Resource Creation and its Evaluation for Aspect Based Sentiment Analysis in Telugu

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics by Research

by

REGATTE YASHWANTH REDDY
201525036
yashwanthreddy.r@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
APRIL 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **"Resource Creation and its Evaluation for Aspect Based Sentiment Analysis in Telugu"** by **Yashwanth Reddy Regatte**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Radhika Mamidi

This thesis is dedicated to my grandparents.

Acknowledgments

I would like to extend my gratitude to my advisor Radhika mam, for always being supportive and guiding me in the right direction. I would also like to thank all the professors for helping me gain knowledge.

I would like to thank my friends Rama Rohit, Krishna Chaitanya, Harsha, Jyotish, Siddhartha, Shanmukh, Akhil, and Sudheer for always helping me and motivating me.

Finally, Thanks to my parents, family and friends for always supporting me.

Abstract

In recent years, sentiment analysis has gained popularity as it is essential to moderate and analyse the information across the Internet. It has various applications like opinion mining, social media monitoring, and market research. Aspect Based Sentiment Analysis (ABSA) is an area of sentiment analysis which deals with sentiment at a finer level. ABSA classifies sentiment with respect to each aspect to gain greater insights into the sentiment expressed.

Significant contributions have been made in ABSA, but this progress is limited only to a few languages with adequate resources. Telugu lags behind in this area of research despite being one of the most spoken languages in India and an enormous amount of data being created each day. In this thesis, we create a reliable resource for aspect based sentiment analysis in Telugu. The data is annotated for three tasks namely Aspect Term Extraction, Aspect Polarity Classification and Aspect Categorisation. Further, we develop baselines for the tasks using deep learning methods demonstrating the reliability and usefulness of the resource.

In addition to this, experimentation has been done with transformer models, as they have led to pivotal changes in the field of NLP. Specifically, the setting of monolingual and multilingual models has been chosen to compare the effective contribution of the dataset created for ABSA in Telugu.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The last couple of decades has seen a dramatic rise in the usage of social media around the world. Social media has become the major medium for information consumption. While this has revolutionized communication, it is not without several pitfalls. There have been quite a few instances, where social media has been used for spreading inflammatory agendas. For instance, negligence of Facebook in prompt flagging of hate-speech during the Rohingya crisis in Myanmar has led to a mass genocide [1]. This is a direct consequence of inability of being able to analyse content in a low resource language for hate speech detection.

This calls for an imminent need for reliable tools to analyse sentiment of texts at a large scale. Decades of research was spent in perfecting sentiment analysis models in colonial languages such as English, but their applicability to low-resource languages have been largely ignored. In developing countries such as India, a large fraction of population use social media to consume information in languages that are usually not the targeted by researchers in the field (such as Telugu, Tamil, Marathi etc.). This thesis aims to participate in research aimed at bridging this gap in Telugu language.

In the next section, we formally examine the problem of sentiment analysis, and its motivation in the field of Natural Language Processing. Specifically, the section elucidates Aspect-Based-Sentiment Analysis in Telugu language, which is the major contribution of the thesis. In the further sections, we describe the major contributions of this thesis and the layout for the rest of the chapters.

1.1 Motivation

Natural language processing is a sub-field of artificial intelligence that deals with the interactions between machines and human languages. It is concerned with the ways in which computers can be made to understand human language and respond in a way that is natural for humans. The ultimate goal of NLP is to make a machine perceive human language to the level of humans, that is, understand the complexities of human communication and respond in a way that is helpful and not frustrating for humans.

[1] https://www.theguardian.com/technology/2021/dec/06/rohingya-sue-facebook-myanmar-genocide-us-uk

For instance, the usage of idioms and sarcasm makes humans struggle to interpret the essence of the text. Interpretation of the meaning of a text can change depending on the context. Understanding the sentiment or emotion in a piece of text often helps iron out the complexities involved. Works like [90] explore detection of such complex constructs.

The emergence and rapid growth of the internet broke multiple barriers and provided people with a platform to voice opinions, communicate with close ones, and access multiple things which were otherwise distant. The availability of this enormous amount of information poses several challenges that lead to continuously evolving research areas. Sentiment Analysis is one such critical area. Sentiment analysis is a process of computationally identifying and categorizing opinions expressed in a text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative or neutral. In the business world, sentiment analysis is used to track the mood of the stock market, customers, or employees. It can be used to monitor social media, review sites, and open-ended survey responses to get a better understanding of the general sentiment around a company, product or service.

For example, if a company is considering launching a new product, it may use sentiment analysis to track how the public is feeling about similar products. If the sentiment is negative, the company may decide to not launch the product. Sentiment analysis can also be used to monitor customer satisfaction. If a company sees that sentiment around their product is negative, they may take steps to improve the product or service. The internet powers users with the freedom of expressing their views through blogs and social media platforms. Processing and analyzing this information to infer user sentiments play a crucial role in assessing users' views about products and also in moderating and providing better experiences to users.

Many sentiment analysis models come at the price of being explainable to human understanding. This makes it hard for human evaluators to assess the performance of black-box models for real-world usage. A well known problem is that the black-box models sometimes derive insights from certain aspects of the text that are not relevant to the underlying sentiment. The real-world deployment of such models can be catastrophic to certain segments of the society. For eg. in [41], it has been demonstrated how hundreds of sentiment anaysis systems have an inherent racial and gender bias. A second problem in these models, is that the sentiment of a text is sometimes conflicting in nature, and this is hard to understand for human evaluators. These problems, to some extent, can be alleviated by performing a finer-level sentiment analysis.

Aspect-based sentiment analysis (ABSA) is a finer-level sentiment analysis paradigm that assigns polarity to each targeted aspect. For example, if you are analyzing sentiment about a restaurant, you might want to consider aspects such as food quality, service, and value. This gives one not only a deeper insight but also fine-grained general feedback about the product, company or service (in the previous example, a restaurant).

Aspect based sentiment analysis (ABSA) is a relatively new field, and as such, there are a number of challenges that need to be addressed. One of the main challenges is the lack of standardization in terms of what constitutes an aspect. This can make it difficult to compare and contrast results from different studies. Additionally, the challenge of dealing with implicit aspects can make it difficult to obtain accurate results. Implicit aspects are those that are not explicitly mentioned in a text but can be inferred. This becomes even more challenging in longer texts as there may be multiple aspects mentioned and it can hard to determine which is the most important one to focus on. Additionally, once the correct aspect has been identified, it can be challenging to accurately extract sentiment from text, as the context around the aspect can be important in understanding the overall sentiment.

Research in ABSA has made decent progress recently. Properly structured annotated datasets like [72] are key to this. Many research works in ABSA in English have treated this as a benchmark dataset to evaluate their performance enhancements.

Telugu is a complex language with a rich history and literature. It is also the fourth most spoken language in India and sixteenth across the world. It has a unique grammatical structure, and its vocabulary includes many words with multiple meanings. This can make it difficult for machines to accurately process and translate Telugu text. There is a lot of web content being generated in Telugu every day. Despite this, Telugu is considered a low-resourced indic language. As the web content that is getting generated is not transitioned into annotated corpora in most of the NLP tasks, including ABSA. NLP in Telugu itself is a challenging task due to its free word order and agglutinative nature. But, the biggest challenge that hinders the advancement of research in ABSA in Telugu is the availability of the resources. Hence, I wanted to solve this problem of data scarcity in my native language and contribute to advancing the state of NLP models.

1.2 Main Contributions

Novel dataset for SA in Telugu

We have created a unique dataset with Telugu movie reviews in Telugu language for ABSA with over 5000 sentences and 90k tokens. It consists of annotated data for three tasks, namely, aspect term detection, aspect polarity classification, and aspect categorization. It helps improve ABSA in Telugu and brings it on par with high-resourced Indian languages like Hindi.

Framework for data collection, annotation, data analysis

Annotation tools and frameworks reduce the laborious human effort of annotating by a large extent. To address the need, we created an annotation framework to streamline the data collection and annotations. The framework is used to build a language-agnostic web app that can be used to clean and annotate data for other languages with ease.

Baselines for ABSA in Telugu

We performed several experiments with different combinations of embeddings with LSTM, out of which the combination of LM-LSTM + CRF with Fastext embeddings proved to be the best with 83.1 F1 score, for aspect term extraction task. For aspect based sentiment analysis and aspect term categorization tasks, we experimented with different variations of attention and LSTM combinations, memory networks, out of which TC-LSTM gave the best results with F1 scores 72.32 and 73.36 respectively. With this establishment of the baselines for the above mentioned tasks, we demonstrated the reliability and usefulness of the resource.

Fine-tuning Pre-trained models to enhance ABSA

We further performed experiments by finetuning the pre-trained language models viz. BERT and XLMRoberta on our dataset for all the three tasks. We compared both the pre-trained models. BERT was trained only on English language, whereas XLMRoberta is a pre-trained model which is exposed to data from multiple languages including Telugu and its typologically similar languages. XLMRoberta proved to be the best overall performer and has shown a significant improvement over the baseline models we set.

1.3 Layout of the thesis

In this section, we describe the structure of the reminder of the thesis.

In the second chapter, we concise the efforts in the field of SA and ABSA specifically. We discuss about the work done in ABSA and the datasets created for the progress of ABSA. How the progress of ABSA has been hand in hand with the evolution of NLP in general. We then discuss about the work done in SA in Telugu.

In the third chapter, we describe about the movie review dataset creation for the tasks aspect term extraction, aspect based sentiment analysis and aspect term categorization, the framework we built to do that. We also discuss about the challenges we faced.

In the fourth chapter, we discuss about the machine learning and deep learning methods we used to evaluate the dataset for all the three tasks. We consolidate the results and analyse them.

In the fifth chapter, we discuss about the advancement of NLP in utilising deep learning techniques for language representation. We describe the experiments performed with pre-trained models, showcase the results and analyse them.

Chapter 2

Background and Related Work

Sentiment analysis identifies the polarity of a given text. It can be at various levels of granularity, and aspect-based sentiment analysis deals with sentiment at the finest level. Over the past decades, it has evolved as an important area in the field of Natural language processing. Research in sentiment analysis has advanced drastically, from getting the sentiment through the number of intent conveying words to using the state-of-the-art deep learning models which represent a language computationally in the best way possible. In this chapter, we discuss the evolution of sentiment analysis and different levels of sentiment analysis. We also discuss the datasets created for various tasks in sentiment analysis and the evolution of the approaches toward tackling it. We also look at the progress that has been made in Indian languages and in Telugu.

2.1 Sentiment Analysis

The same set of words or phrases can carry different connotations based on the context and intention with which they are spoken. In such a scenario, determining the emotion/sentiment associated with the text becomes a crucial, challenging task. With the rise of the internet, there is an abundance of data, and the collection of opinions and reviews has become a lot easier. This has far-reaching implications in various walks of life. For instance platforms like Twitter and Facebook played a major role in electoral outcomes across the globe. Assessing the mood and reaction of the public on these platforms can swing the outcome. Works like [91] explore this dimension. Even from an industrial perspective, for the growth and stability of a business, customer satisfaction is the most important metric. Customer satisfaction can be known only through customer feedback. [66] is one of the first works to collect a set of customer reviews and perform sentiment analysis. They collected movie reviews from IMDB and labeled the data according to the ratings given to the movies by the customer. This was one of the first works to use machine learning models like SVM and naive bayes in sentiment analysis.

The most rudimentary form of Sentiment classification setting is binary i.e classifying the object as positive or negative. Later on, the prevalence of neutral sentences leads to the rise of the neutral class as well. Another similar class is conflict, wherein the object of evaluation has mixed signals. This case advocates for the usage of ABSA as it gives fine-grained insights in such cases. Another perspective to this is quantifying and using relative measures like review ratings across social media.

2.2 Granularity based variations

2.2.1 Word Level

Word is the atomic unit of analysis in textual context. Although at the face of it, it is a trivial task with words having sentiments attached to them, subjective inference of sentiment given a sentence, the document makes it challenging. [39] work explores the word-level sentiment analysis by leveraging the global, local and original sentiment context of the world.

2.2.2 Phrase Level

Incrementally, phrase level is the next segment which deals with sentiment analysis of a group of words in a given context as seen in [97], where initially the deduction of neutral sentiment is done and post that positive and negative sentiment is determined. In [106] phrase-level sentiment analysis is leveraged to improve recommendation systems based on user feedback's sentiment analysis, further validating the multiple applications of Sentiment Analysis.

2.2.3 Sentence Level

Polarity is determined for a particular sentence at this level of hierarchy. Various supervised, unsupervised and lexical approaches have been used in order to tackle SA at sentence level. [44] uses ontology-based techniques to classify twitter sentences, which effectively counters the syntactical inconsistency due to the character limit on Twitter. [50] used Bi-LSTM with an attention mechanism to address sentence-level SA. [31] uses semi-supervised learning to combine HowNet lexicon to train Phrase recursive autoencoders.

2.2.4 Document Level

As indicated in the name, document level analysis assesses the sentiment of the document as a whole. The primary challenge associated with this is cross-theme sentiment identification and aggregation. Works like [65] focus on how to leverage features in the context of themes. In addition, works like [11] delve into domain-specific document-level sentiment analysis.

2.3 Applications

Sentiment Analysis is a crucial piece of knowledge to have in Social media settings and comes in handy to monitor, proctor, and curate content online. Especially in today's context, it is very crucial as Social directly or indirectly influences people's choices ranging from personal shopping decisions to political inclinations. Given this, Sentiment Analysis becomes a quintessential, indispensable piece of technology.

Another crucial application is feedback inference. Multiple companies and their brands receive opinions from the internet, and it becomes imperative for acceptance of product/company to gauge this.

Although not directly, Sentiment Analysis extends its influence to other tasks like recommender systems. As can be seen in works like [47] sentiment analysis is used in multiple forms to aid the rating prediction model. User sentiments on products have been leveraged as features to enhance recommendation systems.

2.4 Aspect based sentiment analysis

Research in sentiment analysis has gained a lot of traction over the years. Many developments have been made since customer reviews were analyzed in [37]. ABSA is one of its branches that gained momentum in recent years due to its analysis at a finer level. Deep learning advancements led to improvements in ABSA. Some of the works in ABSA in recent times are [73], [51], [98], [18], [102]. Most of these works use deep learning methods. But resource creation tasks like [72], and [71] are key for the progress of ABSA.

The lack of reliable resources of this kind restricts Indian Languages from research developments in ABSA. There have been some developments recently in Hindi. In [2], a dataset for Hindi was created, and in [3], an approach was developed for ABSA in Hindi using the created dataset. However, in Telugu, there is no dataset available for ABSA. In Telugu, [1] has a corpus of song lyrics for sentiment analysis. [32] has several products, book, and movie reviews annotated at a document level. [62] and [63] have annotated data for sentiment analysis at sentence level.

As discussed in the above sections, sentiment analysis was done at lexical, sentence, and document levels. ABSA is a finer-level sentiment analysis where polarity towards each aspect is calculated. [37] was the first work to explore sentiment analysis at this level. This work added a different dimension to traditional sentiment analysis by mining the features about which customers have expressed their opinions. This led to an outbreak of research in Sentiment Analysis. and thereafter many approaches have been developed for ABSA since then.

ABSA primarily consists of two tasks namely, Aspect Term Extraction(ATE) and Aspect Sentiment Classification (ASC). ATE can be formulated as a token classification or sequence labeling problem, which was the base for many approaches. On these lines, different neural network models like CRF, RNN, and CNN have been used to tackle ATE.

2.4.1 Aspect Term Extraction

This sub-task deals with identifying aspects of a given subject of evaluation. For example in a review like “Cinematography is top-notch in Prince, but sloppy screenplay makes it a mediocre watch”, there are two aspects that are being discussed i.e “Cinematography” and “screenplay”.

The objective of ATE is to identify these terms. Generally, this task is projected as a supervised problem in the likes of sequence labeling tasks like NER. For the current work of interest as well, the same paradigm was leveraged. As a natural choice, CRF models have been tried on this task in [105]. [100] explore the initial set of Neural network models like RNNs, and CNNs to tackle this problem. Post the attention shift in NLP, works like [99] explore novel post-training approach on BERT and establish the generalizability of this approach across datasets.

As CRF is one of the traditional methods for sequence labeling tasks, initial efforts to assess ATE were mostly CRF based. [72], uses CRF as the baseline to evaluate their data set, [103] extends semi-CRF model [81] for ATE. It was the state-of-the-art before RNN-based models were used for ATE. Although CRFs are often successful, they require a lot of work to create the feature set and feature function expansion. This often happens task-by-task, so a lot of engineering effort is needed.

However, the availability of data is a very big challenge, and to overcome this data augmentation methods on top-quality annotated data have been explored in [105]. In this zone, [94] explores the self-training paradigm to solve the data problem. However, this poses the challenge of pseudo-labels, which have been overcome by using curriculum training to identify pseudo-labels at each iteration. In [34], attention-based autoencoders have been used to tackle the problem wherein aspect coherence is optimized by making use of word-to-word embedding relationships. Attention is utilized to identify the less relevant or noisy portions of the subject.

2.4.2 Aspect Category Detection

This sub-task involves the categorization of aspects into pre-defined buckets, largely used in a domain-specific setting. This task can be regarded as an aggregation/generalization on top ATE, wherein implicit, predefined categories are also extracted. In a review like “Not worth your time and money.”, the price can be one of the aspect categories. Although it’s not explicitly mentioned, a pre-determined set of categories is used to assign multiple labels (i.e categories) to the given subject. Similarly to ATE, ACD has also been explored in supervised and unsupervised contexts. In a supervised setting, ACD is formulated as a multi-label classification problem. [61], [82] explore ACD problem from a supervised perspective.

The unsupervised treatment of this task feels more intuitive and extensible from ATE. Once the aspects are extracted from ATE, these are clustered to identify the umbrella categories they fall into. Alternatively, these terms can be mapped or classified into a pre-existing set of categories. [34] explores this and maps the identified aspect terms. Works like [92] step further and use word embeddings of aspect terms and aspect categories to establish this mapping.

2.4.3 Aspect Sentiment Classification

Aspect Sentiment Classification sub-task is a follow-up to the above two tasks of ATE and ACD. Once the aspect terms and categories are identified, ASC determines the sentiment associated with them. The choice of using term/category or both don't change the setting of the problem much and can be treated equivalently. However, aspect terms are literally present in the subject, and hence we can make use of their position in assessing their sentiment. Doing the same for aspect categories is an involved task as they can be mapped to multiple aspect terms.

Initial works like [15] explore ASC using basic linguistic features like word lemmas, and word frequencies. Other works like [42] leverage lexicon-based approaches to tackle the ASC problem.

With the onset of deep learning, there have been a lot of advances in ASC. [26] uses Recursive neural networks to arrive at parse-tree-like structures and use that for determining the sentiment of the aspect. [87] introduces two LSTM-based frameworks, TD-LSTM and TC-LSTM with slight modifications to the original model, where TD-LSTM where the context before and after the aspect term is leveraged as a feature whereas TC-LSTM extends it further by introducing a component which captures the connection between the aspect term and each context word. Both methods proved to be effective in ASC.

Apart from these, the attention mechanism in different settings of DNNs is used to assess ASC as the intuition is that different parts of the sentence pay attention to the opinion towards the aspect term. [54] leverages attention which is learned interactively to generate representations for the sentence and aspect term respectively. [30] provides a fine-grained attention mechanism to tackle multi-word aspect terms and designed an aspect alignment loss to identify the interactions between aspect terms and context.

CNN is another popular DNN architecture that was effectively adapted from computer vision. Various CNN-based frameworks have been developed to solve NLP challenges, similarly, it has been done in ASC as well. [49]l employs a CNN layer to extract salient features from the transformed word representations originating from a bi-directional RNN layer. [38] introduces a novel parameterized convolutional neural network to tackle ASC.

Another dimension along which research has progressed in ASC is Memory networks. [88] proposes a setup of sequential layers, each using neural attention models over external memory. [17] combines attention and recurrent neural networks non-linearly. The weighted memory mechanism is used to do efficient feature engineering and for custom memory for opinion targets. [16] used aspect based discrete opinion trees with aspect-context attention scores as the syntactic distances. The scores were obtained using reinforcement learning with a novel attention-based regularization.

As with the recent trend in NLP post [25], ASC has also seen exploration with task-specific fine-tuning, however, [101] proposes an additional post-tuning wherein domain-specific data is used under objectives similar to pre-training i.e MLM and NSP. By doing this, the bias from non-domain data in pre-training comes down and is seen to help with the eventual performance of the task. [8] is another work that utilizes BERT and proposes a novel approach to extract aspects with their span and then classify their polarity.

2.4.4 Datasets

This section describes various datasets that have a pivotal role in the field of ABSA and also those which have influenced my work. Opinion is an inseparable part of Sentiment analysis, and naturally, reviews become a central piece in Sentiment analysis. One of the first publicly available datasets is Amazon's reviews dataset, which was used in the ABSA context in [37]. [89] has a collection of service and university reviews, annotated in an opinion-centric manner. Another dataset that was instrumental in advancing the field is Twitter comments data. Works like [26] have used this dataset in the context of ABSA.

Sem Eval tasks in 2014 and 2016 laid foundations for proper dataset creation to foster research in aspect based sentiment analysis. The dataset was in English. The first task in 2014 helped identify Aspect terms, Aspect categories, and their polarities. Two datasets were created for laptop reviews and restaurant reviews respectively, consisting of around 3800 sentences each. This was the benchmark dataset with a significant number of sentences and quality in the area of Aspect Based sentiment analysis, which paved the way for research.

The Semeval task in 2016 was one of the continuations to the semeval tasks of 2014. This task helped extend the dataset creation for aspect based sentiment analysis in 8 other languages. The datasets were created in 7 different domains, each language not necessarily containing datasets related to all domains. The same annotation guidelines were framed for datasets in different languages in a domain. It gave me the chance to explore the cross-lingual or language-agnostic approaches.

[2] established a benchmark for aspect based sentiment analysis in Hindi. This work inspired benchmarks for several other Indian languages. The dataset includes subtasks of aspect term identification and aspect sentiment classification. The dataset also features 5000 review sentences overall from 12 different domains. [10] creates a gold standard for ABSA in Hindi and enhances the quality over [2]. They demonstrate it by using the same DNN models on both datasets and find the same models perform better on their dataset. [57] created datasets in cricket and restaurant domain in Bengali which contain 2900 and 2800 sentences respectively.

2.5 Sentiment Analysis in Indian Languages

This section describes the setting of Sentiment analysis and specifically ABSA in the lens of Indic languages. [9] provides a lexicon for the Hindi language in the context of sentiment polarity. Adjectives and verbs were scored based on their polarities leveraging Hindi wordnet and additionally, a product review dataset has been annotated. [74] explores the challenging problem of cross-lingual sentiment polarity detection. They make use of wordnet connections available across languages to overcome this challenge. This approach has shown significant improvement on top of the trivial method of using bilingual dictionaries. [68] explores Twitter data's sentiment analysis in Hindi, Bengali, and Tamil. This work is one of the primal works to advance data and research in Indic setting. [28] and [27] propose approaches for code-mixed sentiment analysis and demonstrate it with experiments on Dravidian language datasets. [19] creates twitter based SA corpus for several low resource languages Telugu, Bengali and Hindi. The corpus creation methodology can be extended to any resource-scarce languages. They also propose an approach for multilingual emoji prediction.

Coming to ABSA in Indian languages, [2] created a dataset for ABSA in Hindi in different domains and set the baselines using different ML models. [33] proposed a rule-based approach for ABSA in Hindi. [4] proposed a Bi-LSTM-based approach which is language agnostic and experiments were done in Hindi as well. [5] proposes an approach to improve word embedding coverage using bilingual embeddings and does a case study with ABSA. In [67], BERT ensembles are used to experiment with ABSA Hindi with data from different domains. [57] proposed a CNN-based approach for ABSA in Bengali.

2.5.1 NLP in Telugu

Telugu is a low-resource language in terms of NLP. Several works have either tried to improve the resources or tried to adapt contemporary NLP technologies to Telugu in different areas. One such domain which has seen light is sentiment analysis. [56] is one of the recent works in which the problem of resource scarcity in Telugu is addressed both in terms of datasets and pre-trained models. They have created annotated datasets for tasks like sentiment analysis, emotion identification, hate speech detection, and sarcasm detection. They also created lexicons for the above tasks. They developed pre-trained language models ELMo-Te, BERT-Te, RoBERTa-Te, ALBERT-Te, and DistilBERT-Te which can be leveraged to finetune several downstream tasks. [62]is the first work in sentiment analysis in not only Telugu but also Dravidian languages. Machine learning techniques like Naive Bayes, SVM, Linear regression, and random forest were implemented to assess sentiment analysis on a corpus that consists of 1644 sentences. [1] was one of the first works in Telugu to explore multi-modal sentiment analysis. Audio features which incorporate the mood, emotion and acoustics of the song were used along with the textual lyrics of the song to compute the sentiment of a given song. [32] is one of the first works for document-level sentiment analysis. Data was collected for different domains like book reviews, movie reviews, product reviews and song lyrics. Building Cross-domain and generalized classifiers for all domains were built. [63] created a corpus consisting of 5000+ sentences, to make it the largest dataset for SA in Telugu at that time. [46], [79], [43] present approaches for Telugu-English code-mixed SA. [45] and [40] are other works on SA in Telugu. All the above works focus on sentiment analysis either at document level or sentence level or word level. Our work is the first in Telugu on ABSA.

Other areas of NLP include developing tools for Telugu. [7] developed a POS tagger for Telugu, [64], a POS tagger on Telugu-English code-mixed social media data, and [86], a rule based morph analyzer for Telugu.

In the area of Dialogue Systems(DS), [84] created a corpus for DS in Telugu and proposed a rule based approach. [24] built an end-to-end DS using memory networks. [55] developed a spoken Telugu DS on agricultural data. [29] created a DS implementing question classification strategy using ML techniques and processing it to an SQL query to retrieve the required information, for Telugu hospital domain data.

[13] presents a methodology for question answer pair generation from Telugu short stories and [14] makes use of it to create a learning assistant in Telugu.

Chapter 3

Dataset creation

In this chapter, we describe the steps for the creation of annotated corpus, brief about the tasks for which data is created. We also provide glimpses of the annotation tool we developed and the samples of data.

3.1 Introduction

The emergence and rapid growth of the internet made the availability of information easily accessible. The availability of this enormous amount of information posed several challenges that led to new research areas. Sentiment Analysis is one of them. The internet powers users with the freedom of expressing their views through blogs and social media platforms. Processing and analysing this information helps in attaining an adequate understanding of customers' opinions. Sentiment Analysis (SA) enables practical judgment of a product/service by assigning polarities to the reviews given by its users.

We can perform sentiment analysis at different levels. Aspect Based Sentiment analysis (ABSA) is a finer level sentiment analysis that assigns polarity to each targeted aspect instead of the entire review to increase the level of understanding. An Aspect can be a property or component of the product/service. ABSA involves the tasks of (a) Identifying the Aspect terms in the sentences of the user review and (b) Assigning polarity to the identified Aspect terms. There can be more than one Aspect term in a sentence.

There are many approaches developed to assess ABSA and its challenges in recent works. But most of these works are limited only to a few languages, where a majority of them are in English. Very few of them focus on Indian languages.

Even though Telugu is the fourth most spoken language in India and sixteenth across the world, aspect-based sentiment analysis in Telugu remains unexplored. There are several challenges with Telugu. It is a morphologically complex language due to its agglutinative nature. The biggest challenge that hinders the advancement of research in ABSA in Telugu is the availability of the resources for the research. Linguists created some resources in Telugu for Sentiment Analysis (SA), but they address SA only at coarser levels like document and sentence levels. In this chapter, we describe the creation of an annotated dataset for ABSA in the movie review domain and pave the way for further research advancements in ABSA in Telugu.

The annotated data is created for three tasks, namely,

- Aspect Term Identification
- Aspect Polarity Classification
- Aspect Categorisation

In the first task, we identify the aspect terms in the review sentences. In the second, polarity is assigned to each identified aspect term whereas the third task is to classify each aspect term into a category. We fix the number of categories into which the aspect terms are classified so that the polarity of these abstract categories can be obtained. We perform several experiments using deep learning methods to demonstrate the usage of the dataset. We analyse the results and arrive at baselines for all the three tasks in the created corpus.

3.2 Data Creation and Annotation

In this section, we gave a detailed description of dataset creation and the challenges faced. In this section, we use the transliterated form of Telugu along with their English translations to list out the examples. It is to ensure better readability. However, the original dataset is in Telugu Script.

3.2.1 Data Scraping and Cleaning

We crawled several movie review websites such as 123Telugu.com, eenadu.net, Telugu.samayam.com. Initially, there were 10000 sentences from the scraped data. The raw data contained undesirable characters and sentences. For example, English words and URLs appeared in the middle of the sentences. We automated the process of removal of such words and characters. We eliminated the sentences which were used to describe the story of the movie as they have no opine towards the movie manually. We removed unnecessary lines like side-headings from the reviews. We corrected spelling mistakes and punctuation marks wherever necessary. There were 5027 review sentences after these pre-processing steps. All the statistics of the data are described in the section 3.2.3

3.2.2 Data Annotation

Annotated data is created for three tasks, (i) identifying the aspect terms in each sentence, (ii) assigning polarity to each aspect term, either *positive*, *negative* or *neutral* and (iii) categorising the aspect term into one of the six categories, viz. *story*, *acting*, *direction*, *music*, *technical* and *general*. Two annotators, who are native Telugu speakers, are asked to annotate the data for these tasks. We provided them with clear guidelines as shown below:

- The annotators read and understand every review sentence carefully.
- They identify the aspect terms in the sentence.
- For each aspect term present in the sentence, they provide the polarity and the category.

A sample annotation data is shown in the figure 3.6 . We developed a tool to facilitate the task of annotation. The tool is shown in the following Figure 3.1. The flow of annotation using the tool is shown in the figures 3.3 and 3.5

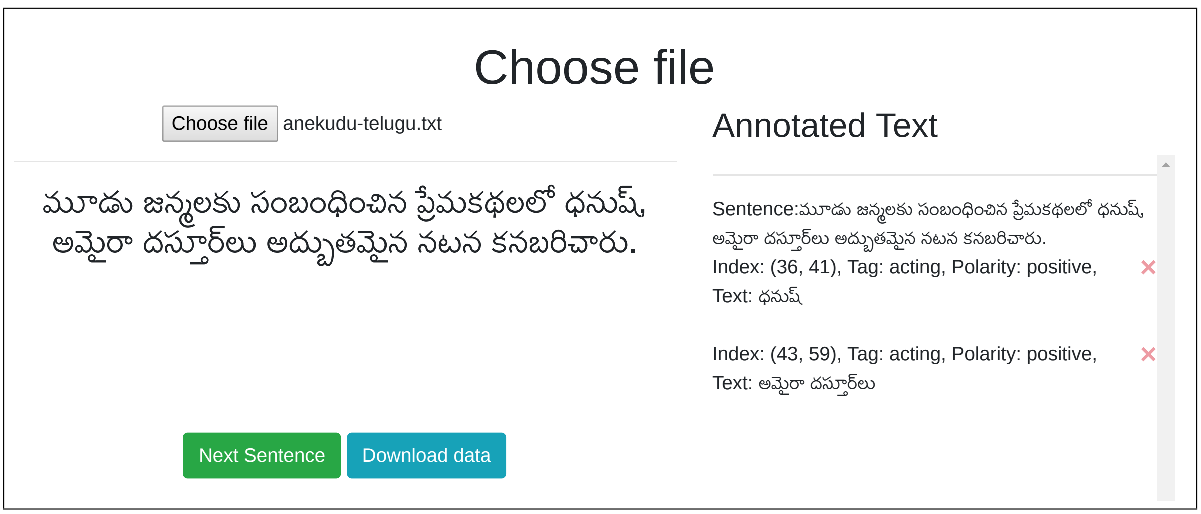

Figure 3.1 annotation tool

Aspect Term Identification: Annotators are handed over the task to identify the aspect terms in all the sentences. There is a possibility of having multi-word aspect terms and multiple aspect terms in a sentence. Identifying multi-word aspect terms is sometimes a challenging task. Consider the example sentence, “ee cinema lO allu arjun bAgA naTinchADu.” (*In this movie, Allu Arjun acted well.*). “allu arjun” is the aspect term. Whereas in the sentence “ee cinema lO allu arjun naTana AkaTTukundi”. (*In this movie, Allu Arjun’s acting has impressed.*), “allu arjun” is a modifier. Hence, the head of the noun phrase, “allu arjun naTana” (*Allu Arjun’s acting*), “naTana” (*acting*) is the aspect term. Modifiers are treated in the same way as the adjectives. They are not included in the aspect term. Another challenge is that, few nouns in a sentence can be mistaken for aspect terms.

Identifying the aspect terms is considered as a sequence labeling task. Labels, either ‘B’, ‘I’ or ‘O’ are given to each word. A word is tagged ‘B’ if it is the beginning of an aspect term. It is tagged ‘I’ if it is inside an aspect term and is tagged ‘O’ if it is not part of any aspect term. An example is shown in the

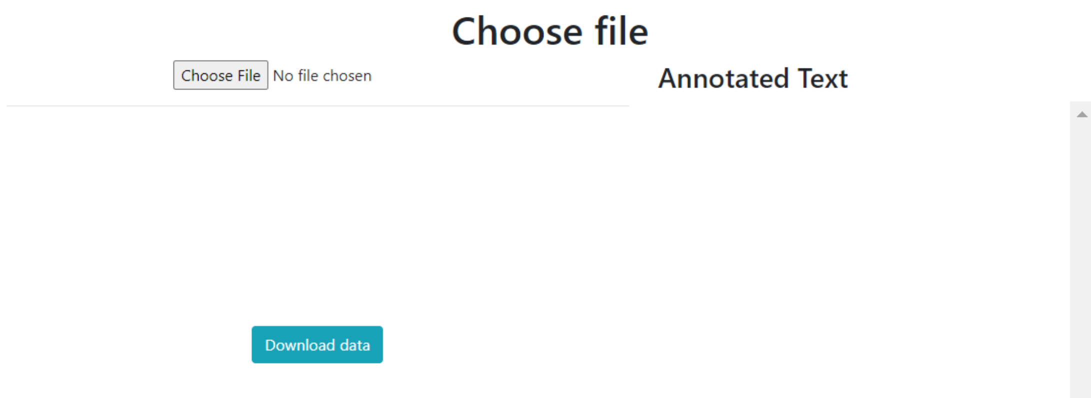

Figure 3.2 On clicking "Choose file" button, one can select a file that contains movie review sentences. One sentence appears at a time on the left pane, from which aspect terms can be selected and polarity and category can be assigned to it.

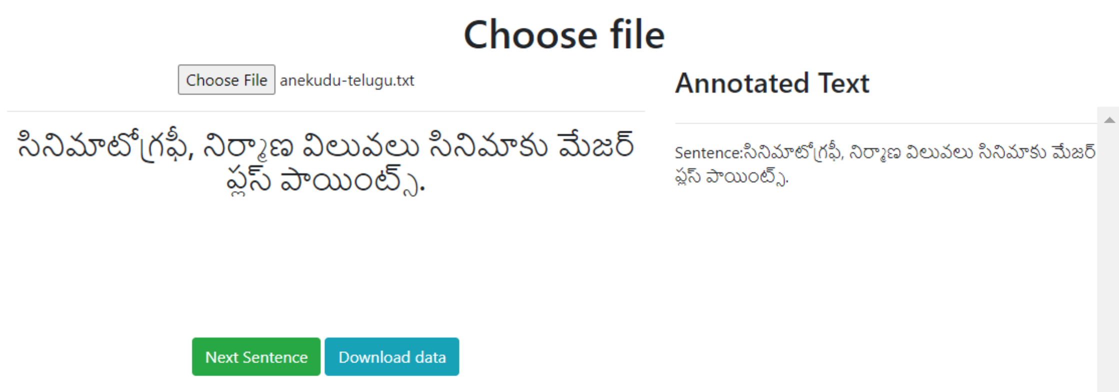

Figure 3.3 On selecting a file, the sentence to be annotated appears in the center pane as shown above

Figure3.7. We use Cohen's Kappa Coefficient [20] to determine the inter-annotator agreement between the annotators. The kappa coefficient is calculated as follows:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{3.1}$$

where P_o is the observed proportionate agreement between the annotators and P_e is the random agreement probability. The kappa score was found out to be 93.57% which endorses the acceptable quality of the annotated dataset. Disagreements and discrepancies in annotation were resolved through discussions between the annotators.

Polarity Classification and Aspect Categorisation: Assigning polarity to the identified aspect terms is the second task. Each identified aspect term is annotated with its polarity. A few examples are provided in the Figure 2. The inter-annotator agreement was measured using the kappa score which is 97.13% for the polarities . The score shows that the dataset is reliable and comprehensive.

The identified aspect terms are annotated with categories in the third task. Aspect category, similar to polarity, is assigned as a property to the aspect term. Each aspect falls into a category. For example, in the sentence “allu arjun naTana, trivikram darshakatvam, devi sri prasAd andinchina pAtalu, ee cinema ki pradhAna AkarshaNa gA nilichAyi.” (*Allu Arjun’s acting, Trivikram’s direction and songs given by Devi Sri Prasad are the highlights in the movie.*), the Aspect “natana” (*acting*) falls into the category of *acting*, “darshakatvam” (*direction*) falls into the category of *direction* and “pAtalu” (*songs*) falls into the category of *music*. Aspect terms like “cinematography”, “screenplay” , “visuals” fall into the category of *technical*. Other examples are shown in the Figure 3.6. The kappa coefficient for this task was measured to be 96.72%.

3.2.3 Data Description and Statistics

The sentences in the dataset are in Telugu Script. The annotated data for aspect term identification is in the form of *word/tag*. Annotated data for aspect polarity classification and aspect term categorisation is in JSON format. Each JSON file contains an array of JSON objects. Each object consists of two properties, *‘sentence’* and *‘aspectTerms’*. The property *‘sentence’* has the review sentence. The property *‘aspectTerms’* has an array of JSON objects where each object has the properties of *‘aspectTerm’*, *‘start’*, *‘end’*, *‘polarity’* and *‘category’*. *‘aspectTerm’* has the aspect term itself. *‘start’* and *‘end’* have the beginning and ending indexes of the aspect term. *‘polarity’* and *‘category’* are the annotated polarity of the aspect term and the annotated category of the aspect term respectively. Figure 3.8 shows a glimpse of the dataset.

The dataset contains 5027 sentences and 92848 tokens. The total number of aspect terms in all those sentences is 7130. Of which, 3521 aspect terms are of positive polarity, 2480 are of negative polarity and 1129 are of neutral polarity. The number of neutral aspect terms is relatively low. This is because most of the reviewers tend to describe more about the positive and negative elements resulting in more number of positive and negative aspect terms. The statistics are reported in the Table 1. The aspect terms are distributed uniformly across all other categories except the general category. The reason for large number of aspect terms of general category is that there are many elements in a movie which are described about like production values, scenes, dance, locations etc., which cannot be put into any of the other five categories. Though all these elements add value to the movie, individually they cannot define a movie.

3.3 Summary

In this chapter, we documented the process of creating the annotated data for three tasks namely, Aspect term identification, Aspect polarity classification and Aspect category detection. We described different steps data pre-processing, annotation, challenges faced and provided glimpses of the data an-

Number of Sentences		5027
Number of tokens		92848
Aspect Terms	#positive	3521
	#negative	2480
	#neutral	1129
	total	7130
Aspect Categories	#Story	548
	#Action	603
	#Direction	301
	#Music	382
	#Technical	554
	#General	4742
	total	7130

Table 3.1 Dataset Statistics. #w denotes the count of w.

notation tool and the dataset with examples. In the next chapter, we perform several experiments on the created dataset to demonstrate its usability.

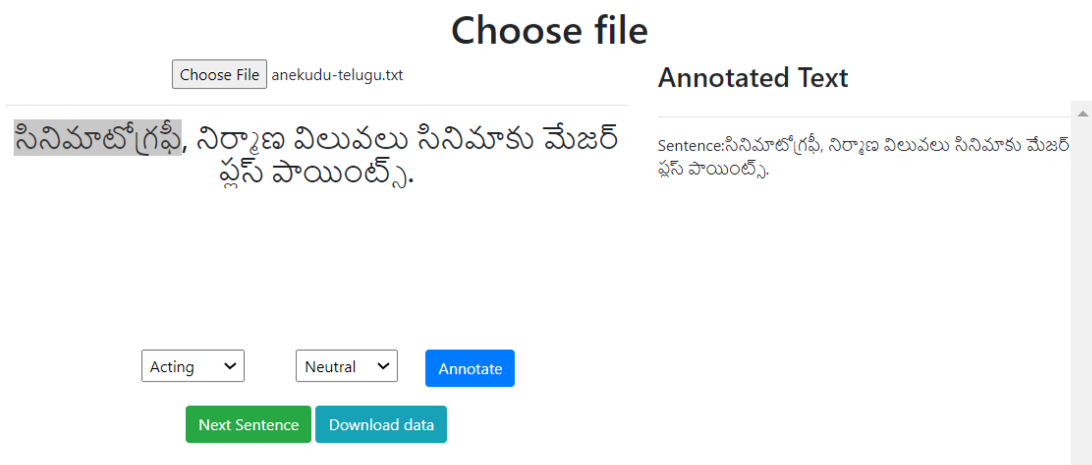

Figure 3.4 In the sentence that is to be annotated, aspect term needs to be selected. On selecting, dropdowns appear at the bottom, from which the the category and polarity can be assigned it, on clicking annotate, the aspect term is included on the right pane.

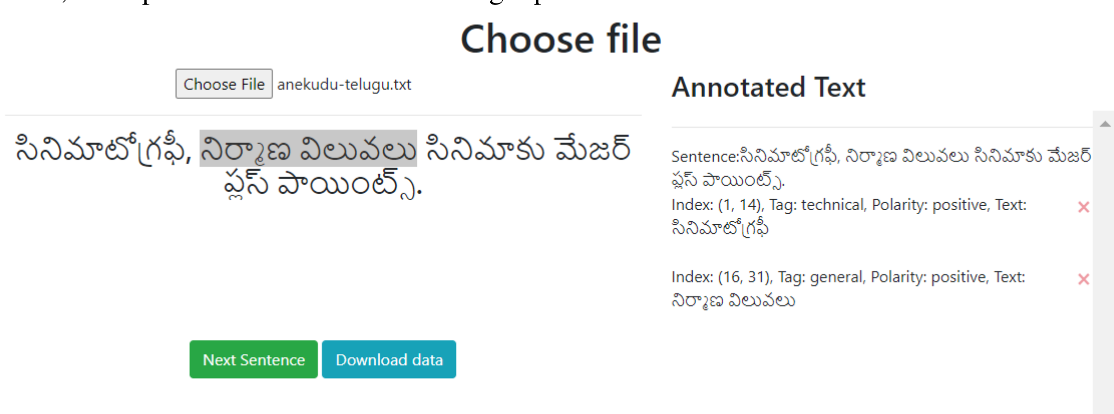

Figure 3.5 One can choose to remove the annotated sentences on the right panel and once finishing annotating the file, the Download data button can be used to download the annotated sentences as a json file

	Review Sentences	Aspect Terms	Polarity	Category
Telugu Script **Transliterated** **English**	ఈ సినిమా లో లొకేషన్లు అందంగా ఉన్నాయి. ee cinema lO locationlu andangA unnAyi. In this movie, locations are good.	లొకేషన్లు (locationlu)	positive	general
Telugu Script **Transliterated** **English**	ఈ సినిమా కథ బాగున్నా, స్క్రీన్ప్లే సినిమా వేగాన్ని దెబ్బతీసింది. ee cinema katha bAgunnA, screenplay cinema vEgAnni debbatEsindi. Even if the movie's story is good, Screenplay affected the movie badly.	కథ (katha) స్క్రీన్ప్లే (screenplay)	positive negative	story technical
Telugu Script **Transliterated** **English**	రత్నవేలు కెమెరా వర్క్ ఆకట్టుకోగా, థమన్ బ్యాక్గ్రౌండ్ స్కోర్ పర్వాలేదనిపించింది. ratnavElu camera work AkaTTukOgaa, thaman background score parvAlEdanipinchindi. Ratnavelu's work with camera has impressed where thaman's music is okay.	కెమెరా వర్క్, (camera work) బ్యాక్గ్రౌండ్ స్కోర్ (Background score)	positive neutral	technical music

Figure 3.6 Examples for ABSA in Telugu

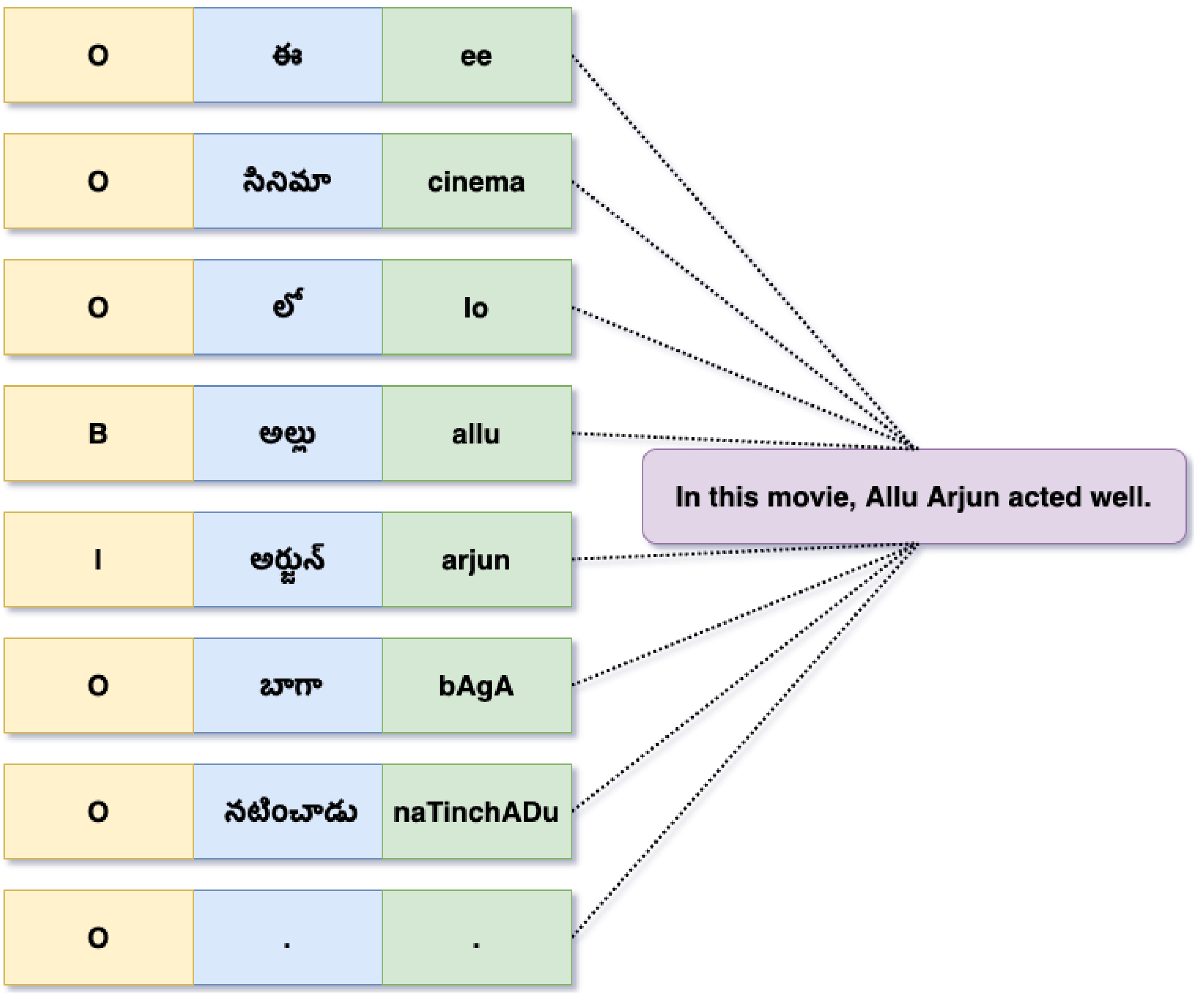

Figure 3.7 Example for Aspect Term Identification

```
[
...,
 {
     "sentence": "రత్నవేలు కెమెరా వర్క్ ఆకట్టుకోగా, థమన్ బాక్గ్రౌండ్ స్కోర్ పర్వాలేదనిపించింది.",
     "aspectTerms": [
       {
         "aspectTerm": "కెమెరా వర్క్",
         "start": 10,
         "end": 22,
         "category": "technical",
         "polarity": "positive"
       },
       {
         "aspectTerm": "బాక్గ్రౌండ్ స్కోర్",
         "start": 40,
         "end": 60,
         "category": "music",
         "polarity": "negative"
       }
     ]
},
...
]
```


Figure 3.8 Dataset annotation structure

Chapter 4

Evaluation of dataset

In this chapter, we experiment with several deep-learning models for the three tasks we created the dataset. We used different versions of LSTM in combination with CRF for aspect term identification. For aspect polarity classification, apart from ML models like naive bayes and SVM, we used TC-LSTM, TD-LSTM, ATAE-LSTM, and Interactive attention networks and Memory networks. We use the same above models for the third task aspect category detection. We describe the experimental setup for all the above models, document the results, and provide an analysis over the achieved results.

4.1 Method for Aspect Term Identification, Aspect Polarity Classification and Categorisation

In this section, we describe the models that we used for aspect term identification, aspect polarity classification, and aspect categorization. We experimented using various deep learning models along with traditional approaches. For all the three tasks, the models are implemented without using any domain-specific or language-specific external resources or tools.

4.1.1 Aspect Term Identification

An aspect term in a sentence can range from a single word to multiple words. Also, a sentence can contain multiple aspect terms. Hence, we treat the task of aspect extraction as a sequence labeling task and the dataset is annotated accordingly. The task is to label each word of the sentence with 'B', 'I', 'O' tags which denote the beginning, inside and outside respectively. We experimented with multiple sequence labeling models like LSTM+CRF, bi-LSTM+CRF[6]. The best performance was obtained when we used the Language Model - Long Short-Term Memory - Conditional Random Field (LM-LSTM-CRF) model[52]. This model has been successful in solving sequence labeling tasks like POS tagging, Named Entity Recognition, etc.

Figure 4.2 shows the architecture of LM-LSTM-CRF model. This model augments sequence labelling by concurrently training it with language models. As Telugu is an agglutinative language, we

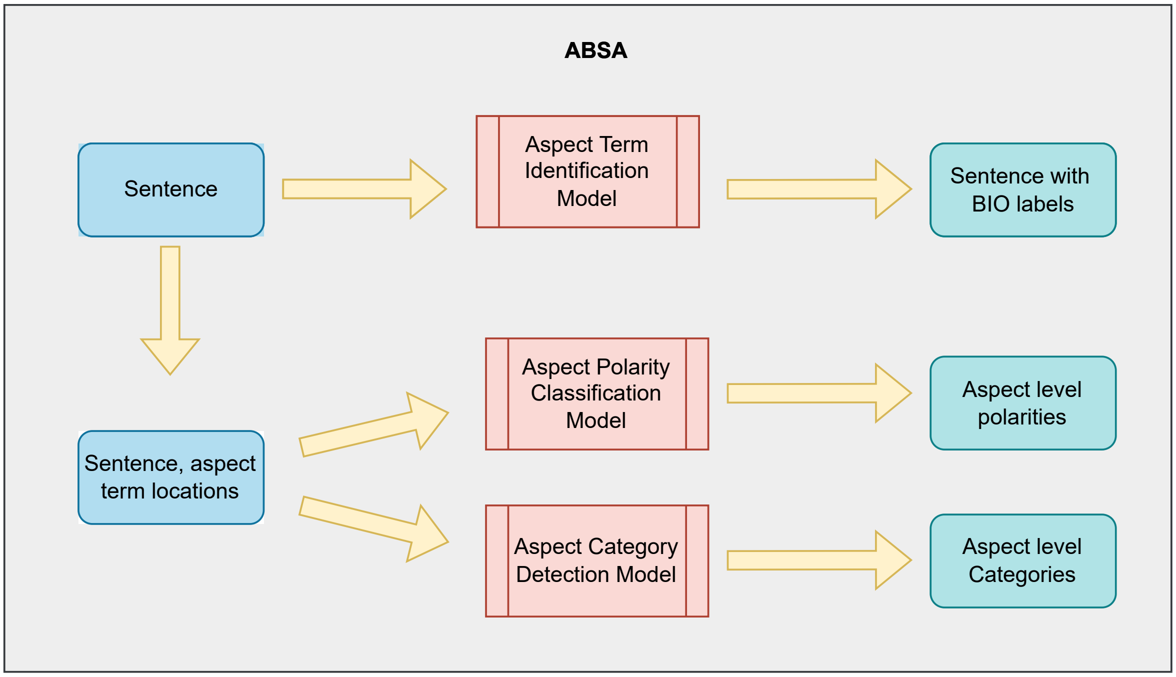

Figure 4.1 Framework for ABSA evaluation

need both character and word level information. Hence the LM-LSTM-CRF model fits our task as it incorporates:

- **Word level information:** It uses word embeddings to capture word level information in a sentence.

- **Character level information:** The model incorporates character level information using character level bi-LSTMs. The information handles complex agglutinative words formed by stringing morphemes together. This cannot be attained by using word embeddings alone.

- **Contextual level information:** The bi-LSTM layer takes concatenation of word level and character level features and extracts contextual information from the sentence.

- **Language Model:** Language model trained concurrently also helps in extracting character level knowledge from the self-contained order information.

- **Conditional Random Fields:** The model uses Conditional Random Fields which predicts labels not just based on the current word but also on its neighborhood which is very important for aspect extraction.

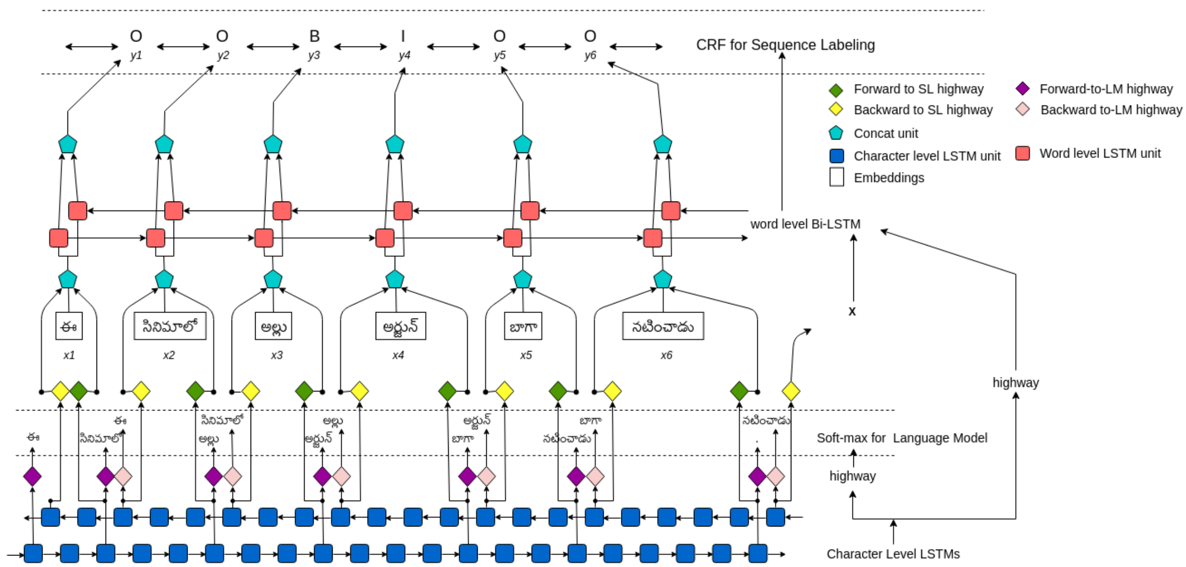

Figure 4.2 LM-LSTM-CRF architecture

4.1.2 Aspect Term Identification and Polarity Classification

As both aspect categorisation and polarity classification are classification tasks, we experimented using similar models by changing the last output layer alone. For these tasks, various deep learning models along with traditional approaches are experimented to set the baselines.

4.1.2.1 SVM

Support vector machine based classifier is used including following different features.

- **TFIDF+Unigrams** The TFIDF values of bag of Unigrams are used as features.
- **TFIDF+Bigrams** The TFIDF values of bag of Bigrams are used as features.

SVMs are known to be efficient in solving text analytics problems[22]. For our classification tasks, we append the aspect term to the sentence and then provide it as input to SVM for both training and prediction.

4.1.2.2 Naive Bayes

We concatenate the aspect term on both sides of the sentence. We use the TFIDF representation of the resultant sentence as features. We provide these features to Bernoulli Naive Bayes classifier which has been found to perform well in text-related domain [77] to classify the aspect term.

4.1.2.3 LSTM

LSTM model architecture as in [87] is used. We append the aspect term to be classified on both sides of the sentence. We parse it and provide it as input to the embedding layer. The output of the embedding layer is passed through an LSTM layer which is used to classify the corresponding aspect term.

4.1.2.4 Target-Dependent LSTM (TD-LSTM)

Target-Dependent LSTM model [87] provides the preceding and following contexts surrounding the aspect term as a feature representation to the model. The use of surrounding contexts would improve the accuracy of target-dependent classification. Figure 4.3 shows the architecture of the model.

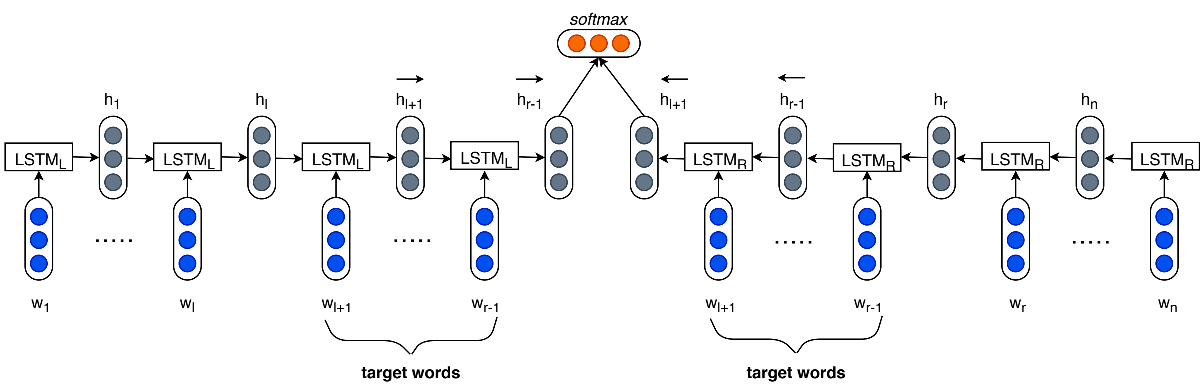

Figure 4.3 TD-LSTM model architecture

4.1.2.5 Target-Connection LSTM (TC-LSTM)

Target-Connection LSTM [87] extends the idea of TD-LSTM by incorporating aspect connection. It utilizes the connection between each word and aspect words when forming the representation of the sentence. Figure 4.4 shows the architecture of the TC-LSTM model.

4.1.2.6 Attention-based LSTM with Aspect Embedding (ATAE-LSTM)

ATAE-LSTM appends the aspect embeddings with each word embedding vector to represent the context. These concatenated embeddings are passed through LSTM networks and then the hidden states are combined with aspect embeddings to supervise the generation of attention vectors. The attention vectors are used to produce final representation for aspect term classification. Figure 4.5 shows the architecture of the ATAE-LSTM model.

[96]

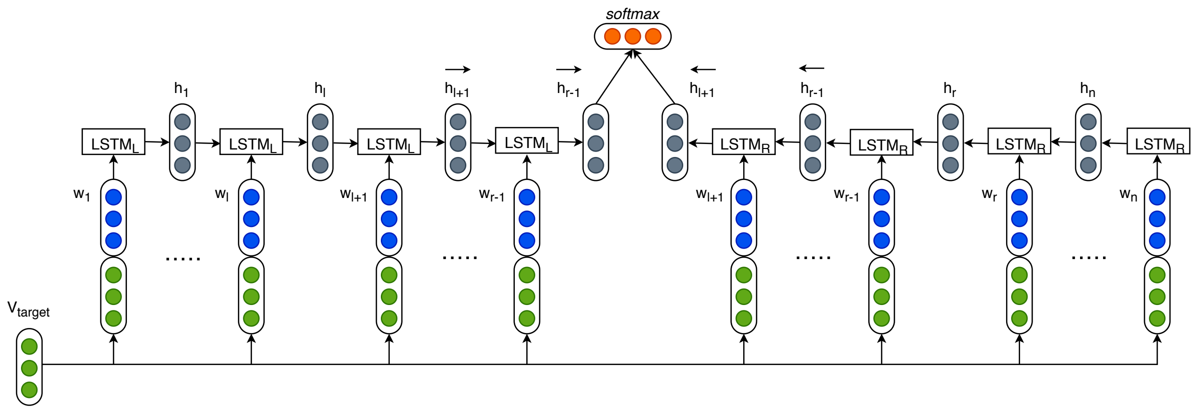

Figure 4.4 TC-LSTM model architecture

4.1.2.7 Interactive Attention Networks (IAN)

IAN [54] models aspect and context interactively. It uses aspects' hidden states and context's hidden states to generate supervised attention vectors and captures important information from aspect and context. With this design, aspect and context influence the prediction interactively.

4.1.2.8 Deep Memory Networks

[88] Deep Memory Networks capture the importance of each context word when referring to the classification of an aspect. Text representation and degree of importance are calculated using the neural attention model over an external memory. Figure 4.6 shows the architecture of the Deep Memory Network model.

4.2 Ablation studies

We compared the performance of the above-mentioned models to set baselines for all the three tasks in ABSA. We performed a 5-fold cross-validation to report the final results. We experimented with different types of embeddings:

- **Random Embeddings:** The word and character embeddings are randomly initialized and are fine-tuned during the training of the model.

- **Byte-Pair Encoding (BPEmb):** BPEmb is a collection of pre-trained subword embeddings based on Byte-Pair Encoding and trained on Wikipedia. This handles inflections and out of vocabulary words very well. The embedding of a word is considered as the sum of all the subword embeddings for our model. [35]

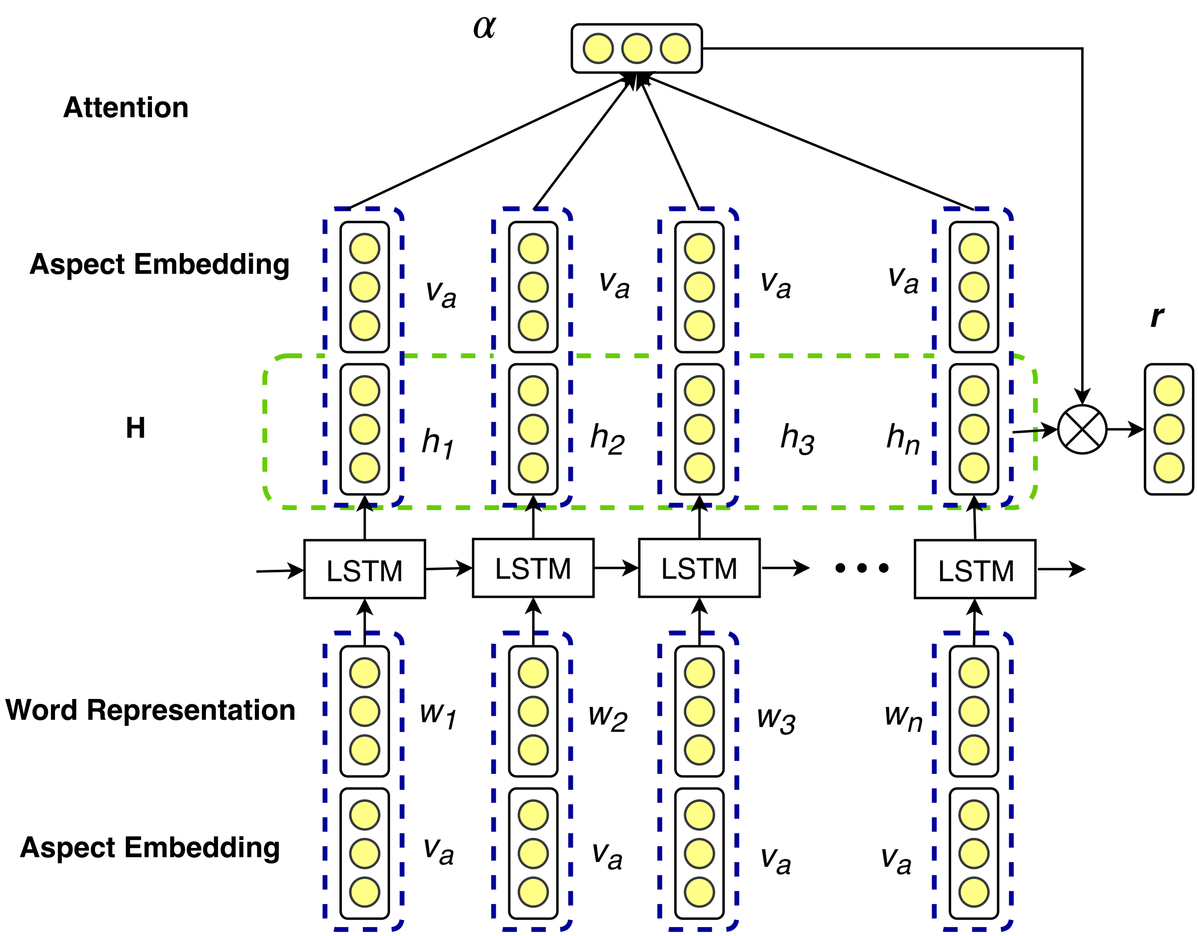

Figure 4.5 ATAE-LSTM model architecture

- **Pre-trained Telugu word embeddings:** There are pre-trained word embeddings available for Telugu which were obtained by running word2vec on large corpus [1]. These embeddings are used to initialize the embedding layer.
- **Fasttext Embeddings:** We initialize the embedding layer using fasttext word embeddings. [12]

4.2.1 Model configuration and training

4.2.1.1 Aspect Term Identification

We tokenized each sentence into a list of words. We only retain those words appearing more than 5 times in building vocabulary. The hyperparameters of the model are tuned on the validation set. In our experiments, we set the word embedding dimension to 300, character embedding dimension to 30,

[1] `https://bit.ly/2JQNYrw`

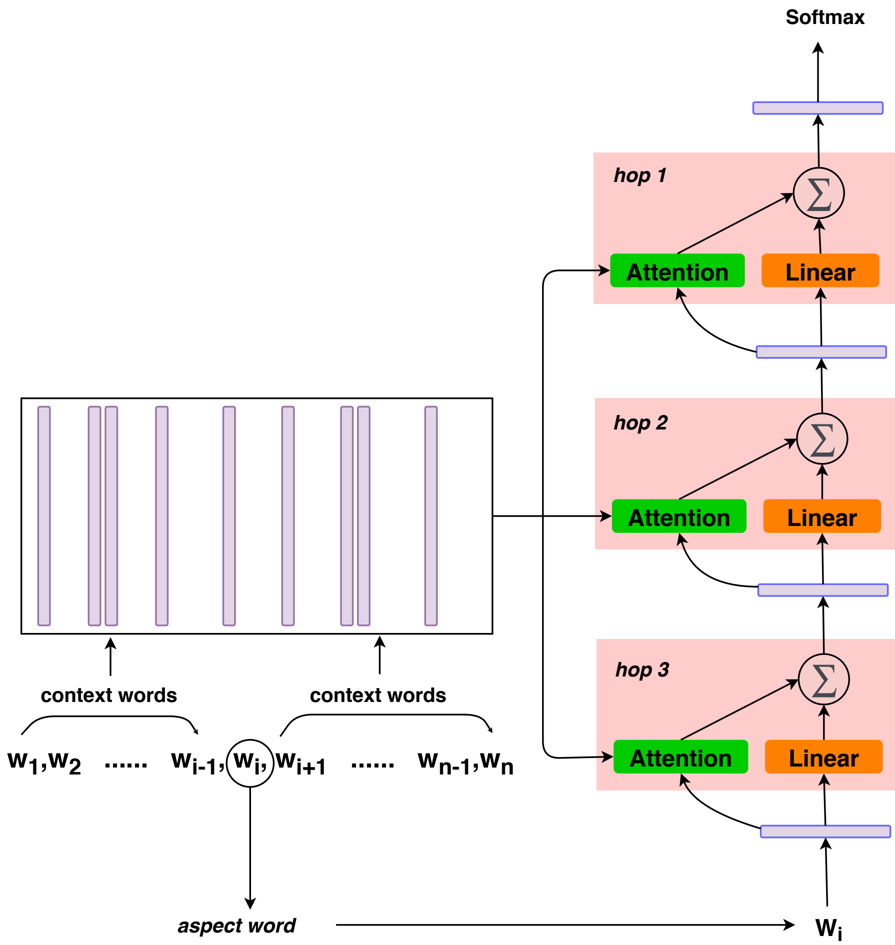

Figure 4.6 Deep Memory Network model architecture

word and character LSTM dimension to 300. The combination of forward and backward LSTM gives us 600 dimension for sentence annotation.

For training, we use a batch size of 10 and maximum words in a sentence as 50. We pad and truncate sentences to convert them to sequences of fixed length. We used stochastic gradient descent to train all models with momentum of 0.9.

4.2.1.2 Aspect Categorisation and Polarity Classification

The hyperparameters of the model are tuned on the validation set. In our experiments, we set the word embedding dimension to 300 and the LSTM dimension to 300. Max sequence length is set to 50 and sentences are padded/truncated accordingly. A dropout of 0.1 and l2 regularisation is used. For training, we use a batch size of 64.

4.2.2 Results and Analysis

Table 4.1 shows the results of aspect term identification. The results show that LM-LSTM-CRF + Fasttext embeddings outperforms all other models. LM-LSTM-CRF + BPEmb and LM-LSTM-CRF + Fasttext embeddings perform better because they handle out-of-vocabulary words using subword information which is very useful in case of languages like Telugu. The performance of LM-LSTM-CRF + random embeddings model is comparatively low because random embeddings do not capture contextual information like embeddings pre-trained on large corpus. As Telugu is a highly agglutinative language, it requires the model to have character level information. The reason for the low performance of LSTM+CRF and bi-LSTM+CRF models is that they do not incorporate character level information required to handle agglutinative words. We were able to attain an F1 score of 83.1% as a baseline for our dataset.

Table 4.2 and 4.3 shows results of aspect categorisation and aspect polarity classification. We used word embeddings fine-tuned on our dataset as they were performing better than other embeddings. Results show that TC-LSTM outperforms all other models. This is because it incorporates aspect connection with each word while forming sentence representation and this helps in cases of long-term dependencies, where words carrying polarity information are far away from the aspect term. As the vocabulary size was less, the performance of attention-based models was relatively low. Traditional SVM, Naive Bayes performance was comparable to other models though they are only based on bag-of-words features. We were able to obtain baseline accuracy of 79.71% and 79.68% for aspect categorization and aspect polarity identification respectively.

Methods	Precision	Recall	F1 score
LSTM + CRF	74.6%	69.2%	70.7%
bi-LSTM + CRF	79.3%	74.9%	75.8%
LM-LSTM-CRF + random Embeddings	81.3%	77.4%	77.7%
LM-LSTM-CRF + Pre-trained Telugu word2vec	82.3%	83.0%	81.5%
LM-LSTM-CRF + BPEmb	84.1%	82.6%	82.4%
LM-LSTM-CRF + Fasttext Embeddings	**84.4%**	**84.2%**	**83.1%**

Table 4.1 Results of aspect extraction

Methods	Precision	Recall	F1 score	Accuracy
SVM + TFIDF + Unigrams	70.8%	60.81%	65.42%	60.81%
SVM + TFIDF + Bigrams	69.81%	63.62%	66.57%	64.7%
Naive Bayes	62.16%	41.30%	46.40%	41.3%
LSTM	73.34%	67.91%	68.92%	74.79%
TD-LSTM	72.99%	**73.77%**	72.82%	76.33%
TC-LSTM	**74.54%**	72.58%	**73.36%**	**79.71%**
ATAE-LSTM	73.79%	68.36%	69.83%	75.91%
IAN	70.60%	68.96%	69.65%	74.93%
Deep Memory Networks	72.21%	66.19%	67.49%	74.37%

Table 4.2 Results of aspect categorisation

4.3 Summary

In this chapter, we described the experiments we have performed on the three tasks for which the dataset was created. We described the experimental setup and documented the results. The best performing model for aspect term identification is LM-LSTM-CRF feeded with Fasttext embeddings with an F1 score of 83.1%. For aspect polarity classification and aspect category detection, TC-LSTM performs the best with F1 scores 73.36% and 72.32% respectively.

Methods	**Precision**	**Recall**	**F1 score**	**Accuracy**
SVM + TFIDF + Unigrams	56.72%	51.31%	51.04%	51.36%
SVM + TFIDF + Bigrams	58.74%	52.33%	52.46%	52.33%
Naive Bayes	56.31%	51.9%	52.22%	51.9%
LSTM	69.9%	67.16%	68.18%	73.88%
TD-LSTM	70.34%	70.06%	66.92%	77.32%
TC-LSTM	**74.65%**	**72.04%**	**72.32%**	**79.68%**
ATAE-LSTM	68.05%	69.69%	68.61%	73.03%
IAN	67.66%	67.92%	67.78%	73.53%
Deep Memory Networks	71.50%	68.49%	69.58%	74.72%

Table 4.3 Results of aspect based sentiment analysis

Chapter 5

Application of pre-trained models

5.1 Introduction

To improve the performance of ABSA, the next steps would be to look at transformers and transfer learning methods that have shown promise in both classification tasks like text classification[85], sarcasm detection [23]and text generation tasks like text summarization[59] and neural machine translation [107]. All these works adopt the transfer learning approach which involves using a transformer-based model that is trained on large amounts of data. The pre-trained model is then fine-tuned on the labeled data of the specific downstream task. This leads to a model that performs significantly better than models trained from scratch.

The original idea of transfer learning originated in CV where a large model is trained to solve a generalized task, the large generalized model is then trained on a task with lesser data. ImageNet is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds of thousands of images. The project has been instrumental in advancing computer vision and deep learning research [78] With this inspiration, experiments were performed to know how transferable neural networks are for NLP applications as in [60]. The observations were that the transferability largely depends on how semantically similar the source and target tasks are. In transfer learning settings, the outermost layers are task-specific, whereas the inner layers contain transferable information.

One of the key breakthroughs in NLP came from leveraging knowledge of large corpora of unlabelled data to build continuous representations of words. The CBOW and the Skip-Gram models [58] have eliminated the limitation that the words be treated as atomic vectors, and have significantly improved the performance across several complex NLP tasks. Another key embedding method came from the GloVe model [69] where the distance between the embedding vectors of words that co-occur is minimized. Embedding methods as such have helped project language entities into semantically meaningful spaces and have demonstrated superior performances on all downstream NLP tasks.

Despite the success of these embedding methods, in practice, domain adaptation capabilities are observed to be limited warranting training from scratch for each domain. The Universal Language Fine

Tuning Model (ULMFit)[36] has presented a framework for effective transfer learning across domains for several NLP tasks. Another limitation of these embedding methods was their context-agnostic nature. This is especially problematic in languages such as English, where polysemous words are not uncommon. The ELMo [70] methods provide context-sensitive embeddings for words. Several key ideas from this method have laid the foundation for building large-scale language models in NLP.

Deep learning networks like RNN, LSTM, GRU have been developed to effectively capture dependencies between words of a sentence. But there are limitations to these models. They cannot capture the long-term dependencies effectively due to diminishing and exploding gradients. Moreover, training these models is computationally expensive because of the inherently sequential nature of these models. This makes it impossible to train models on large corpora. In 2018, [93] proposed the transformer network which to a large extent addressed the shortcomings of RNNs.

Transformers use self-attention that eschews recurrence which addresses several shortcomings in recurrent neural networks and their derivatives. Self-attention in tandem with positional encoding captures all these dependencies without the need for recurrence. It also boosts the training latency, as training can be parallelized due to the elimination of the sequential processing of tokens. This property of transformers makes them very capable of being able to be trained on large data for self-supervised modeling. Transformers, as a standalone model, has improved the performances in all standard NLP benchmarks. But the real potential of the transformers was unleashed due to their ability to train and generalize on extremely large sets of unlabelled data through self-supervised learning. [25] was one of the first such models, that made use of transformers to build a robust language model. Instead of training transformers on a downstream task directly, they were trained in a self-supervised manner using MLM(masked language modeling) and NSP(next sentence prediction) as their objectives. This knowledge of this pre-trained model is leveraged on specific NLP tasks by using fine-tuning (a form of transfer learning).

Along with BERT, various other models like GPT[75], XLNet[104], DistilBert[80], MASS[83], Roberta[53], BART[48] have emerged with transformer architecture at their core, but with differences in their training objectives or hyperparameters.

Roberta[53] is a BERT-based model with several improvisations. It proved that the choice of hyperparameters while training and the data size could greatly impact the pre-trained model. Roberta also proposed a dynamic masking approach where the masking is randomized while inputting each sequence to the model while training. It also eliminated the loss generated by the next sentence prediction training objective. All these factors contributed to the State-of-the-art performance of Roberta in several renowned NLP tasks.

However, most of these models have been exposed to only English data during the pre-training phase and hence don't extend much to Telugu, as the cross-lingual transfer is not very effective primarily because they are not typologically similar languages. Training models of that scale from scratch is not a feasible task as data of that scale is not available as in the case of high-resource languages like English etc. So in the current setting, the best way to solve this is to leverage multilingual pre-trained models which are exposed to some amount of Telugu and other typologically similar languages thereby enabling cross-lingual transfer. XLM-R [21] model is the best candidate for this, as it is the state-of-the-art model, trained on a multilingual model with optimized hyper-parameter setting.

This chapter describes about experiments done around transformer models on the ABSA Telugu movie review dataset. The primary motive of this experiment is to gauge the difference between a pre-trained model trained on high-resource language and fine-tuned on the dataset and a pre-trained model that is exposed to Telugu and other typologically similar languages to Telugu and fine-tuned in a similar setting.

5.2 Transformers

The Transformer architecture[93] was proposed to eliminate recurrence from the dependencies and in training, which is a key improvement over RNN and its derivatives. Instead of sequential dependencies, transformers use a self-attention mechanism to capture the relations. The dependencies of a word are captured from both directions. Self-attention also boosts the training speed, as it can be parallelized.

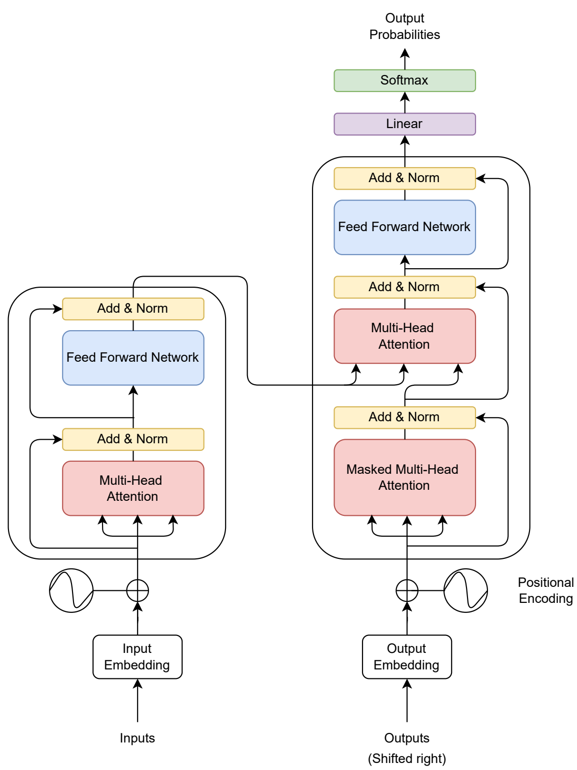

Figure 5.1 Transformer Encoder Decoder

Transformers consist of an encoding and a decoding component. The encoding component is a stack of six identical encoding units. Similarly, the decoding component is a stack of decoders of the same number.

Encoder: Each encoding unit consists of two layers, the self-attention layer and the other feed-forward network layer. A self-attention module takes in n inputs and returns n outputs. It helps the encoder look at other words in the input sentence and find out who they should pay more attention to as it encodes a specific word. The outputs are aggregates of these interactions and attention scores.

Self-Attention Layer: In the self-attention layer, firstly, the word embedding is transformed into three separate matrices queries, keys, and values by multiplying word embedding against three matrices with learned weights. The query and key vectors are turned into a single value via the dot product. The output from the dot product is then adjusted using normalization and softmax. We multiply each word embedding value vector by its softmax, this squeezes out low-value words, which will have a low softmax vector. Finally, the output of the self-attention layer is computed by the addition of results. The self-attention calculation involves only dot multiplication and some scalar operations on vectors, it can easily be written as a (fast, parallelizable) single-pass matrix operation. This makes both the forward and backward passes on this layer fast to calculate.

The outputs of the self-attention layer are fed to a feed-forward neural network. The exact same feed-forward network is independently applied to each position.

Decoder: The decoder has one additional layer, an encoder-decoder attention layer. An encoder-decoder attention layer, operates on the output of the decoder's self-attention layer and the output of the final encoder as input, helping the decoder focus on relevant parts of the input sentence. The decoder network's self-attention layer has all of the positions after the current word position masked by allowing to use of information that occurred prior to its current position in the sentence.

Encoder-Decoder Attention Layer: The encoder-decoder attention layer is a self-attention layer with different sources. It takes its query matrix from the previous layer, the self-attention layer. It uses the key and value matrices from the output of the encoder. The sharing of the key and value matrices from the encoder with the decoder is thus enabling the transfer of learning in the encoder-decoder architecture. It serves the same purpose as in seq2seq models.

5.3 BERT

BERT, Bidirectional representations from Transformers [25] proposes a novel innovation of training Transformer for language modeling. BERT utilizes the attention mechanism to learn the contextual relations between words or tokens in a text sequence.

In the previous section, it is described that the transformer has two separate components, the encoder that reads the input and converts it into a representation and the decoder that produces a prediction for the targeted task. BERT only requires the encoder mechanism, as its goal is language modeling. Choosing an effective prediction goal is a challenge while training a language model. The most frequent approach is to predict the next word given a sequence of words. But it limits contextual learning because of the directional aspect. To counter this, BERT uses two training strategies, namely Masked Language Modeling (MLM) and Next Sentence Prediction (NSP)

5.3.1 Masked LM (MLM)

In MLM, unlike predicting the next word given a sequence, in regular language modeling, 15 percent of tokens are masked before giving it as the input for the encoder layer, and the masked tokens are represented by the token [MASK]. The objective is to predict the masked tokens of the original sentence using the unmasked words as the context. To achieve this, the encoder output is fed to a classification layer. These output vectors are further multiplied with the embedding matrix to transform them into vocabulary dimensional vectors, which are further applied with a softmax layer to get the probability distribution.

The BERT loss calculated ignores the prediction of non-masked words and considers only the prediction of masked words which significantly delays the convergence of the model.

5.3.2 Next Sentence Prediction (NSP)

In NSP, given a pair of sentences, the model predicts whether the second sentence of the pair is the subsequent sentence of the first one in the original document. For training this, 50 percent of the input pairs of sentences are actual subsequent sentences. For the rest of the 50 percent input pairs, the second sentence of the pair is randomly selected in such a way that it is not the subsequent one of the first, under the assumption that the second sentence is disconnected from the first.

The input is processed as follows to help the model understand the differences between the two sentences:

1. The first sentence is preceded by a [CLS] token which indicates the beginning of the sentence pair and a [SEP] token is inserted in between the two sentences.

2. An additional embedding with a vocabulary of 2 is added to each token to identify whether the token is from sentence A or sentence B similar to positional encoding.

3. As presented in [93], each token is added with an embedding for positional encoding.

Post processing the input sentences, they are passed to the BERT model. The output of the [CLS] token from there, which technically represents the meaning and context of the entire sentence, is used to classify whether the sentence pair is pair of subsequent sentences or not. That output is passed to a classification layer to transform into a 2x1-shaped vector. The vector is now given to a softmax layer to determine the result.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies.

5.4 Roberta

RoBERTa (Robustly Optimized BERT Pre-training Approach) [53] has almost similar architecture as compared to BERT, with changes in the pre-training steps like removal of next sentence prediction from the training task, optimizing the hyperparameters and increasing the data size and also the sequence lengths. It reduced the time of pre-training and also produced better results. The main changes that are made in RoBERTa are mentioned below:

1. Removal of NSP: NSP as mentioned in the above subsection is a training objective in which for a pair of input sentences, the model needs to predict whether the second sentence is a subsequent sentence of the first one. It is removed for the model to improve upon the MLM objective, which led to significantly better performance.

2. Bigger batch sizes & longer sequences: BERT is trained for 1M steps with a batch size of 256 sequences. RoBERTa is trained in longer batches of 8K and in 31K steps. This change improves the perplexity of the MLM objective. It also makes the training faster as large batches can be parallelized with distributed data parallel training.

3. Dynamic masking pattern: In BERT, masking is done while pre-processing the input only once, which results in a static single mask. In RoBERTa, to avoid static masking, two kinds of masking were done. In the first approach, the training data was duplicated 10 times and each time a different masking strategy was used over the 40 epochs of training. Whereas in the second, approach the masking strategy was different each time a sequence was fed to the model. The first approach produced almost similar results to the original BERT model whereas the second approach is slightly better than the first.

RoBERTa model was trained on 160 GB of data containing English Corpus Wikipedia data, News articles, open web text, and stories data. RoBERTa model achieves better results in comparison to BERT.

BERT and RoBERTa both being big neural network architectures, with huge millions of a number of parameters. Training these models from scratch on a small dataset would result in over-fitting. So, it is better to use a pre-trained model that was trained on a huge dataset, as a starting point. Then we can train the model on our relatively smaller dataset.

5.5 Experimental Setup

The crux of the experiment is to evaluate pre-trained transformer models on the dataset created from Telugu movie reviews for ABSA.

For the sake of experimentation, simpletransformers [76] library has been used to experiment with different pre-trained transformer models, with the primary focus on the multilingual pre-trained model XLM-R[21]. The primary motive of this experiment is to gauge the difference between a pre-trained model trained on high-resource language and fine-tuned on the dataset and a pre-trained model that is exposed to Telugu and other typologically similar languages to Telugu and fine-tuned in a similar setting.

5.5.1 Pre-trained Models

BERT[25] and XLM-R[21] are the two pre-trained models used for the purpose of this experiment. Both models have been subject to similar fine-tuning setups as described later.

5.5.1.1 BERT

BERT model [25] i.e *bert-base-cased* was used, and this was pre-trained on data from BookCorpus containing 11,038 books and English Wikipedia pages.

5.5.1.2 XLM-R

XLM-Roberta model[21] i.e *xlm-roberta-base* was pre-trained on 2.5TB of data and has been exposed to about 100 languages of which Telugu and other typological equivalents of Telugu are also present. The base version used in the experiment has 250 Million parameters and a vocabulary size of 250k.

5.5.2 Finetuning

These models have been fine-tuned for three tasks i.e aspect detection, aspect Sentiment classification, and aspect Category Detection.

5.5.2.1 Aspect Detection

Each review sentence was tokenized, and the corresponding annotated B, I, O notations for the aspects were used as the training, and evaluation data in a 80:20 manner. The model was fine-tuned on this data for 5 epochs with a learning rate of 1e-5, with early stopping enabled at the patience of 2 epochs with cross-entropy loss.

5.5.2.2 Aspect Sentiment Classification

From each review sentence, aspect units are generated. Each aspect unit consists of the sentence where the aspect was present, the aspect term, and the annotated sentiment label. Positive, Negative and Neutral labels are present in the data. This data was divided into test, eval in 80:20 manner. The pre-trained model was finetuned on this data for 7 epochs, with a learning rate of 1e-5, with early stopping enabled at the patience of 2 epochs with cross-entropy loss.

5.5.2.3 Aspect Category Detection

This task shares a commonality with the Aspect Sentiment Classification task, barring the fact that the target classes are different. Different labels such as story, acting, direction, technical, music, and general are present in the data. This task shares a similar fine-tuning setup as Aspect Sentiment Classification barring the change that 10 epochs have been used for fine-tuning.

In these settings, various pre-trained transformer models have been fine-tuned, and the results in these settings have been documented in the below section.

5.6 Results and Discussion

It can be seen from 5.1, 5.2, 5.3 that XLM-R outperforms BERT in all three tasks, as it has been exposed to Telugu and other typologically similar languages in the pre-training stage which is not the case with BERT. However, it has to be noted that BERT model also performs comparably close to XLM-R and beats the LSTM benchmark in Aspect Sentiment Classification and Aspect Category Detection, emphasizing the significance of fine-tuning on the task-specific dataset created. Although pre-training doesn't expose the model to the language of interest i.e Telugu in this case, fine-tuning on the Telugu ABSA dataset led to significant improvements in comparison to all the prior models.

	Precision	**Recall**	**F1 Score**
BERT	77.8	80.7	79.2
XLM-R	86.5	83.8	85.2

Table 5.1 Aspect Term Extraction Results

	Precision	**Recall**	**F1 Score**	**Accuracy**
BERT	83	83	83	83
XLM-R	86	86	86	86

Table 5.2 Aspect Sentiment Classification Results

	Precision	**Recall**	**F1 Score**	**Accuracy**
BERT	94	93	93	93
XLM-R	94	94	94	94

Table 5.3 Aspect Category Detection Results

It was also seen that XLM-R model performed better specifically in the cases of multi-word aspect terms, and longer sentences in comparison to the LSTM benchmark models. The multi-word part can be attributed to language-specific knowledge from the pre-training stage as multi-word aspects are complex and might involve the structure of Telugu in some cases.

It can also be seen that the models perform the best on the Aspect Category Detection task and least on the Aspect Term Extraction task. This is because the categories have to be referred around the aspect terms, however, the identification of aspect as such is a far more challenging task.

In essence, it can be seen that like the other tasks of NLP, even in this realm, transformer-based models beat the early DNN models. However, by achieving comparable results with monolingual and multilingual models exposed to Telugu, the importance of the dataset has been validated.

5.7 Conclusion

In this chapter, we performed experiments on our movie review dataset for all three tasks. We used BERT which was trained on English data and XLM-R which is a multilingual model that was exposed to many languages including Telugu and its typologically similar languages. We evaluated the results in both settings and observed that XLM-R which is exposed to Telugu outperforms BERT in all three tasks of the dataset. We also observed that XLM-R outperforms the baselines that were set for the dataset.

Chapter 6

Conclusions and Future Work

In the wake of recent developments, the availability of high quality annotated task-specific data is essential for improving the quality of NLP applications in low resource languages. There is a lot of data that is prevelant, and users have started adopting Indic languages in native scripts as well as transliterated scripts across various social media platforms. In order to make the best use of this data and also to explore further avenues for interaction in multiple languages, creating annotated data becomes a stepping stone. In the context of Telugu, we created a comprehensive dataset for ABSA for Telugu movie reviews.

At the heart of the project is an annotation framework, which was developed for the purpose of making annotation easy, and can be easily extended/used for other similar use-cases. To start with, raw data was collected from various movie review websites. The data was pre-processed to eliminate noise introduced in web ecosystem like urls, html code snippets etc. Also, the data was filtered to eliminate non-Telugu words and spelling corrections for Telugu words was done. Using the framework mentioned above, the sanitized date was annotated in a reliable manner to obtain datasets for aspect term identification, aspect polarity classification and aspect term categorisation. A set of comprehensive guidelines were designed to direct the annotators and the data was cross annotated to ensure that the quality is high. Our dataset is freely available for download[1] to encourage further exploration in this domain and NLP in Telugu. This can also benefit other Dravidian languages in terms of cross-lingual transfer when there is shortage of data.

As with the NLP field at large, the central piece of modeling relied on Deep Neural Networks (DNNs). Various DNN based solutions were considered for the three tasks for which annotation was done i.e. aspect term identification, aspect polarity classification and aspect term categorisation. Extensive experimentation was done in order to firstly arrive at baselines for the newly created datasets. These baseline models include Classical ML models like Naive Bayes, SVM, and also some trivial DNN variants. This was done in order to gauge the experiments and have a comparison criterion for the candidate models.

[1] `https://bit.ly/3Wus4w8`

6.1 Future Work

The current work presents a high quality annotated dataset which can be worked upon to improve the modeling accuracy. This work can be extended further by exploring it through the alternative lens of cross lingual transfer, which is very relevant in low resource settings. The natural extension to this work is exploring end to end ABSA, where instead of the three model setting, given an input sentence, a single model can be used to do all the tasks i.e aspect term identification, aspect polarity classification and aspect term categorisation as mentioned in [95]

Related Publications

1. Yashwanth Reddy Regatte, Rama Rohit Reddy Gangula, and Radhika Mamidi. 2020. **Dataset Creation and Evaluation of Aspect Based Sentiment Analysis in Telugu, a Low Resource Language.** *In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 5017–5024, Marseille, France. European Language Resources Association.*

2. Yashwanth Reddy Regatte and Radhika Mamidi. 2023. **Fine-tuning Pre-trained models for ABSA in Telugu.** (to be submitted)

Bibliography

[1] H. Abburi, E. Sai, S. Gangashetty, and R. Mamidi. Multimodal sentiment analysis of telugu songs. 12 2017.

[2] M. S. Akhtar, A. Ekbal, and P. Bhattacharyya. Aspect based sentiment analysis in hindi: resource creation and evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2703–2709, 2016.

[3] M. S. Akhtar, A. Ekbal, and P. Bhattacharyya. Aspect based sentiment analysis: Category detection and sentiment classification for hindi. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 246–257, Cham, 2018. Springer International Publishing.

[4] M. S. Akhtar, A. Kumar, A. Ekbal, C. Biemann, and P. Bhattacharyya. Language-agnostic model for aspect-based sentiment analysis. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 154–164, Gothenburg, Sweden, May 2019. Association for Computational Linguistics.

[5] M. S. Akhtar, P. Sawant, S. Sen, A. Ekbal, and P. Bhattacharyya. Improving word embedding coverage in less-resourced languages through multi-linguality and cross-linguality: A case study with aspect-based sentiment analysis. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(2), dec 2018.

[6] R. Alzaidy, C. Caragea, and C. L. Giles. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*, WWW '19, pages 2551–2557, New York, NY, USA, 2019. ACM.

[7] S. Badugu. Morphology based pos tagging on telugu. 2014.

[8] V. Bajaj, K. Pant, I. Upadhyay, S. Nair, and R. Mamidi. TEASER: Towards efficient aspect-based SEntiment analysis and recognition. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 102–110, Held Online, Sept. 2021. INCOMA Ltd.

[9] A. Bakliwal, P. Arora, and V. Varma. Hindi subjective lexicon: A lexical resource for Hindi adjective polarity classification. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1189–1196, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[10] A. Bhattacharya, A. Debnath, and M. Shrivastava. Enhancing aspect extraction for Hindi. In *Proceedings of the 4th Workshop on e-Commerce and NLP*, pages 140–149, Online, Aug. 2021. Association for Computational Linguistics.

[11] J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[13] M. Bommadi, S. Terupally, and R. Mamidi. Question and answer pair generation for Telugu short stories. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 355–361, Indian Institute of Technology Patna, Patna, India, Dec. 2020. NLP Association of India (NLPAI).

[14] M. Bommadi, S. Terupally, and R. Mamidi. Automatic learning assistant in Telugu. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, pages 29–37, Online, Aug. 2021. Association for Computational Linguistics.

[15] C. Brun, D. N. Popa, and C. Roux. XRCE: Hybrid classification for aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 838–842, Dublin, Ireland, Aug. 2014. Association for Computational Linguistics.

[16] C. Chen, Z. Teng, Z. Wang, and Y. Zhang. Discrete opinion tree induction for aspect-based sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2064, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[17] P. Chen, Z. Sun, L. Bing, and W. Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[18] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, and H. Wang. Aspect-level sentiment classification with heat (hierarchical attention) network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 97–106, New York, NY, USA, 2017. ACM.

[19] N. Choudhary, R. Singh, V. A. Rao, and M. Shrivastava. Twitter corpus of resource-scarce languages for sentiment analysis and multilingual emoji prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1570–1577, 2018.

[20] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[21] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2019.

[22] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[23] T. Dadu and K. Pant. Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 51–55, Online, July 2020. Association for Computational Linguistics.

[24] P. Danda, P. Jwalapuram, and M. Shrivastava. End to end dialog system for telugu. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 265–272, 2017.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[26] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[27] S. Dowlagar and R. Mamidi. Cmsaone@dravidian-codemix-fire2020: A meta embedding and transformer model for code-mixed sentiment analysis on social media text, 2021.

[28] S. Dowlagar and R. Mamidi. Graph convolutional networks with multi-headed attention for code-mixed sentiment analysis. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 65–72, Kyiv, Apr. 2021. Association for Computational Linguistics.

[29] S. R. Duggenpudi, K. S. S. Varma, and R. Mamidi. Samvaadhana: A telugu dialogue system in hospital domain. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 234–242, 2019.

[30] F. Fan, Y. Feng, and D. Zhao. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3433–3442, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[31] X. Fu, W. Liu, Y. Xu, and L. Cui. Combine hownet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis. *Neurocomputing*, 241:18–27, 2017.

[32] R. R. R. Gangula and R. Mamidi. Resource creation towards automated sentiment analysis in Telugu (a low resource language) and integrating multiple domain sources to enhance sentiment prediction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[33] K. Garg and P. K. Buttar. Aspect based sentiment analysis of hindi text review. *International Journal of Advanced Research in Computer Science*, 8:831–836, 2017.

[34] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[35] B. Heinzerling and M. Strube. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA).

[36] J. Howard and S. Ruder. Universal language model fine-tuning for text classification, 2018.

[37] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[38] B. Huang and K. Carley. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[39] T. Ito, K. Tsubouchi, H. Sakaji, T. Yamashita, and K. Izumi. Word-level contextual sentiment analysis with interpretability. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4231–4238, Apr. 2020.

[40] G. Janardana Naidu and M. Seshashayee. Sentiment analysis for telugu text using cuckoo search algorithm. In *Smart Computing Techniques and Applications*, pages 253–257. Springer, 2021.

[41] S. Kiritchenko and S. M. Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. *CoRR*, abs/1805.04508, 2018.

[42] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, Dublin, Ireland, Aug. 2014. Association for Computational Linguistics.

[43] A. Kodirekka and A. Srinagesh. Sentiment extraction from english-telugu code mixed tweets using lexicon based and machine learning approaches. In *Machine Learning and Internet of Things for Societal Issues*, pages 97–107. Springer, 2022.

[44] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades. Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, 40(10):4065–4074, 2013.

[45] R. Kumar and R. Shriram. Sentiment analysis using bi-directional recurrent neural network for telugu movies. *Int J Innov Technol Explor Eng*, 9(2):241–245, 2019.

[46] S. S. V. Kusampudi, P. Sathineni, and R. Mamidi. Sentiment analysis in code-mixed telugu-english text with unsupervised data normalization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 753–760, 2021.

[47] X. Lei, X. Qian, and G. Zhao. Rating prediction based on social sentiment from textual reviews. *IEEE Transactions on Multimedia*, 18:1910–1921, 09 2016.

[48] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[49] 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)*, pages 946–956, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[50] X. Li, X. Sun, Z. Xu, and Y. Zhou. Explainable sentence-level sentiment analysis for amazon product reviews, 2021.

[51] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, 235:107643, 2022.

[52] L. Liu, J. Shang, F. F. Xu, X. Ren, H. Gui, J. Peng, and J. Han. Empower sequence labeling with task-aware neural language model. *CoRR*, abs/1709.04109, 2017.

[53] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[54] D. Ma, S. Li, X. Zhang, and H. Wang. Interactive attention networks for aspect-level sentiment classification. *CoRR*, abs/1709.00893, 2017.

[55] G. V. Mantena, S. Rajendran, S. V. Gangashetty, B. Yegnanarayana, and K. Prahallad. Development of a spoken dialogue system for accessing agricultural information in telugu. In *Proceedings of ICON-2011, 9th international conference on natural language processing*, 2011.

[56] M. Marreddy, S. R. Oota, L. S. Vakada, V. C. Chinni, and R. Mamidi. Am i a resource-poor language? data sets, embeddings, models and analysis for four different nlp tasks in telugu language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, apr 2022. Just Accepted.

[57] M. A. Masum, S. J. Ahmed, A. Tasnim, and M. S. Islam. Ban-absa: An aspect-based sentiment analysis dataset for bengali and it's baseline evaluation, 2020.

[58] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[59] D. Miller. Leveraging bert for extractive text summarization on lectures, 2019.

[60] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How transferable are neural networks in nlp applications?, 2016.

[61] S. Movahedi, E. Ghadery, H. Faili, and A. Shakery. Aspect category detection via topic-attention network, 2019.

[62] S. Mukku, N. Choudhary, and R. Mamidi. Enhanced sentiment classification of telugu text using ml techniques. 07 2016.

[63] S. S. Mukku and R. Mamidi. ACTSA: Annotated corpus for Telugu sentiment analysis. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 54–58, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[64] K. Nelakuditi, D. S. Jitta, and R. Mamidi. Part-of-speech tagging for code mixed english-telugu social media data. In *CICLing*, 2016.

[65] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 751–760, New York, NY, USA, 2010. Association for Computing Machinery.

[66] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics, July 2002.

[67] A. Pathak, S. Kumar, P. Roy, and B.-G. Kim. Aspect-based sentiment analysis in hindi language by ensembling pre-trained mbert models. *Electronics*, 10:2641, 10 2021.

[68] B. Patra, D. Das, A. Das, and R. Prasath. *Shared Task on Sentiment Analysis in Indian Languages (SAIL) Tweets - An Overview*. 12 2015.

[69] 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)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[70] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations, 2018.

[71] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June 2016. Association for Computational Linguistics.

[72] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, Aug. 2014. Association for Computational Linguistics.

[73] G. Qiu, B. Liu, J. Bu, and C. Chen. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.*, 37(1):9–27, Mar. 2011.

[74] B. R, A. Joshi, and P. Bhattacharyya. Cross-lingual sentiment analysis for indian languages using linked wordnets. pages 73–82, 12 2012.

[75] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. 2018.

[76] T. C. Rajapakse. Simple transformers. `https://github.com/ThilinaRajapakse/simpletransformers`, 2019.

[77] I. Rish. An empirical study of the naïve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell*, 3, 01 2001.

[78] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[79] K. Saikrishna and C. Subalalitha. Sentiment analysis on telugu–english code-mixed data. In *Intelligent Data Engineering and Analytics*, pages 151–163. Springer, 2022.

[80] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019.

[81] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[82] K. Schouten, O. van der Weijde, F. Frasincar, and R. Dekker. Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data. *IEEE Transactions on Cybernetics*, 48(4):1263–1275, 2018.

[83] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mass: Masked sequence to sequence pre-training for language generation, 2019.

[84] M. C. Sravanthi, K. Prathyusha, and R. Mamidi. A dialogue system for telugu, a resource-poor language. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 364–374. Springer, 2015.

[85] C. Sun, X. Qiu, Y. Xu, and X. Huang. How to fine-tune bert for text classification? In M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, editors, *Chinese Computational Linguistics*, pages 194–206, Cham, 2019. Springer International Publishing.

[86] K. N. Sunitha and N. Kalyani. A novel approach to improve rule based telugu morphological analyzer. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 1649–1652, 2009.

[87] D. Tang, B. Qin, X. Feng, and T. Liu. Effective lstms for target-dependent sentiment classification, 2015.

[88] D. Tang, B. Qin, and T. Liu. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[89] C. Toprak, N. Jakob, and I. Gurevych. Darmstadt service review corpus, 2010.

[90] O. Tsur, D. Davidov, and A. Rappoport. Icwsm — a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. *Proceedings of the International AAAI Conference on Web and Social Media*, 4(1):162–169, May 2010.

[91] A. Tulasi, K. Gupta, O. Gurjar, S. S. Buggana, P. Mehan, A. B. Buduru, and P. Kumaraguru. Catching up with trends: The changing landscape of political discussions on twitter in 2014 and 2019, 2019.

[92] S. Tulkens and A. van Cranenburgh. Embarrassingly simple unsupervised aspect extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3182–3187, Online, July 2020. Association for Computational Linguistics.

[93] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[94] Q. Wang, Z. Wen, Q. Zhao, M. Yang, and R. Xu. Progressive self-training with discriminator for aspect term extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 257–268, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[95] X. Wang, G. Xu, Z. Zhang, L. Jin, and X. Sun. End-to-end aspect-based sentiment analysis with hierarchical multi-task learning. *Neurocomputing*, 455:178–188, 2021.

[96] Y. Wang, M. Huang, X. Zhu, and L. Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[97] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada, Oct. 2005. Association for Computational Linguistics.

[98] H. Wu, Z. Zhang, S. Shi, Q. Wu, and H. Song. Phrase dependency relational graph attention network for aspect-based sentiment analysis. *Knowledge-Based Systems*, 236:107736, 2022.

[99] H. Xu, B. Liu, L. Shu, and P. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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)*, pages 2324–2335, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[100] H. Xu, B. Liu, L. Shu, and P. S. Yu. Double embeddings and CNN-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[101] H. Xu, B. Liu, L. Shu, and P. S. Yu. Bert post-training for review reading comprehension and aspect-based sentiment analysis, 2019.

[102] W. Xue and T. Li. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[103] B. Yang and C. Cardie. Extracting opinion expressions with semi-Markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and*

Computational Natural Language Learning, pages 1335–1345, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[104] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[105] Y. Yin, F. Wei, L. Dong, K. Xu, M. Zhang, and M. Zhou. Unsupervised word and dependency path embeddings for aspect term extraction, 2016.

[106] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '14, page 83–92, New York, NY, USA, 2014. Association for Computing Machinery.

[107] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T.-Y. Liu. Incorporating bert into neural machine translation, 2020.