i} \leftarrow AASR(S, a_i)$

Query generation:
$q \leftarrow FCLayer(r_{a_t})$ ▷ Transform the target AASR into query of MemNet

Memory networks (MemNet):
$Q \leftarrow GRU_a([r_{a_1}, r_{a_2}, \ldots, r_{a_M}])$ ▷ initial input memory
$Q' \leftarrow GRU_m(Q)$ ▷ initial output memory

for `i:[1,`H`]` **do** ▷ memory network hops
$z \leftarrow qQ^T$ ▷ match with target aspect
$\beta \leftarrow softmax(z)$
$o \leftarrow \beta^T Q'$ ▷ response vector
$Q \leftarrow Q'$ ▷ input memory for the next hop
$q \leftarrow q + o$ ▷ update target AASR (query)

Classification:
$\hat{y} = \underset{j}{\text{argmax}}(softmax(q)[j])$ ▷ softmax classification
$TestModel(V)$

procedure AASR(S,a) ▷ generation of aspect-aware sentence representation
$R_a \leftarrow GRU_s([w_1 \oplus a, w_2 \oplus a, \ldots, w_L \oplus a])$ ▷ $S = [w_1, w_2, \ldots, w_L]$
$z \leftarrow FCLayer(R_a)$
$\alpha \leftarrow softmax(z)$
$r_a \leftarrow \alpha^T R_a$
return r_a

procedure TESTMODEL(V)
V is fed to the learnt model, as during the training, for classification. The trainable parameters (θ) are mentioned in Section 4.3.3.

Hyper-Parameters — Grid search was used in our experiments for hyper-parameter optimization. Optimal hyper-parameters are enlisted in Table 4.1.

4.4 Experiments

4.4.1 Dataset Details

We used SemEval-2014 ABSA dataset[3], containing samples from restaurant and laptop domains, to evaluate our method. Distribution of labels is shown in Table 4.2. Further,

[3] http://alt.qcri.org/semeval2014/task4

Hyper-Parameter	Restaurant	Laptop
D_s	300	400
D_o	350	400
Hop Count	3	10

TABLE 4.1: Optimal hyper-parameters.

Table 4.3 presents the distribution of sentences containing single aspect and multiple aspects.

Domain	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	805	196	633	196
Laptop	987	341	866	128	460	169

TABLE 4.2: Count of the samples by class labels in SemEval 2014 dataset.

Domain	Train		Test	
	SA	MA	SA	MA
Restaurant	1007	2595	285	835
Laptop	917	1396	259	379

TABLE 4.3: Count of the samples by the appearance of single aspect/multiple aspects in the source sentence in SemEval 2014; SA and MA stand for Single Aspect and Multiple Aspects, respectively.

4.4.2 Baseline Methods

Our method is compared with the following methods from the literature:

LSTM — This baseline method feeds the sentence to an LSTM to infuse contextual information into the constituent words. Mean pooling over the output of the LSTM is taken to obtain the sentence representation. This sentence representation is fed to a dense layer with softmax activation for classification. This baseline shows the impact of the absence of aspect information on the classification performance.

TD-LSTM — Tang et al. (2016a) feeds the words, concatenated with the target aspect-term, appearing before and after the target aspect term to two distinct LSTMs. Mean pooling over the outputs of the two LSTM are concatenated and sent to a dense layer with softmax activation for sentiment classification.

AE-LSTM — Wang et al. (2016) feeds the sentence to an LSTM to infuse contextual information into the containing words. Target-aspect representation is concatenated to each of the output vectors of the LSTM. Attention pooling is applied to these concatenated vectors to obtain intermediate aspect representation. This is passed through an affine transformation and concatenated with the final output of the LSTM. This fed to a dense layer with softmax activation for sentiment classification.

ATAE-LSTM — ATAE-LSTM (Wang et al., 2016) follows the same steps as AE-LSTM. However, the only distinction being the LSTM accepts concatenation of target aspect-term representation and word representations.

IAN (SOTA) — Ma et al. (2017) feeds the target aspect and the context sentence to two different LSTMs. Output of each LSTM are max-pooled to a aspect and sentence representation vectors. Now, this aspect representation vector is used for attention pooling over the sentence LSTM output and vice versa. Output of these two attention pooling are concatenated and fed to a dense layer with softmax activation for aspect-sentiment classification.

4.4.3 Experimental Settings

The following three types of experiments were conducted to compare IARM with the baseline methods:

Overall Comparison — For each individual domain, we compare IARM against the baseline methods.

Single-Aspect and Multi-Aspect Scenarios — The trained IARM model is evaluated independently on test samples having sentences with single and multiple aspects. The same is done for the state-of-the-art IAN.

Cross-Domain Evaluation — The IARM model trained with the training samples of one domain is evaluated with the test samples of the other domain. The same is done for IAN also.

4.5 Results and Discussion

Overall Comparison — Table 4.4 shows that IARM surpasses the state-of-the-art IAN and other baseline methods on both domains — by 1.7% and 1.4% on laptop and restaurant domains, respectively, over IAN. This demonstrates that the consideration of neighboring aspect information using MemNet leads to improvement in the target-aspect sentiment classification.

Model	Domain	
	Restaurant	Laptop
Majority Classifier	53.4	65.0
LSTM	74.3	66.5
TD-LSTM	75.6	68.1
AE-LSTM	76.2	68.9
ATAE-LSTM	77.2	68.7
IAN (SOTA)	78.6	72.1
IARM	**80.0**	**73.8**

TABLE 4.4: Domain-wise accuracy (%) of the discussed models; best performance for each domain is indicated with bold font.

Single-Aspect and Multi-Aspect Scenarios — According to Table 4.5, IARM outperforms the SOTA model IAN on both single-aspect and multi-aspect cases on both domains. Both models are expected to perform better on multi-aspect case than single-aspect case due to the higher sample count of multi-aspect case in both domains (Table 4.3). However, IAN performs better on single-aspect scenario than multi-aspect scenario. This is indicative of the fact that the presence of multiple aspects in a single sentence has confounding effect on IAN, leading to inferior performance on those aspects than IARM.

Model	Restaurant		Laptop	
	SA	MA	SA	MA
IAN (SOTA)	75.4	77.7	72.5	71.6
IARM	**78.6**	**80.48**	**73.4**	**74.1**

TABLE 4.5: Accuracy (%) of the models for single aspect and multi aspect scenario; SA and MA stand for Single Aspect and Multiple Aspects, respectively.

Cross-Domain Evaluation — According to Table 4.6, the SOTA IAN is beaten by IARM on both cross-domain cases. This illustrates the capability of IARM to learn domain-invariant features from the training data.

Model	Rest $\rightarrow$ Lap	Lap $\rightarrow$ Rest
IAN (SOTA)	64.6	72.0
IARM	**66.7**	**74.0**

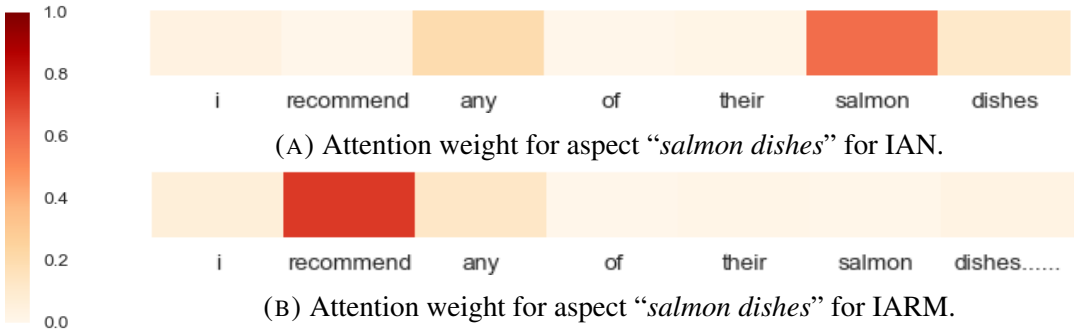
TABLE 4.6: Accuracy (%) on cross-domain scenarios; Rest: Restaurant domain, Lap: Laptop domain; A $\rightarrow$ B represents that the model is trained on the train-set of domain A and tested on the test-set of domain B.

4.5.1 Case Study

We study, interpret, and compare the functioning of IARM and IAN on single-aspect and multi-aspect cases with samples from the dataset.

Single-Aspect Case — Table 4.5 shows that IARM beats IAN on single-aspect case. We demonstrate this with a sample — "*I recommend any of their salmon dishes.....*" containing aspect "*salmon dishes*" with *positive* ground sentiment; IAN misclassifies this aspect as *negative* due to its attention on the wrong context word "*salmon*" without sentimental intensity, as depicted in Fig. 4.2a.

IARM, however, pays attention to the correct context word "*recommend*", bearing *positive* sentiment, as depicted in the visualization of α attention (Eq. (4.7)) in Fig. 4.2b. Hence, correct sentiment label is predicted.



(A) Attention weight for aspect "*salmon dishes*" for IAN.

(B) Attention weight for aspect "*salmon dishes*" for IARM.

FIGURE 4.2: Attention weights for IAN and IARM for "*I recommend any of their salmon dishes*".

Multi-Aspect Case — Following Table 4.5, IARM beats IAN on multi-aspect case as well. We posit that the existence of multiple aspects in context sentence has a confounding effect on IAN, specifically, regarding the association between aspect-term and its respective sentiment carrying word. Let us consider this sample — "*Coffee is a better deal than overpriced cosi sandwiches*" with aspect-terms "*coffee*" and "*cosi sandwiches*". IAN is unable to make the connection between aspect "*cosi sandwiches*" and its sentiment-bearer "*overpriced*". Rather, it erroneously connects "*cosi sandwiches*" to "*better*", which is sentiment bearer for the other aspect "*coffee*". This is visible in the visualization of attention of IAN Fig. 4.3a. As such, IAN misclassifies the sentiment of "*cosi sandwiches*" as *positive*.

On the other hand, IARM utilizes word-level aspect-aware attention (α) and MemNet to disambiguate the connection between aspect and its corresponding sentiment-bearing term. This is achieved by the repeated matching of target aspect with the neighboring aspects via MemNet. This ability of IARM is illustrated by the α attention visualizations in Fig. 4.3b and Fig. 4.3c, where the network correctly makes the association between the aspect-terms and its corresponding sentiment bearer, leading to accurate prediction.

(A) Attention weights for aspect "*cosi sandwiches*" for IAN.

(B) Attention weights for aspect "*cosi sandwiches*" for IARM.

(C) Attention weights for aspect "*coffee*" for IARM.

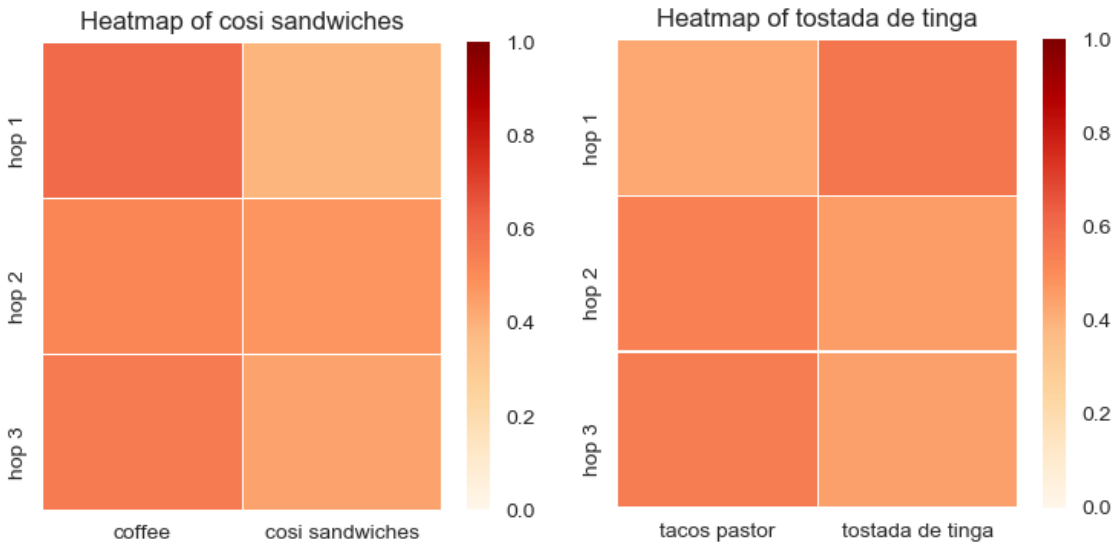
FIGURE 4.3: Attention weights for IAN and IARM for the sentence "*Coffee is a better deal than overpriced cosi sandwiches*".

The incorporation of information from neighboring aspects is done in Eq. (4.18). The amount of incorporation is calculated by the β attention weights in Eq. (4.13). The visualization of β attention (Fig. 4.4a) shows this phenomenon — the information from aspect "*coffee*" is being channeled into to the target AASR of "*cosi sandwiches*" for three hops. It can be supposed this information pertains to the sentiment bearer of aspect "*coffee*" — "*better*". "*better*" provokes a comparison between two aspects. As such, it induces the first aspect, "*coffee*", having *positive* sentiment and other one aspect, "*cosi sandwiches*", having *negative* sentiment. Lack of such modeling in IAN leads to misassociation, which is followed by misclassification.

Often conjunction has major role in structure and semantics of the sentence. As an example, "*my favs here are the tacos pastor and the tostada de tinga*" contains two aspects *"tacos pastor"* and *"tostada de tinga"* that are connected with conjunction *"and"*. Hence, they share a common sentiment-bearing term "*favs*". Fig. 4.4b illustrates how IARM is capable of mining such patterns — by exchanging information between aspects. Similar case appears in Fig. 4.5, where aspects *"atmosphere"* and *"service"* share *"good"* as sentiment marker, caused by the presence of *"and"*.

4.5.2 Error Analysis

We found that IARM misclassifies in certain scenarios. For instance, the aspect "*choices of sauces*" is misclassified as *neutral* in the sentence "*They bring a sauce cart up to your table and offer you 7 or 8 choices of sauces for your steak (I tried them ALL).*". We suppose this is caused by the misinterpretation of the sentiment-bearing phrase "*7 or 8 choices of sauces*", as the quantity seven or eight is a relative term. The Understanding of such cases comes from commonsense. Hence, we believe that addition of commonsense knowledge to our model would help it correctly classify such cases.



(A) MemNet β attention weights for sentence "*Coffee is a better deal than overpriced cosi sandwiches.*".

(B) MemNet β attention weights for sentence "*my favs here are the tacos pastor and the tostada de tinga.*".

FIGURE 4.4: MemNet β attention weights for IARM.

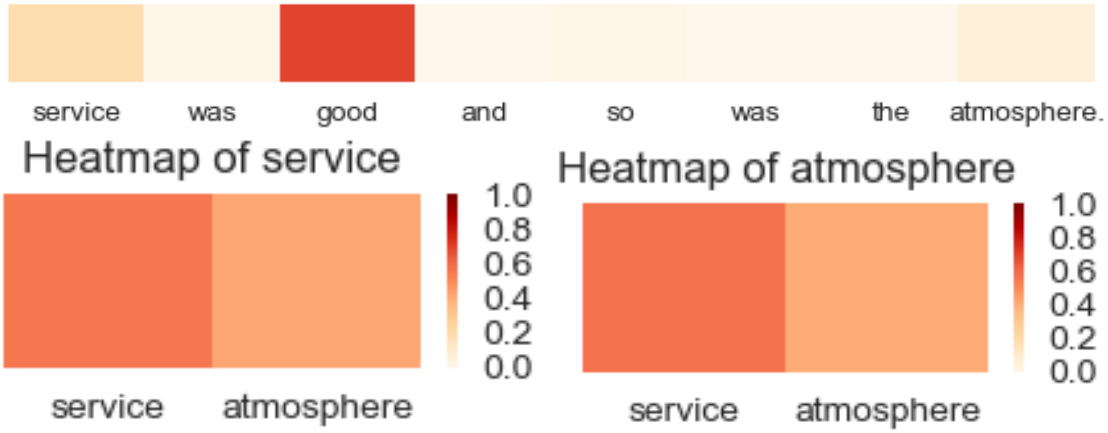


FIGURE 4.5: MemNet β attention weights for the sentence "*service was good and so was the atmosphere*".

Another similar case where IARM fails is while classifying the aspect "*breads*" within sentence "*Try the homemade breads.*". Our model fails to interpret the corresponding sentiment-bearing term "*try*" as *positive*, due to its default sentiment being *neutral*. Thus, it misclassifies the aspect as *neutral*. We suppose that the inclusion of commonsense knowledge would aid IARM in making the correct prediction.

4.5.3 Hop-Performance Relation

Fig. 4.6 shows how the hop count influences the performance of IARM. We observed that three and ten are the best hop count for restaurant and laptop domain, respectively. Also, it is observable that the plot of restaurant domain is smoother than the plot of laptop domain. We suppose that higher count of restaurant samples (Table 4.2) has led to this relatively smoother plot.

Another interesting observation is the overall declining trend of performance with the increase of hops, for restaurant domain, with two peaks at hop 3 and 10. This could be indicative of cyclic nature MemNet, the peaks denoting the start and end of a cycle. Laptop domain, however, exhibits a rather zig-zag pattern than a cyclic trend.

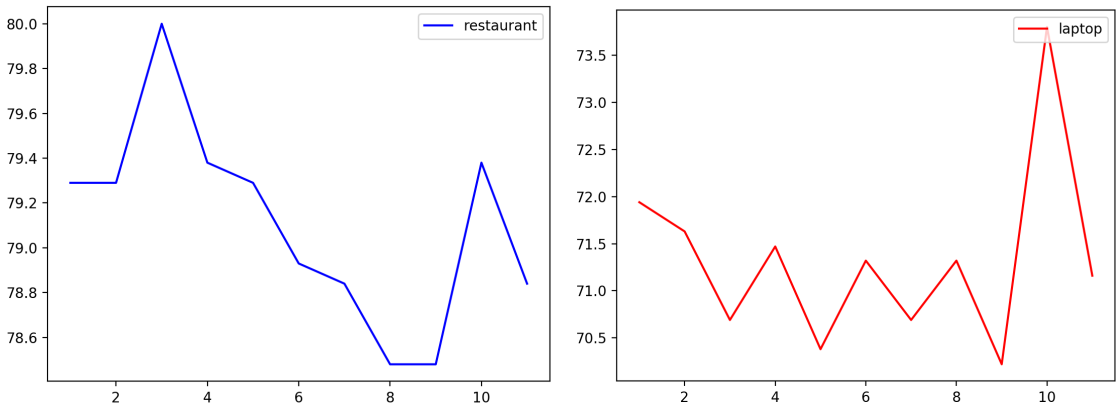


FIGURE 4.6: Hop-Accuracy plot for both domains.

4.6 Conclusion

This chapter has described a novel framework for aspect-based sentiment analysis, namely IARM. IARM exploits the dependency among constituent aspects within a sentence, using recurrent neural network and memory network, to improve their sentiment classification performance. We empirically verify this claim. Specifically, our model surpasses the state of the art by 1.6% overall, on two domains. That said, there still exist various paths for further improvement — improvement of aspect-aware sentence representations, improvement of aspect matching and dependency mining schemes, and more.

Chapter 5

DialogueRNN: An Attentive RNN for Emotion Recognition in Conversations

5.1 Introduction

By the virtue of internet and proliferation of internet-enabled devices, like smartphones, tablets, laptops, communication and discussion over various topics through the means of internet has exploded over last few decades. These discussions are often conversational in nature, facilitated by platforms like Facebook, Twitter, YouTube, etc. Various parties are looking to make sense of this copious amount of conversational data to estimate public perception of their products, service, policies, or any subject matter. This estimations are often performed in terms of emotions. To this end, in this chapter, we discuss an RNN-based emotion recognition in conversation (ERC) method that could attend to these needs.

Existing works in the literature, even the SOTA (Hazarika et al., 2018), mostly ignore the importance of individual parties. Especially, the models are not cognizant to the speaker of the target utterance and history of the speaker within the conversation. On the other hand, our model (DialogueRNN) strives to overcome these limitations by profiling individual parties during the flow of the conversation. We model the emotion of target utterance as a function of three factors — its speaker, its global context defined by preceding utterances, and its emotional context. As illustrated in Fig. 5.1, the role of preceding utterances is crucial in determining the context and in turn emotion of the current utterance. Inclusion of these factors into our model has significantly outperformed the state of the art (Table 5.3). Besides, our method is capable of working on multi-party scenario, in a scalable fashion, unlike the SOTA where the number of parameter increases linearly with the number of parties.

The presented model in this chapter, DialogueRNN, utilizes three GRUs (Chung et al., 2014) — namely speaker GRU, global GRU, and emotion GRU. The inbound utterance is passed to global GRU and speaker GRU to update the global context and speaker state, respectively. To provide global GRU speaker context, we also feed the current speaker state to global GRU. We believe that pooling these global GRU outputs would provide contextual information from the preceding utterances by various participants. Again, the speaker state is updated using speaker GRU, with the pooled global context

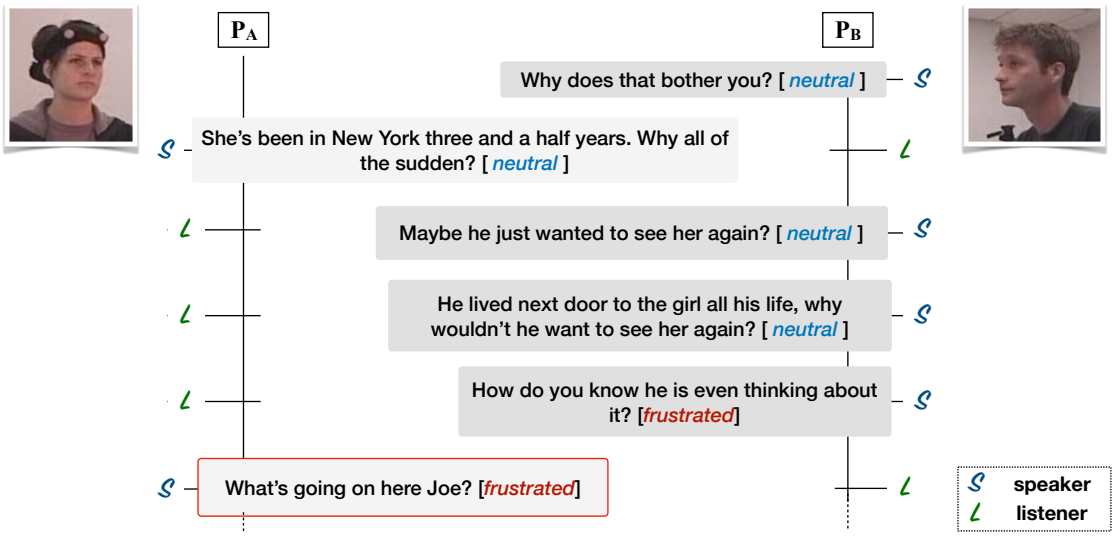


FIGURE 5.1: Illustration of a dialogue where P_A's emotion is directly influenced by the behavior of P_B.

to provide reference to the inbound utterance. This updated speaker state is fed to emotion GRU produce emotion representation of the inbound utterance. Emotion GRU also considers the emotion representation of the previous utterance as context and this might also aid in predicting possible emotion shift. Finally, this new emotion representation is fed to a softmax classifiers for emotion prediction.

Emotion GRU and global GRU model inter-party relationship. In contrast, party GRU profiles the parties individually along the conversation flow. DialogueRNN assembles these three GRUs in a specific manner that results in a custom RNN, that beats the state-of-the-art emotion classifiers (Hazarika et al., 2018; Poria et al., 2017), possibly due to conversation-specific improved context representation.

The remaining chapter is structured as — Section 5.2 mentions related works; Section 5.3 discusses our approach in details; Section 5.4 describes the experimental settings; Section 5.5 presents the results and analyses of our experiments; and finally, Section 5.6 makes a concluding remark to finish this chapter.

5.2 Related Works

Emotion detection has experienced cross-disciplinary interest due to its application in various fields such as cognitive science, psychology, natural language processing, and more (Picard, 2010). Initially, the work of Ekman (1993) established correlation between emotion and facial cues, which inspired the idea of multimodal emotion recognition. Later, Datcu and Rothkrantz (2008) blended audio with visual information for multimodal emotion recognition. One of the earliest works on emotion recognition in the context of NLP was done by Alm et al. (2005), that introduced text as a modality. Text-based emotion recognition was further advanced by Strapparava and Mihalcea

(2010). Contextual information was utilized in the work of Wöllmer et al. (2010) for multimodal emotion recognition. In the recent years, deep learning-based methods were employed for multimodal emotion recognition (M. Chen et al., 2017; Poria et al., 2017; Zadeh et al., 2018a,d).

Since, humans often express their emotions through conversations, it necessary to understand the dynamics of conversations to understand human emotion. Ruusuvuori (2013) argues the influence of emotion in conversations. Richards et al. (2003) suggests that emotion dynamics in conversations is inter-personal in nature. This motivated us to model inter-personal interactions in our method. Also, to adopt the sequential nature of conversations, we employ recurrent connection, following Poria et al. (2017).

Thanks to the successful application of memory networks (Sukhbaatar et al., 2015), it has been applied to solve various NLP problems, such as, machine translation (Bahdanau et al., 2014), question answering (Kumar et al., 2016; Sukhbaatar et al., 2015), speech recognition (Graves et al., 2014), and many more. Hence, memory networks was chosen to model inter-speakers interaction by Hazarika et al. (2018), which produced state-of-the-art performance.

5.3 Methodology

5.3.1 Problem Definition

Given M parties, represented as $p_1, p_2, \ldots, p_M$, in a conversation, the goal is to assign appropriate emotion labels (*happy*, *sad*, *neutral*, *angry*, *excited*, and *frustrated*) to each of the containing N utterances $u_1, u_2, \ldots, u_N$. The speaker of utterance u_t is $p_{s(u_t)}$, where s represents the mapping between an utterance and corresponding party index. Further, each $u_t \in \mathbb{R}^{D_m}$ is represented by a feature vector, extracted as discussed in Section 3.3.1.

5.3.2 Our Model

We make the following three assumptions as to the factors that influence an utterance in a conversation:

1. the speaker,
2. the context defined by the previous utterances,
3. the emotion within the previous utterances.

The presented model DialogueRNN,[1] depicted in Fig. 5.2a, incorporates these factors as follows — we model each party with a *party state* that evolves along the conversation when the corresponding party speaks. This traces the emotion dynamics within the conversation, that eventually influences individual utterances. Further, we construct a *global*

[1]Implementation available at https://github.com/senticnet/conv-emotion

state per utterance, shared among the parties, as context representation which encodes information from the previous utterances and party states. This provides necessary context for informative party states. At last, target emotion representation is derived from the party state and previous speaker states as context. This is fed to classifiers for final emotion classification.

We employ three GRU cells (Chung et al., 2014) to update global, party, and emotion state, respectively. The GRU cells computes a new state, based on the previous state and new input:

$$h_t = GRU_*(h_{t-1}, x_t), \tag{5.1}$$

where t represents time-step (an utterance in our case), h_{t-1} and h_t are the previous state and new updated state, respectively and x_t is the new input. Each GRU cell has two sets of parameters — $W_{*,\{h,x\}}^{\{r,z,c\}}$ and $b_*^{\{r,z,c\}}$.

5.3.2.1 Global State (Global GRU)

Through the joint encoding of utterance and speaker state, global state captures the context of an inbound utterance, which is also speaker-specific utterance representation. Pooling over these global states aids in extracting inter-utterance and inter-speaker dependencies that helps to generate better context representation. We model this context representation using $GRU_\mathcal{G}$ cell of output size $D_\mathcal{G}$, with u_t and $q_{s(u_t),t-1}$ as inputs:

$$g_t = GRU_\mathcal{G}(g_{t-1}, (u_t \oplus q_{s(u_t),t-1})), \tag{5.2}$$

where $W_{\mathcal{G},h}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{G} \times D_\mathcal{G}}$, $W_{\mathcal{G},x}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{G} \times (D_m + D_\mathcal{P})}$, $b_\mathcal{G}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{G}}$, $q_{s(u_t),t-1} \in \mathbb{R}^{D_\mathcal{P}}$, $g_t, g_{t-1} \in \mathbb{R}^{D_\mathcal{G}}$, $D_\mathcal{G}$ is the global state vector size, $D_\mathcal{P}$ is the party state vector size, and $\oplus$ denotes concatenation.

5.3.2.2 Party State (Party GRU)

Our model traces each participating speaker with fixed sized vectors $q_1, q_2, \ldots, q_M$ along the conversation. These states hold speaker-specific information within the conversation, pertaining emotion recognition. These states are updated taking into account the current (time-step t) role of a party (either speaker or listener) and the inbound utterance u_t. At the beginning of the conversation, all of these party state vectors are set to null vectors. The primary motivation of this unit is to enforce the knowledge of the speaker of each utterance.

Speaker Update (Speaker GRU) — In general, responses are constructed based on context, defined by the previous utterances in a conversation. Thus, the context c_t for

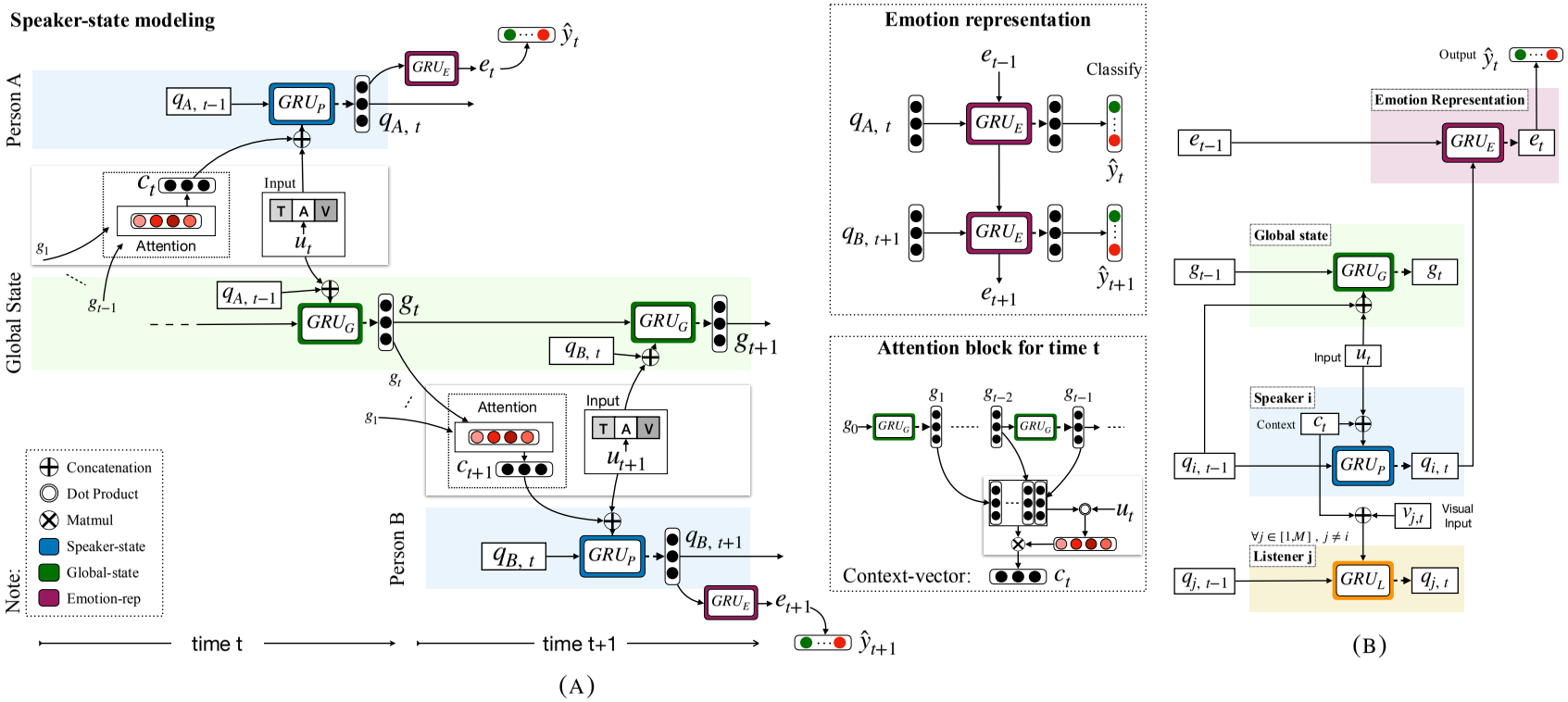


FIGURE 5.2: (a) Depiction of DialogueRNN architecture. (b) Updation of global, speaker, listener, and emotion states at t^{th} utterance in a dialogue. Person i is the speaker and persons $j \in [1, M]$ and $j \neq i$ are the listeners.

utterance u_t is constructed as follows:

$$\alpha = \text{softmax}(u_t^T W_\alpha [g_1, g_2, \ldots, g_{t-1}]), \tag{5.3}$$

$$\text{softmax}(x) = [e^{x_1}/\Sigma_i e^{x_i}, e^{x_2}/\Sigma_i e^{x_i}, \ldots], \tag{5.4}$$

$$c_t = \alpha [g_1, g_2, \ldots, g_{t-1}]^T, \tag{5.5}$$

where $g_1, g_2, \ldots, g_{t-1}$ are previous $t-1$ global states ($g_i \in \mathbb{R}^{D_\mathcal{G}}$), $W_\alpha \in \mathbb{R}^{D_m \times D_\mathcal{G}}$, $\alpha^T \in \mathbb{R}^{(t-1)}$, and $c_t \in \mathbb{R}^{D_\mathcal{G}}$. Eq. (5.3) calculates attention scores (importance measure) over the preceding global states that correspond to the preceding utterances, with respect to the inbound utterance u_t. In other words, based on emotional relevance to the inbound utterance u_t, commensurate amount of attention score is assigned to the preceding contextual utterances. At last, Eq. (5.5) calculates the context vector c_t by pooling the preceding global states with α.

In order to update the current speaker state $q_{s(u_t),t-1}$ to $q_{s(u_t),t}$, $GRU_\mathcal{P}$ of size $D_\mathcal{P}$ is employed. $GRU_\mathcal{P}$ performs this updation by taking into account the inbound utterance u_t and context c_t:

$$q_{s(u_t),t} = GRU_\mathcal{P}(q_{s(u_t),t-1}, (u_t \oplus c_t)), \tag{5.6}$$

where $W_{\mathcal{P},h}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{P} \times D_\mathcal{P}}$, $W_{\mathcal{P},x}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{P} \times (D_m + D_\mathcal{G})}$, $b_\mathcal{P}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{P}}$, and $q_{s(u_t),t}$, $q_{s(u_t),t-1} \in \mathbb{R}^{D_\mathcal{P}}$. This step infuses inbound utterance information and its context, defined as the outputs of global GRU, into to the speaker state $q_{s(u_t)}$ that aids in emotion classification downstream.

Listener Update (Listener GRU) — Here, we model the influence of inbound utterance u_t from the speaker on the listeners. Two updation schemes were experimented with:

- leave the listener states uninfluenced:

$$\forall i \neq s(u_t), q_{i,t} = q_{i,t-1}; \tag{5.7}$$

- introduce $GRU_\mathcal{L}$ that changes the listener states by taking into account the context c_t and non-verbal visual cues (facial expressions) $v_{i,t}$:

$$\forall i \neq s(u_t), q_{i,t} = GRU_\mathcal{L}(q_{i,t-1}, (v_{i,t} \oplus c_t)), \tag{5.8}$$

  where $v_{i,t} \in \mathbb{R}^{D_\mathcal{V}}$, $W_{\mathcal{L},h}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{P} \times D_\mathcal{P}}$, $W_{\mathcal{L},x}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{P} \times (D_\mathcal{V} + D_\mathcal{G})}$, and $b_\mathcal{L}^{\{r,z,c\}} \in \mathbb{R}^{D_\mathcal{P}}$. We extract the listener visual features of party i for utterance u_t using a model trained on FER2013 dataset by Arriaga et al. (2017), where $D_\mathcal{V} = 7$.

We reached the conclusion experimentally that the simpler first scheme is enough, given the result that the second scheme performs very similarly with higher number of parameters due the extra $GRU_\mathcal{L}$. We surmise that this indifference is caused by the fact that a listener is pertinent only when he/she utters. Simply put, a silent party is

non-existent in a conversation. As such, as soon as a party speaks, the corresponding party state q_i is updated using context c_t containing information from previous utterance, making explicit listener state tracking redundant. Table 5.3 verifies this observation.

5.3.2.3 Emotion Representation (Emotion GRU)

The emotion representation e_t, corresponding to utterance u_t, is generated from speaker state $q_{s(u_t),t}$ and the previous emotion representation e_{t-1}. Due to the importance of context to the emotion of the inbound utterance, relevant contextual information from the other party states $q_{s(u_{<t}),<t}$ is passed through e_{t-1} to form the emotion representation e_t. This connects party states to each other. This connection is modeled with $GRU_{\mathcal{E}}$ of size $D_{\mathcal{E}}$:

$$e_t = GRU_{\mathcal{E}}(e_{t-1}, q_{s(u_t),t}), \tag{5.9}$$

where $e_{\{t,t-1\}} \in \mathbb{R}^{D_{\mathcal{E}}}$, $W_{\mathcal{E},h}^{\{r,z,c\}} \in \mathbb{R}^{D_{\mathcal{E}} \times D_{\mathcal{E}}}$, $W_{\mathcal{E},x}^{\{r,z,c\}} \in \mathbb{R}^{D_{\mathcal{E}} \times D_{\mathcal{P}}}$, $b_{\mathcal{E}}^{\{r,z,c\}} \in \mathbb{R}^{D_{\mathcal{E}}}$, and emotion representation vector is of size $D_{\mathcal{E}}$.

It is noticeable that information on other party states are already being fed to speaker state through global state. Hence, it could be argued that previous emotional state e_{t-1} is unnecessary. However, we experimentally show in ablation study (Section 5.5.6) that without the presence of emotion GRU the performance drops. In addition, speaker and global GRUs together form the encoder network, whereas the emotion GRU plays the role of decoder. As such, inter-party connections formed in these two cases are not necessarily equivalent.

5.3.2.4 Emotion Classification

We feed to emotion representation e_t to a two-layer perceptron with the second layer with softmax activation for emotion classification of utterance u_t:

$$l_t = \text{ReLU}(W_l e_t + b_l), \tag{5.10}$$

$$\mathcal{P}_t = \text{softmax}(W_{smax} l_t + b_{smax}), \tag{5.11}$$

$$\hat{y}_t = \underset{i}{\text{argmax}}(\mathcal{P}_t[i]), \tag{5.12}$$

where $W_l \in \mathbb{R}^{D_l \times D_{\mathcal{E}}}$, $b_l \in \mathbb{R}^{D_l}$, $W_{smax} \in \mathbb{R}^{c \times D_l}$, $b_{smax} \in \mathbb{R}^c$, $\mathcal{P}_t \in \mathbb{R}^c$, and $\hat{y}_t$ is the estimated label of utterance u_t.

5.3.2.5 Training

Categorical cross-entropy with L2-regularization is employed as loss (L) for training:

$$L = -\frac{1}{\sum_{s=1}^{N} c(s)} \sum_{i=1}^{N} \sum_{j=1}^{c(i)} \log \mathcal{P}_{i,j}[y_{i,j}] + \lambda \left\|\theta\right\|_2, \tag{5.13}$$

where there are total N dialogues, dialogue i having $c(i)$ number of utterances; $\mathcal{P}_{i,j}$ and $y_{i,j}$ are probability distribution over the emotion labels and expected emotion label for utterance j of dialogue i; λ denotes the regularization weight; and θ contains the parameters:

$$\theta = \{W_\alpha, W_{\mathcal{P},\{h,x\}}^{\{r,z,c\}}, b_{\mathcal{P}}^{\{r,z,c\}}, W_{\mathcal{G},\{h,x\}}^{\{r,z,c\}}, b_{\mathcal{G}}^{\{r,z,c\}}, W_{\mathcal{E},\{h,x\}}^{\{r,z,c\}}, b_{\mathcal{E}}^{\{r,z,c\}}, W_l, b_l, W_{smax}, b_{smax}\}.$$

Adam Kingma and Ba (2015) optimizer, based on SGD, was employed to train the network. On the other hand, the hyper-parameters were optimized using grid search. Those are enlisted in Table 5.1.

Algorithm 3 DialogueRNN algorithm

procedure DIALOGUERNN(U, S) ▷ U=utterances in the conversation, S=speakers
  Initialize the participant states with null vector:
  for `i:[1,`M`]` **do**
    $q_{0,i} \leftarrow 0$
  Set the initial global and emotional state as null vector:
  $g_0 \leftarrow 0$
  $e_0 \leftarrow 0$
  Pass the dialogue through RNN:
  for `t:[1,`N`]` **do**
    $e_t, g, q_t \leftarrow$ DIALOGUECELL$(e_{t-1}, g, q_{t-1}, U_t, S_t)$
  return e

procedure DIALOGUECELL(_e, g, _q, u, s)
  Update global state:
  $g_t \leftarrow GRU_{\mathcal{G}}(g_{t-1}, u \oplus \_q_s)$
  Get context from preceding global states:
  $c \leftarrow Attention([g_1, g_2, \ldots, g_{t-1}], u)$
  Update participant states:
  for `i:[1,`M`]` **do**
    if $i = s$ **then**
      Update speaker state:
      $q_s \leftarrow GRU_{\mathcal{P}}(\_q_s, u \oplus c)$
    else
      Update listener state:
      $q_i \leftarrow \_q_i$
  Update emotion representation:
  $e \leftarrow GRU_{\mathcal{E}}(\_e, q_s)$
  return e, g, q

5.3.3 DialogueRNN Variants

The basic model is summarized in Algorithm 3. However, we experimented with the following variants of DialogueRNN (Section 5.3.2):

DialogueRNN + Listener State Update (DialogueRNN$_l$) — Both speaker and listener states are updated using two distinct GRUs with the updated speaker state $q_{s(u_t),t}$, as in Eq. (5.8).

Bidirectional DialogueRNN (BiDialogueRNN) — This is the bidirectional variant of DialogueRNN. Two distinct forward and backward DialogueRNN modules are used for forward and backward pass of the input utterance sequence, respectively. Emotion representation outputs from these two RNNs are concatenated and fed to the classifier (Section 5.3.2.4). We believe context from both past and future utterances would improve classification performance. However, this cannot be used in real-time scenarios.

DialogueRNN + attention (DialogueRNN+Att) — Attention pooling is applied over all the surrounding output emotion representations from DialogueRNN, for each utterance u_t. Attention scores are calculated by matching target emotion representation e_t with the rest of emotion representations $e_{\neq t}$ in the dialogue, as described in Eqs. (5.14) and (5.15). This filter information from both past and future utterance based on relevance.

Bidirectional DialogueRNN + Emotional attention (BiDialogueRNN+Att) — Attention pooling is applied over all the surrounding output emotion representations from BiDialogueRNN, for each utterance u_t. Attention scores are calculated by matching target emotion representation e_t with the rest of emotion representations $e_{\neq t}$ in the dialogue:

$$\beta_t = \text{softmax}(e_t^T W_\beta [e_1, e_2, \ldots, e_N]), \tag{5.14}$$

$$\tilde{e}_t = \beta_t [e_1, e_2, \ldots, e_N]^T, \tag{5.15}$$

where $e_t \in \mathbb{R}^{2D_\mathcal{E}}$, $W_\beta \in \mathbb{R}^{2D_\mathcal{E} \times 2D_\mathcal{E}}$, $\tilde{e}_t \in \mathbb{R}^{2D_\mathcal{E}}$, and $\beta_t^T \in \mathbb{R}^N$. At last, $\tilde{e}_t$ are passed to a classifier for emotion recognition, as in Section 5.3.2.4.

Hyperparameter	DialogueRNN	BiDialogueRNN	DialogueRNN+Att	BiDialogueRNN+Att
$D_\mathcal{G}$	300	150	150	150
$D_\mathcal{P}$	400	150	150	150
$D_\mathcal{E}$	400	100	100	100
D_l	200	100	100	100
lr	0.0001	0.0001	0.0001	0.0001
λ	0.00001	0.00001	0.00001	0.00001

TABLE 5.1: Hyper-parameter for DialogueRNN variants; lr = learning rate.

5.4 Experimental Setting

5.4.1 Datasets Used

IEMOCAP (Busso et al., 2008), AVEC (Schuller et al., 2012), and MELD (Poria et al., 2019b) are the three multimodal datasets used to evaluate our models. MELD is used exclusively in multimodal setting. Training and test partitions of IEMOCAP and AVEC datasets do no share speakers. Table 5.2 presents the split of samples for all the datasets.

Dataset	Partition	Utterance Count	Dialogue Count
IEMOCAP	training + validation	5810	120
	test	1623	31
AVEC	training + validation	4368	63
	test	1430	32
MELD	training	9989	1039
	validation	1109	114
	test	2610	280

TABLE 5.2: Dataset split ((train + val) / test $\approx 80\%/20\%$).

IEMOCAP (Busso et al., 2008) multimodal dataset consists of dyadic conversations among ten unique speakers. The constituent utterances of each conversation are annotated with one of the six emotion labels — *anger, happy, sad, neutral, excited,* and *frustrated*. The first 8 speakers from sessions one to four are exclusive to training set and the remaining two belong to the test set.

AVEC (Schuller et al., 2012) multimodal dataset is built using SEMAINE database (McKeown et al., 2012) as basis, that contains dyadic conversation between an artificially intelligent agent and human subjects. Each containing utterance in a dialogue comes with four real valued annotations of affective nature — valence ($[-1, 1]$), arousal ($[-1, 1]$), expectancy ($[-1, 1]$), and power ($[0, \infty)$).

MELD (Poria et al., 2019b) multimodal multi-party dataset is built by extending EmotionLines dataset (Hsu et al., 2018), that contains dialogues from the TV series *Friends*. Unlike IEMOCAP and AVEC, the dialogues in MELD contain three parties on average. MELD consists of textual, audio, and visual data from 1,400 dialogues and 13,000 utterances.

5.4.2 Baselines and State of the Art

DialogueRNN and its variants are compared against the following baseline methods:

c-LSTM (Poria et al., 2017) — Inter-utterance dependency is captured using bidirectional LSTM (Hochreiter and Schmidhuber, 1997), resulting in context-aware utterance representations. This does not consider speaker information.

c-LSTM+Att (Poria et al., 2017) — This is a variant of the c-LSTM described above. To mine better context from the neighboring utterances, attention mechanism is applied on the output of bi-LSTM, as described in Eqs. (5.14) and (5.15).

TFN (Zadeh et al., 2017) — Intra- and inter-modal interactions are modeled using vector outer product. It does not use neighboring context information. This model is specific to multimodal case.

MFN (Zadeh et al., 2018a) — Multi-view learning is employed for modality fusion with memory content. It also does not use neighboring context information. This model is specific to multimodal scenario as well.

CNN (Kim, 2014) — This CNN-based model adopts the textual feature extraction network (Section 3.3.1.1) and operates solely with target utterance. As such, lower performance is expected.

Memnet (Sukhbaatar et al., 2015) — Following the work of Hazarika et al. (2018), memory slots consist in preceding utterance representations and inbound utterance is used as query. The output of memory network is fed to softmax classifier for emotion recognition.

CMN (Hazarika et al., 2018) — CMN employs two GRUs for two speakers in a dyadic conversation. The preceding utterances of each speaker is fed to the corresponding GRU. The output vectors of these GRUs form the memory of a memory network, which accepts the target utterance representation as its query. The output of this memory network is passed to a softmax classifier for emotion classification. This is the state-of-the-art method for emotion detection in conversations.

5.4.3 Modalities

The primary focus of our model is textual modality due to its prevalence over visual and acoustic modalities. Still, to diversify its effectiveness we experimented with multimodal data as well.

5.5 Results and Discussion

Table 5.3 presents the performance (F1-score) of various baseline methods and DialogueRNN variants on IEMOCAP and AVEC dataset for textual modality. As per our

Methods	IEMOCAP													AVEC								
	Happy		Sad		Neutral		Angry		Excited		Frustrated		**Average(w)**	Valence		Arousal		Expectancy		Power	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc. F1	MAE	r	MAE	r	MAE	r	MAE	r
CNN	27.77	29.86	57.14	53.83	34.33	40.14	61.17	52.44	46.15	50.09	62.99	55.75	48.92 48.18	0.545	-0.01	0.542	0.01	0.605	-0.01	8.71	0.19
memnet	25.72	33.53	55.53	61.77	58.12	52.84	59.32	55.39	51.50	58.30	67.20	59.00	55.72 55.10	0.202	0.16	0.211	0.24	0.216	0.23	8.97	0.05
c-LSTM	29.17	34.43	57.14	60.87	54.17	51.81	57.06	56.73	51.17	57.95	67.19	58.92	55.21 54.95	0.194	0.14	0.212	0.23	0.201	0.25	8.90	-0.04
c-LSTM+Att	30.56	35.63	56.73	62.90	57.55	53.00	59.41	59.24	52.84	58.85	65.88	59.41	56.32 56.19	0.189	0.16	0.213	0.25	0.190	0.24	8.67	0.10
CMN (SOTA)	25.00	30.38	55.92	62.41	52.86	52.39	61.76	59.83	55.52	60.25	71.13	60.69	56.56 56.13	0.192	0.23	0.213	0.29	0.195	0.26	8.74	-0.02
DialogueRNN	31.25	33.83	66.12	69.83	63.02	57.76	61.76	62.50	61.54	64.45	59.58	59.46	59.33 59.89	0.188	0.28	0.201	0.36	0.188	0.32	8.19	0.31
DialogueRNN$_l$	35.42	35.54	65.71	69.85	55.73	55.30	62.94	61.85	59.20	62.21	63.52	59.38	58.66 58.76	0.189	0.27	0.203	0.33	0.188	0.30	8.21	0.30
BiDialogueRNN	32.64	36.15	71.02	74.04	60.47	56.16	62.94	63.88	56.52	62.02	65.62	**61.73**	60.32 60.28	0.181	0.30	0.198	0.34	0.187	0.34	8.14	0.32
DialogueRNN+Att	28.47	**36.61**	65.31	72.40	62.50	57.21	67.65	**65.71**	70.90	68.61	61.68	60.80	61.80 61.51	0.173	0.35	0.168	0.55	0.177	0.37	7.91	0.35
BiDialogueRNN+Att	25.69	33.18	75.10	**78.80**	58.59	**59.21**	64.71	65.28	80.27	**71.86**	61.15	58.91	63.40 **62.75**	**0.168**	**0.35**	**0.165**	**0.59**	**0.175**	**0.37**	**7.90**	**0.37**

TABLE 5.3: Comparison against the baseline methods for textual modality; Acc. stands for Accuracy, MAE stands for Mean Absolute Error, r stands for Pearson correlation coefficient; bold font signifies the best performances. Average(w) stands for Weighted average.

expectation, all the baseline methods and the state-of-the-art CMN are surpassed by all the DialogueRNN variants.

5.5.1 Comparison with the State of the Art

The performance of DialogueRNN and the state-of-the-art CMN is compared for text modality.

5.5.1.1 IEMOCAP

Following Table 5.3, DialogueRNN outperforms the SOTA by 2.77% accuracy and 3.76% f1-score on average, on IEMOCAP dataset. We surmise that the improvement in performance is achieved by the following elementary differences between our model and the SOTA:

1. modeling of party state using $GRU_{\mathcal{P}}$ (Eq. (5.6));
2. modeling of utterances with respect to the corresponding speaker (Eqs. (5.2) and (5.6));
3. and global context modeling using $GRU_{\mathcal{G}}$ in (Eq. (5.2)).

Due to IEMOCAP containing six emotion labels, which are unbalanced, it is necessary to explore model performance per emotion label. The SOTA CMN is significantly outperformed by DialogueRNN on five among the six labels. The exception is *frustrated* class, where CMN surpasses DialogueRNN by 1.23% f1-score. It might be possible that DialogueRNN would beat CMN using a *frustrated* class-specific classifier. On the other hand, the DialogueRNN variants already outperform CMN on *frustrated* class (Table 5.3).

5.5.1.2 AVEC

CMN is beaten by our model DialogueRNN on all of the four attributes, namely *valence*, *arousal*, *expectancy*, and *power*. This is observable in Table 5.3 where DialogueRNN produces significantly lower MAE and higher pearson correlation coefficient (r) across all four attributes. We posit this is achieved by the modeling of party state and emotion GRU in DialogueRNN, which CMN lacks.

5.5.2 DialogueRNN vs. DialogueRNN Variants

We compare DialogueRNN against its variants on textual modality in the context of IEMOCAP and AVEC dataset.

DialogueRNN$_l$ — The explicit updation of listener state does not result improved performance, as we originally expected. Further, it leads to slightly inferior results, as can be seen in Table 5.3 (DialogueRNN$_l$) for both IEMOCAP and AVEC. Strangely, DialogueRNN$_l$ performs better than DialogueRNN on *happy* emotion label by 1.71% f1-score. This overall performance drop is most likely caused by the training of one extra GRU cell to accommodate listener update. Since, a party becomes relevant only when he/she verbally participates in the conversation, the listener GRU does not bring anything the model but noise. We surmise this noise leads to performance drop.

BiDialogueRNN — Owing to the bidirectional nature of BiDialogueRNN, it captures contextual information from both past and future utterances. As such, evidenced by Table 5.3 BiDialogueRNN performs better than DialogueRNN overall on both datasets.

DialogueRNN+Attn — In contrast to BiDialogueRNN, DialogueRNN+Attn filters information from both past and future utterances using attention mechanism. This enriches emotional context as compared to BiDialogueRNN, resulting in improvement of 1.23% f1-score on IEMOCAP and lower MAE and higher r for AVEC.

BiDialogueRNN+Attn — This setup combines the improvements of both of the previous two variants. This results in even better context representation. Following Table 5.3, this has led to 6.62% and 2.86% improvement in f1-score over the state-of-the-art CMN and simple DialogueRNN on IEMOCAP. This setting also yields the best performance on all the attributes in AVEC.

5.5.3 Multimodal and Multi-Party Setting

Methods	IEMOCAP	AVEC				MELD (Multi Party)
	F1	Valence (r)	Arousal (r)	Expectancy (r)	Power (r)	F1
CNN	–	–	–	–	–	55.02
TFN	56.8	0.01	0.10	0.12	0.12	–
MFN	53.5	0.14	25	0.26	0.15	–
c-LSTM	58.3	0.14	0.23	0.25	-0.04	56.70
CMN	58.5	0.23	0.30	0.26	-0.02	–
DialogueRNN$_{text}$	59.9	0.28	0.36	0.32	0.31	–
BiDialogueRNN$_{text}$	60.3	0.30	0.34	0.34	0.32	–
BiDialogueRNN+att$_{text}$	62.7	0.35	0.59	0.37	0.37	–
BiDialogueRNN+att$_{MM}$	**62.9**	**0.37**	**0.60**	**0.37**	**0.41**	**57.03**

TABLE 5.4: Comparison against the baselines for trimodal (T+V+A) and multi-party setting. BiDialogueRNN+att$_{MM}$ stands for BiDialogueRNN+att in multimodal setting.

Following Table 5.4, we obtain 4.4% f1-score improvement over the state-of-the-art CMN (Hazarika et al., 2018), on IEMOCAP for multimodal scenario. We observe

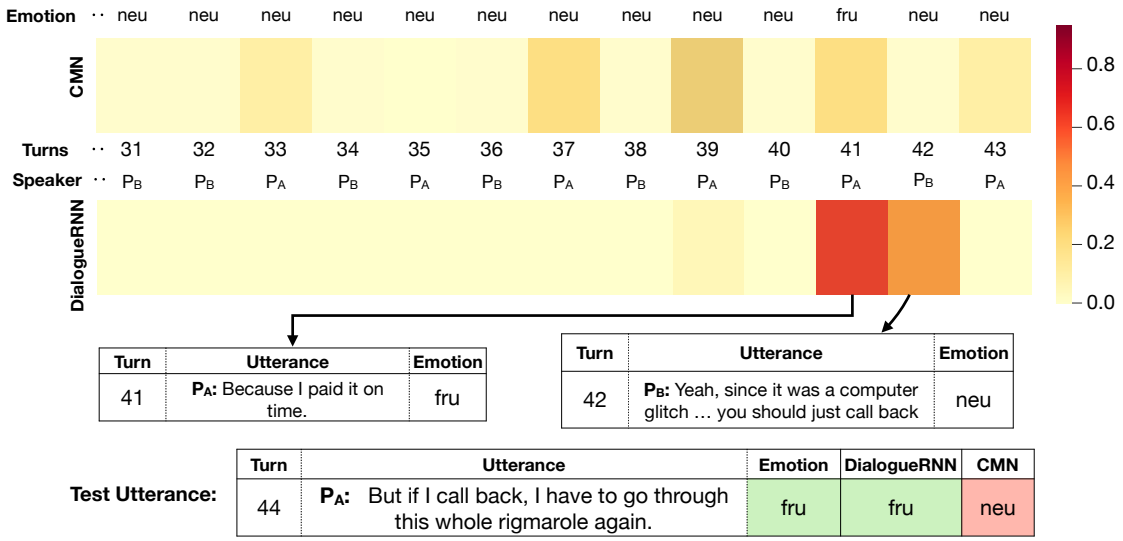


FIGURE 5.3: Comparison of attention scores over utterance history of CMN and DialogueRNN.

consistent improvement on all four attributes of AVEC dataset for multimodal input as well. However, the improvement over textual modality for the same DialogueRNN model is negligible 0.2%. We surmise that this is caused by the simple fusion mechanism (just concatenation), since multimodal fusion is not the focus of this chapter.

For multi-party multi-modal scenario, the improvement over the baseline c-LSTM (Poria et al., 2017) is minimal 0.33% f1-score. However, we must take into account the variation of party counts between dialogues and total of ten parties.

5.5.4 Case Studies

Dependency on preceding utterances (DialogueRNN) — The attention layer over the global states (output of GRU_G) is one of the key parts of our model. In Fig. 5.3, we compare the α attention scores (Eq. (5.3)) on the preceding contextual utterances against the attention scores of the same set of utterances of CMN. It is visible that the attention of DialogueRNN is more focused compared to CMN, where the attention scores are diluted over the utterances. We suppose this unfocused scores has led to misclassification in case of CMN. Since, we observe similar patterns across various samples, we surmise this could be indicative of the confidence of the model about its decision. We also observe a shift in emotion, from *neutral* to *frustrated*, in the inbound utterance by P_A (turn 44) over the previous utterance. CMN fails to predict this change for we suppose due to week dependency (α attention) with the context utterances, leading to misclassifying the inbound utterance as *neutral*. In contrast, our model detects this emotion shift by strongly attending to turn 41 and 42 by P_A and P_B, respectively, leading to correct prediction. This illustrates the ability of DialogueRNN to capture the self and inter-party dependencies that lead to emotion shift.

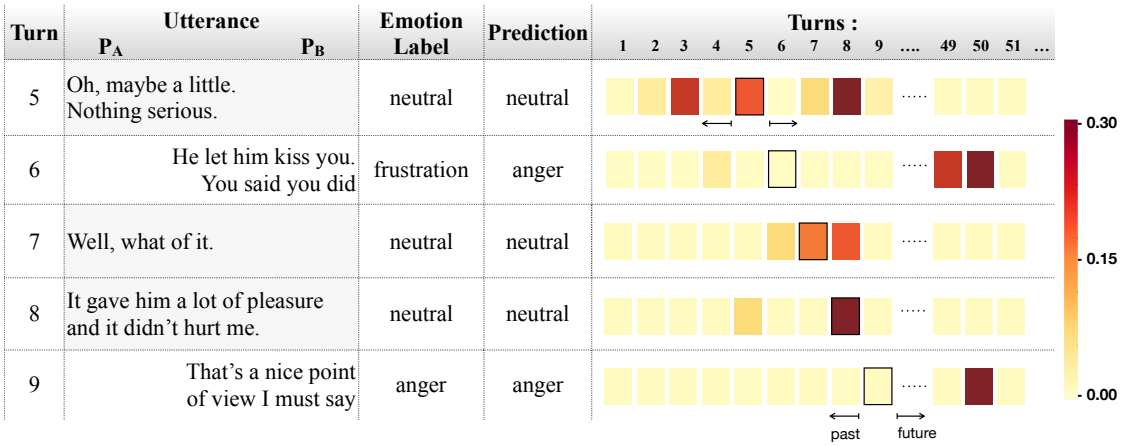


FIGURE 5.4: Illustration of the β attention weights over emotion representations e_t for a segment of conversation between a couple; P_A is the woman, P_B is the man.

Dependency on future utterances (BiDialogueRNN+Att) — Fig. 5.4 depicts the evolution of β attention for inbound utterances (Eq. (5.14)) over an excerpt of five turns from a dyadic conversation. It is noticeable that party P_A is *neutral* and in contrast P_B is angry. We also observe the expected pattern that P_A's utterances (turn 5, 7, and 8) strongly attend to the *neutral* turn 8, which is located in the future with respect to turn 5 and 7. Turn 5 attends to both past and future (turn 5 and 8, respectively) utterances, respectively. We observed similar inter-utterance dependencies between inbound and past/future utterances.

Another instance of fruitful focus on future utterance would be the turns of P_B — turn 6 and 9. These two utterances attend to distant contextual utterances (turn 49 and 50) which are representative of irate state of P_B. Even though, DialogueRNN misclassifies target turn 6, our model is able to match a similar emotional state (*anger*) with the real state (*frustrated*). Such errors by our model are explored in Section 5.5.5.

Dependency on distant context — Fig. 5.5 shows the histogram of correctly predicted IEMOCAP test utterances over the relative distance between the inbound utterance and its corresponding most (left plot) and second-most (right plot) important context utterance. We observe the expected trend of decreasing number of context sentence with increasing relative distance. Nonetheless, a significant proportion (around 18%) of these correctly predicted utterances focus on far away utterances that are $20--40$ turns away. This is indicative of the importance of long-lasting and distant emotional relationships among utterances. These cases are mostly prevalent in the conversations where participants stay in specific affective mood and do not trigger emotion shift. Fig. 5.6 illustrates such a scenario where long-term dependency is relevant. The participants in this conversation mostly retain *happy* mood. However, the 34^{th} utterance, "*Horrible thing. I hated it.*", appears to express negative emotion. But, upon consideration of its context, it is evident that it expresses *excited* emotion actually. Our model focuses back on 11^{th} and 14^{th} utterance to decipher the intended emotion of the target utterance.

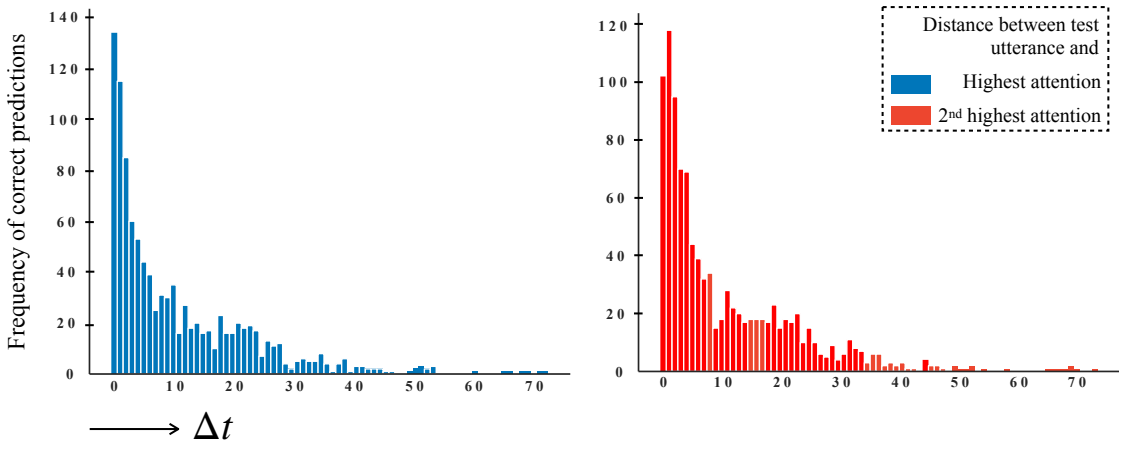


FIGURE 5.5: Histogram of Δt = distance between the target utterance and its context utterance based on β attention scores.

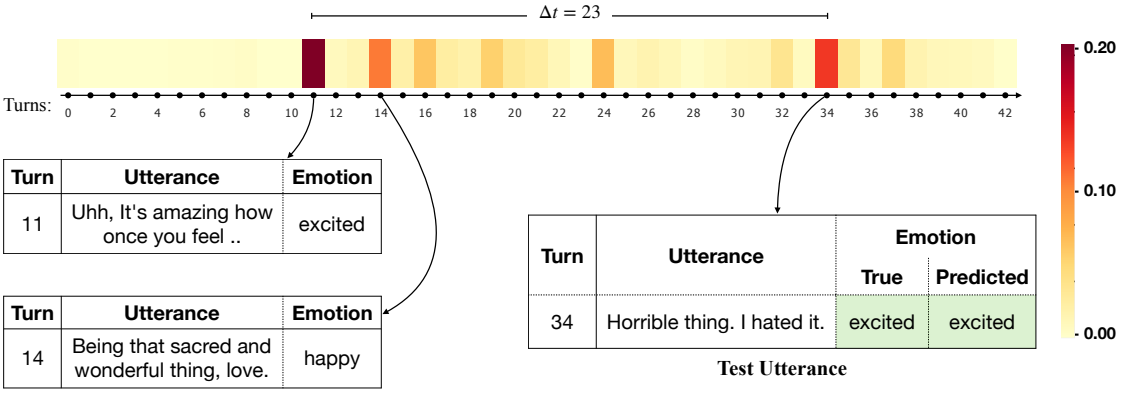


FIGURE 5.6: An example of long-term dependency among utterances.

5.5.5 Error Analysis

A very prevalent tendency among the predictions is cross-prediction among related emotions. The most common of such misclassifications happens for *happy* utterances that are mischaracterized as *excited* and vice versa. Similarly, many *anger* and *happy* utterances are misclassified as the other. We surmise this is caused by the subtle differences between these pair of emotions that are hard to catch. Also, the *neutral* class has high false-positives, which could be due to its majority share among the utterances.

We noticed a significant number of misclassifications at the utterances that express different emotion than its previous utterance from the same party. Our model correctly predicts only 47.5% of the utterances where this *emotion shift* occurs, as compared to 69.2% of the utterances where *emotion shift* is absent. Shift of emotions is caused by many complex latent factors that often originate from the other parties. Investigating and pointing out these factors to predict *emotion shift* stays an open research area.

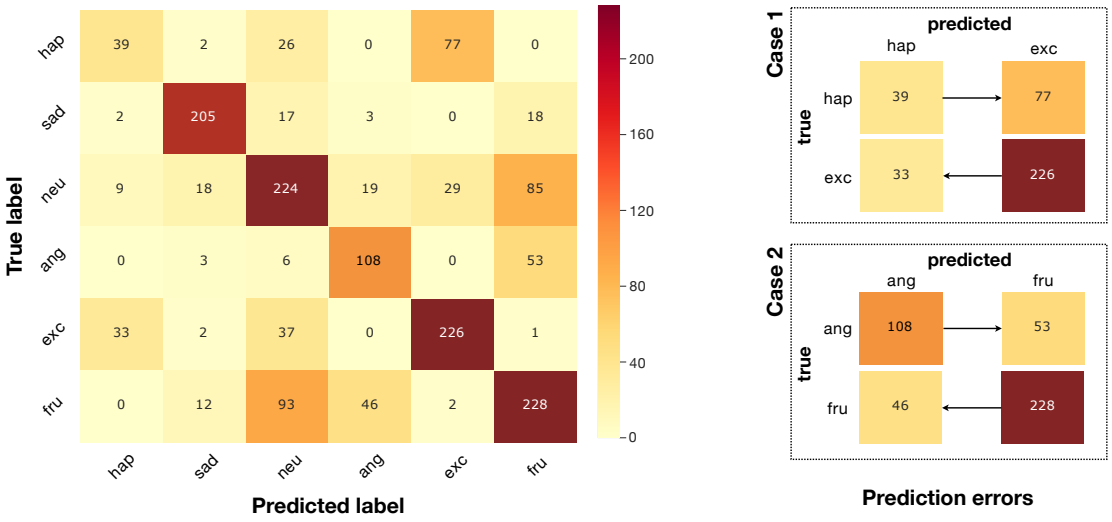


FIGURE 5.7: Confusion matrix of the predictions on IEMOCAP; the image to the right shows the two most common cases of cross-prediction.

5.5.6 Ablation Study

The primary contribution of our method is the incorporation of party state and emotion GRU ($GRU_{\mathcal{E}}$). In order to assess the contribution of each of these two components, we take apart these two off the model one at a time and monitor its performance on IEMOCAP dataset.

Following Table 5.5, the performance of our model drops by a massive 4.33% f1-score by the exclusion of party state. This solidifies the importance of party state. We suppose that the party state aids in mining party-specific emotional context.

Party State	Emotion GRU	F1
-	+	55.56
+	-	57.38
+	+	59.89

TABLE 5.5: Performance of ablated DialogueRNN models on IEMOCAP dataset.

The exclusion of emotion GRU leads to performance fall of 2.51%, which is less as compared to party state. Nonetheless, emotion GRU has pivotal importance as it likely improves context flow from other parties through the previous emotion representation.

5.6 Conclusion

In this chapter, we showcased an RNN-based neural network, namely DialogueRNN, for conversational emotion recognition. As compared to the state-of-the-art method CMN, our model considers the speaker characteristics to generate enhanced context to the inbound utterance. DialogueRNN outperforms the state of the art on both textual and

multimodal setting on multiple datasets. It also outperforms c-LSTM on multi-party setting. Further, unlike the state of the art, our model is scalable to arbitrary number of participants in conversation. Solving the issue of emotion shift would lead to improved conversational emotion detection. As such, we leave this to our future work.

Chapter 6

Conclusion

This chapter concludes this thesis by showcasing the contributions, listing its derived publications, and laying out possible future directions of research.

6.1 Contributions

This thesis presents the novel methods that we developed during the course of my PhD. We started with multimodal sentiment analysis, followed by aspect-based sentiment analysis, and eventually worked our way to opinion mining in multimodal multi-party conversations. We designed and trained neural network-based architectures to solve these tasks. Further, the implementation of these architectures are made available to public to aid future research.

We believe our work has strong potential to make significant contribution to the already active research area of opinion mining and sentiment analysis, owing to the recently found interest from industry and government alike. This interest has come from myriad of potential applications in healthcare, economics, security, management to name a few.

Theoretical Contributions

Our primary theoretical contributions consist in the following aspects we incorporated into our methods:

- **Improved Multimodal Feature Fusion —** Feature fusion is a very important step any multimodal task. Most fusion strategies in the literature only encodes the unimodal features into unified multimodal space. However, we improve upon this strategy by employing an auto-encoder setup to reconstruct the original unimodal representations. This resulted in improved multimodal sentiment and emotion recognition performance.

- **Improved Unsupervised Multimodal Feature Fusion —** Previously, simple concatenation fusion was the only reasonable unsupervised feature fusion method. However, the resulting vector often contains a lot of redundant information across the modalities. Our auto-encoder-based fusion strategy solves this redundancy issue.

- **Inter-Aspect Dependency Modeling —** Accurate aspect-aware sentence representation is key to effective aspect-based sentiment analysis. Often, aspects within the same sentence obfuscates the context of each other as they can be intertwined. Recent works on aspect-based sentiment analysis do not consider this inter-aspect dependency. We modeled this dependency in our model that resulted in improved performance, not only on aspects having neighboring aspects, but also single aspect case.

- **Improved Incorporation of Speaker Information in ERC —** Effective incorporation of speaker knowledge is important to give utterances proper context in conversations. Existing works on ERC do not consider speaker information for classification. We bridge this gap by profiling each participant on the fly and use that profiling during classification.

- **Scalable Multi-Party ERC —** Existing ERC algorithms have dedicated training parameters for each participant. As such, those cannot work with more participants than predefined by the model architecture. However, we dynamically profile each speaker on the fly with their spoken utterances. This resulted in much improved classification performance.

- **Improved Real-time Multi-Party ERC —** Our ERC model, DialogueRNN can be adopted for real-time scenario by setting the context window for global GRU to some fixed size. It would suffer some performance drop though, due to the loss of long-term dependencies when necessary.

Technical Contributions

We used PyTorch (Paszke et al., 2017) to implement and optimize all the presented neural networks. We have also made the implementations available:

- **IARM (Chapter 4) —** `http://github.com/senticnet/IARM`

- **DialogueRNN (Chapter 5) —** `https://github.com/senticnet/conv-emotion`

- **Variational Fusion (Chapter 3) —** will be made available upon acceptance

6.2 Publications

The following publications stemmed from the work done in this thesis:

Journal

1. N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, and S. Poria. "Multimodal sentiment analysis using hierarchical fusion with context modeling". In: *Knowledge-Based Systems* 161 (2018), pp. 124–133. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2018.07.041. URL: http://www.sciencedirect.com/science/article/pii/S0950705118303897.

   Impact Factor: **5.101 (Q1)**

2. S. Poria, N. Majumder, D. Hazarika, E. Cambria, A. Gelbukh, and A. Hussain. "Multimodal Sentiment Analysis: Addressing Key Issues and Setting Up the Baselines". In: *IEEE Intelligent Systems* 33.6 (Nov. 2018), pp. 17–25. ISSN: 1541-1672. DOI: 10.1109/MIS.2018.2882362.

   Impact Factor: **4.464 (Q1)**

3. N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. Gelbukh. "Sentiment and Sarcasm Classification With Multitask Learning". In: *IEEE Intelligent Systems* 34.3 (May 2019), pp. 38–43. ISSN: 1541-1672. DOI: 10.1109/MIS.2019.2904691.

   Impact Factor: **4.464 (Q1)**

4. Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria. "Recent trends in deep learning based personality detection". In: *Artificial Intelligence Review* (Oct. 2019). ISSN: 1573-7462. DOI: 10.1007/s10462-019-09770-z. URL: https://doi.org/10.1007/s10462-019-09770-z.

   Impact Factor: **5.095 (Q1)**

5. N. Majumder, S. Poria, A. Gelbukh, and E. Cambria. "Deep learning based document modeling for personality detection from text". In: *IEEE Intelligent Systems* 32.2 (2017), pp. 74–79.

   Impact Factor: **4.464 (Q1)**

6. S. Poria, N. Majumder, R. Mihalcea, and E. Hovy. "Emotion Recognition in Conversation: Research Challenges, Datasets, and Recent Advances". In: *IEEE Access* 7 (2019), pp. 100943–100953. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2929050.

   Impact Factor: **4.098 (Q1)**

Conference

7. N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria. "DialogueRNN: An Attentive RNN for Emotion Detection in Conversations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. July 2019, pp. 6818–6825. DOI: 10.1609/aaai.v33i01.33016818. URL: https://aaai.org/ojs/index.php/AAAI/article/view/4657.

8. N. Majumder, S. Poria, A. Gelbukh, M. S. Akhtar, E. Cambria, and A. Ekbal. "IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3402–3411. URL: http://www.aclweb.org/anthology/D18-1377.

9. D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh. "DialogueGCN: A Graph Convolutional Neural Network for Emotion Recognition in Conversation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 154–164. DOI: 10.18653/v1/D19-1015. URL: https://www.aclweb.org/anthology/D19-1015.

10. S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea. "MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 527–536. DOI: 10.18653/v1/P19-1050. URL: https://www.aclweb.org/anthology/P19-1050.

6.3 Future Work

In the future, we plan to develop sophisticated encoders for multimodal fusion. We would also like to check if their performance can be boosted with variational fusion. For aspect-based sentiment analysis, there is still a lot of room of improvement in aspect-aware sentence representation that is key to accurate sentiment classification. We intend to employ transformer networks to improve this representation. On emotion recognition in conversation front, we will focus on detecting *emotion shift*, as we believe it will lead to significant performance improvement. We would also like to construct or extend existing ERC dataset that will contain other relevant annotations, representing controlling variables, suggested by Poria et al. (2019a) — intent, topic, speaker personality, etc.

Bibliography

[1] C. O. Alm, D. Roth, and R. Sproat. “Emotions from text: machine learning for text-based emotion prediction”. In: *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics. 2005, pp. 579–586.

[2] O. Arriaga, M. Valdenegro-Toro, and P. Plöger. “Real-time Convolutional Neural Networks for Emotion and Gender Classification”. In: *CoRR* abs/1710.07557 (2017). arXiv: 1710.07557. URL: http://arxiv.org/abs/1710.07557.

[3] D. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[4] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan. “IEMOCAP: Interactive emotional dyadic motion capture database”. In: *Language resources and evaluation* 42.4 (2008), pp. 335–359.

[5] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall. “Sentiment Analysis is a Big Suitcase”. In: *IEEE Intelligent Systems* 32.6 (2017).

[6] 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.

[7] M. Chen, S. Wang, P. P. Liang, T. Baltrušaitis, A. Zadeh, and L.-P. Morency. “Multimodal sentiment analysis with word-level fusion and reinforcement learning”. In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ACM. 2017, pp. 163–171.

[8] W. Chen, D. Grangier, and M. Auli. “Strategies for Training Large Vocabulary Neural Language Models”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1975–1985. DOI: 10.18653/v1/P16-1186. URL: https://www.aclweb.org/anthology/P16-1186.

[9] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: *CoRR* abs/1412.3555 (2014). arXiv: 1412.3555. URL: http://arxiv.org/abs/1412.3555.

[10] D. Datcu and L. Rothkrantz. “Semantic audio-visual data fusion for automatic emotion recognition”. In: *Euromedia’2008* (2008).

[11] L. C. De Silva, T. Miyasato, and R. Nakatsu. “Facial emotion recognition using multi-modal information”. In: *Proceedings of ICICS*. Vol. 1. IEEE. 1997, pp. 397–401.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

[13] M. Ebrahimi, A. Hossein, and A. Sheth. “Challenges of sentiment analysis for dynamic events”. In: *IEEE Intelligent Systems* 32.5 (2017).

[14] P. Ekman. “Facial expression and emotion.” In: *American psychologist* 48.4 (1993), p. 384.

[15] F. Eyben and B. Schuller. “openSMILE:): The Munich open-source large-scale multimedia feature extractor”. In: *ACM SIGMultimedia Records* 6.4 (2015), pp. 4–13.

[16] 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.

[17] A. Graves, G. Wayne, and I. Danihelka. “Neural turing machines”. In: *arXiv preprint arXiv:1410.5401* (2014).

[18] D. Hazarika, S. Poria, A. Zadeh, E. Cambria, L.-P. Morency, and R. Zimmermann. “Conversational Memory Network for Emotion Recognition in Dyadic Dialogue Videos”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2122–2132. URL: http://www.aclweb.org/anthology/N18-1193.

[19] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[20] C.-C. Hsu, S.-Y. Chen, C.-C. Kuo, T.-H. Huang, and L.-W. Ku. “EmotionLines: An Emotion Corpus of Multi-Party Conversations”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by N. C. (chair) et al. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. ISBN: 979-10-95546-00-9.

[21] 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.

[22] L. Kessous, G. Castellano, and G. Caridakis. "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis". In: *Journal on Multimodal User Interfaces* 3.1-2 (2010), pp. 33–48.

[23] Y. Kim. "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882* (2014).

[24] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *Proceedings of ICLR 2015*. 2015.

[25] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *Proceedings of ICLR 2014*. 2014.

[26] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. "Ask me anything: Dynamic memory networks for natural language processing". In: *International Conference on Machine Learning*. 2016, pp. 1378–1387.

[27] C. Li, X. Guo, and Q. Mei. "Deep Memory Networks for Attitude Identification". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM. 2017, pp. 671–680.

[28] X. Li, L. Bing, W. Lam, and B. Shi. "Transformation Networks for Target-Oriented Sentiment Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 946–956. URL: http://www.aclweb.org/anthology/P18-1087.

[29] T. Luong, H. Pham, and C. D. Manning. "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1412–1421. URL: http://aclweb.org/anthology/D15-1166.

[30] D. Ma, S. Li, X. Zhang, and H. Wang. "Interactive Attention Networks for Aspect-Level Sentiment Classification". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, pp. 4068–4074. DOI: 10.24963/ijcai.2017/568. URL: https://doi.org/10.24963/ijcai.2017/568.

[31] N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, and S. Poria. "Multimodal sentiment analysis using hierarchical fusion with context modeling". In: *Knowledge-Based Systems* 161 (2018), pp. 124–133. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2018.07.041. URL: http://www.sciencedirect.com/science/article/pii/S0950705118303897.

[32] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder. "The SEMAINE Database: Annotated Multimodal Records of Emotionally Colored Conversations between a Person and a Limited Agent". In: *IEEE Transactions on Affective Computing* 3.1 (Jan. 2012), pp. 5–17. ISSN: 1949-3045. DOI: 10.1109/T-AFFC.2011.20.

[33] A. Mehrabian. "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament". In: *Current Psychology* 14.4 (Dec. 1996), pp. 261–292. ISSN: 1936-4733. DOI: 10.1007/BF02686918. URL: https://doi.org/10.1007/BF02686918.

[34] R. Mihalcea and A. Garimella. "What men say, what women hear: Finding gender-specific meaning shades". In: *IEEE Intelligent Systems 31(4), pp. 62-67 (2016)* 31.4 (2016), pp. 62–67.

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

[36] F. Morin and Y. Bengio. "Hierarchical probabilistic neural network language model". In: *AISTATS'05*. 2005, pp. 246–252.

[37] A. Paszke et al. "Automatic Differentiation in PyTorch". In: *NeurIPS Autodiff Workshop*. 2017.

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

[39] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://www.aclweb.org/anthology/N18-1202.

[40] R. W. Picard. "Affective Computing: From Laughter to IEEE". In: *IEEE Transactions on Affective Computing* 1.1 (Jan. 2010), pp. 11–17. ISSN: 1949-3045. DOI: 10.1109/T-AFFC.2010.10.

[41] R. Plutchik. "A psychoevolutionary theory of emotions". In: *Social Science Information* 21.4-5 (1982), pp. 529–553. DOI: 10.1177/053901882021004003.

[42] S. Poria, N. Majumder, R. Mihalcea, and E. Hovy. "Emotion Recognition in Conversation: Research Challenges, Datasets, and Recent Advances". In: *IEEE Access* 7 (2019), pp. 100943–100953. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2929050.

[43] S. Poria, E. Cambria, and A. Gelbukh. "Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network". In: *Knowledge-Based Systems* 108 (2016), pp. 42–49.

[44] 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.

[45] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, and L.-P. Morency. "Context-Dependent Sentiment Analysis in User-Generated Videos". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 873–883. URL: http://aclweb.org/anthology/P17-1081.

[46] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea. "MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 527–536. DOI: 10.18653/v1/P19-1050. URL: https://www.aclweb.org/anthology/P19-1050.

[47] J. E. Ramos. "Using TF-IDF to Determine Word Relevance in Document Queries". In: 2003.

[48] J. M. Richards, E. A. Butler, and J. J. Gross. "Emotion regulation in romantic relationships: The cognitive consequences of concealing feelings". In: *Journal of Social and Personal Relationships* 20.5 (2003), pp. 599–620.

[49] S. E. Robertson. "Understanding inverse document frequency: on theoretical arguments for IDF". In: *Journal of Documentation* 60 (2004), pp. 503–520.

[50] 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.

[51] J. Russell. "A Circumplex Model of Affect". In: *Journal of Personality and Social Psychology* 39 (Dec. 1980), pp. 1161–1178. DOI: 10.1037/h0077714.

[52] J. Ruusuvuori. "Emotion, affect and conversation". In: *The handbook of conversation analysis* (2013), pp. 330–349.

[53] B. Schuller. "Recognizing affect from linguistic information in 3D continuous space". In: *IEEE Transactions on Affective Computing* 2.4 (2011), pp. 192–205.

[54] B. Schuller, M. Valster, F. Eyben, R. Cowie, and M. Pantic. "AVEC 2012: The Continuous Audio/Visual Emotion Challenge". In: *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. ICMI '12. Santa Monica, California, USA: ACM, 2012, pp. 449–456. ISBN: 978-1-4503-1467-1. DOI: 10.1145/2388676.2388776. URL: http://doi.acm.org/10.1145/2388676.2388776.

[55] L. Shu, H. Xu, and B. Liu. "Lifelong learning crf for supervised aspect extraction". In: *arXiv preprint arXiv:1705.00251* (2017).

[56] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. M. A. Patwary, P. Prabhat, and R. P. Adams. "Scalable Bayesian Optimization Using Deep Neural Networks". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 2171–2180. URL: http://dl.acm.org/citation.cfm?id=3045118.3045349.

[57] C. Strapparava and R. Mihalcea. "Annotating and identifying emotions in text". In: *Intelligent Information Access*. Springer, 2010, pp. 21–38.

[58] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. "End-to-end Memory Networks". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 2440–2448. URL: http://dl.acm.org/citation.cfm?id=2969442.2969512.

[59] D. Tang, B. Qin, X. Feng, and T. Liu. "Effective LSTMs for Target-Dependent Sentiment Classification". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 3298–3307. URL: http://aclweb.org/anthology/C16-1311.

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

[61] Y. Tay, A. T. Luu, and S. C. Hui. "Learning to Attend via Word-Aspect Associative Fusion for Aspect-based Sentiment Analysis". In: *arXiv preprint arXiv:1712.05403* (2017).

[62] V. Teh and G. E. Hinton. "Rate-coded restricted Boltzmann machines for face recognition". In: *Advances in neural information processing system*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. 2001, pp. 908–914.

[63] 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.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[65] Y. Wang, M. Huang, x. zhu xiaoyan, and L. Zhao. "Attention-based LSTM for Aspect-level Sentiment Classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 606–615. URL: https://aclweb.org/anthology/D16-1058.

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

[67] M. Wöllmer, A. Metallinou, F. Eyben, B. Schuller, and S. S. Narayanan. "Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling". In: *INTERSPEECH 2010*. 2010.

[68] 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". In: *IEEE Intelligent Systems* 28.3 (2013), pp. 46–53.

[69] Y. Wu et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *CoRR* abs/1609.08144 (2016). arXiv: 1609.08144. URL: http://arxiv.org/abs/1609.08144.

[70] F. Xing, E. Cambria, and R. Welsch. "Natural Language Based Financial Forecasting: A Survey". In: *Artificial Intelligence Review* (2017).

[71] T. Young, D. Hazarika, S. Poria, and E. Cambria. "Recent trends in deep learning based natural language processing". In: *IEEE Computational Intelligence Magazine* 13.3 (2018), pp. 55–75.

[72] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. "Tensor Fusion Network for Multimodal Sentiment Analysis". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1103–1114. URL: https://www.aclweb.org/anthology/D17-1115.

[73] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency. "Memory Fusion Network for Multi-view Sequential Learning". In: *AAAI Conference on Artificial Intelligence*. 2018, pp. 5634–5641. URL: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17341/16122.

[74] A. Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency. "Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2236–2246. URL: http://www.aclweb.org/anthology/P18-1208.

[75] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L.-P. Morency. "Multi-attention Recurrent Network for Human Communication Comprehension". In: *AAAI Conference on Artificial Intelligence*. 2018. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17390.

[76] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L.-P. Morency. "Multi-attention recurrent network for human communication comprehension". In: *AAAI Conference on Artificial Intelligence*. 2018, pp. 5642–5649.

[77] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency. "Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages". In: *IEEE Intelligent Systems* 31.6 (2016), pp. 82–88.