Dataset Creation and Evaluation of Aspect Based Sentiment Analysis in Telugu, a Low Resource Language

by

Regatte Yashwanth Reddy, Rama Rohit Reddy Gangula, Radhika Mamidi

in

Language Resources and Evaluation Conference

Report No: IIIT/TR/2020/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2020

Dataset Creation and Evaluation of Aspect Based Sentiment Analysis in Telugu, a Low Resource Language

Regatte Yashwanth Reddy, Gangula Rama Rohit Reddy, Radhika Mamidi
LTRC IIIT Hyderabad
Hyderabad Telangana India
yashwanthreddy.r@research.iiit.ac.in, ramarohitreddy.g@research.iiit.ac.in, radhika.mamidi@iiit.ac.in

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 paper, 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.

Keywords: Aspect based Sentiment Analysis, Indian Languages, Deep Learning, Machine Learning

1. Introduction

The emergence and rapid growth of the internet made the availability of information easy. The availability of this enormous amount of information posed several challenges which led to the development of new research areas. Sentiment Analysis is one of them. 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 effective judgment of a product/service by assigning polarities to the reviews given by its users.

Sentiment analysis can be performed 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 terms in a sentence.

Many approaches are 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. Despite Telugu being the fourth most spoken language in India and sixteenth across the world, the area of aspect based sentiment analysis in Telugu remains unexplored. There are several challenges with Telugu as 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 that it is a low resource language. There have been some resources created in Telugu for Sentiment Analysis (SA) but they address SA only at coarser levels like Document level and Sentence level. In this paper, we create 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, (i) Aspect Term Identification, (ii) Aspect Polarity Classification and (iii) 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.

The further parts of the paper are divided into sections as follows: In section 2, we discussed the related work that is in line with Sentiment Analysis, ABSA and the work done in Indian languages. In section 3, we described the process of data creation and challenges involved in it. In section 4, we discussed about the methods we used to tackle ABSA and in section 5, we gave the details of experiments performed and analysed the results obtained and we finally concluded in section 6.

2. Related Work

Research in sentiment analysis has evolved a lot over the years. Many developments have been made since customer reviews were analysed in (Hu and Liu, 2004). ABSA is one of its branches which 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 (Qiu et al., 2011), (Tang et al., 2016), (Cheng et al., 2017), (Xue and Li, 2018). Most of these works use deep learning methods. But resource creation tasks like (Pontiki et al., 2014) and (Pontiki et al., 2016) are key for the progress of ABSA.

Lack of reliable resources of this kind restricts Indian Languages from research developments in ABSA. There have been some developments recently in Hindi. In (Akhtar et al., 2016), a dataset for Hindi was created and in (Akhtar et al., 2018), an approach was developed for ABSA in Hindi using the created dataset. However, in Telugu, there is no dataset available for ABSA. In Telugu, (Abburi et al., 2017) has a corpus of song lyrics for sentiment analysis. (Gangula and Mamidi, 2018) has several product, book and movie reviews annotated at document level. (Mukku et al., 2016) and (Mukku and Mamidi, 2017) have annotated data for sentiment analysis at sentence level. But sentiment analysis at aspect level remains unexplored in Telugu. In this paper, we created a dataset for aspect based sentiment analysis in Telugu.

3. 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.

3.1. 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 3.3.

3.2. 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. We developed a tool to facilitate the task of annotation. The tool is shown in the following Figure 1.

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. Consider the following example sentence, "allu arjun tana naTana to, danculatO AkaTTukunnADu." (*Allu Arjun has impressed with his acting and dance.*). "naTana" (*acting*) and "danculatO" (*with dances*) may seem like aspect terms. But, it can be understood from the sentence that, the verb "AkaTTukunnADu" (*impressed*) describes the noun "allu arjun" more than the other two. Hence, "allu arjun" is the only aspect term in the sentence.

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 Figure3. We use Cohen's Kappa Coefficient(Cohen, 1960) to determine the inter-annotator agreement between the annotators. The kappa coefficient is calculated as follows:

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

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 2. The kappa coefficient for this task was measured to be 96.72%.

3.3. 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"*,

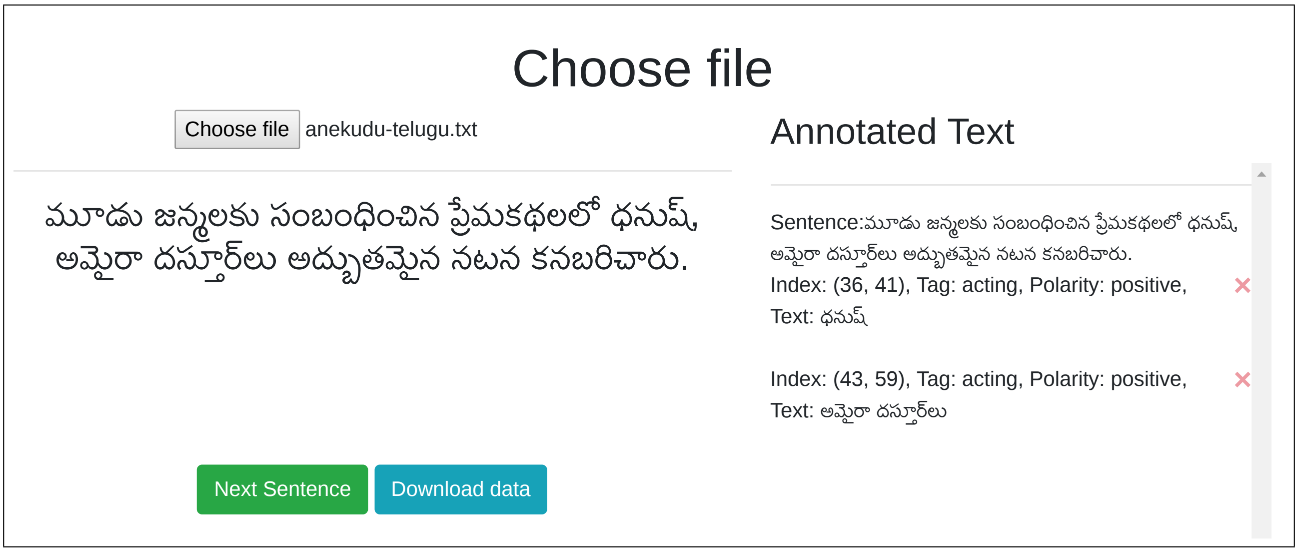

Figure 1: annotation tool

	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 2: Examples for ABSA

"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 4 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.

O	O	O	B	I	O	O	O
ఈ	సినిమా	లో	అల్లు	అర్జున్	బాగా	నటించాడు	.
ee	cinema	lo	allu	arjun	bAgA	naTinchADu	.
In this movie, Allu Arjun acted well.							

Figure 3: Example for Aspect Term Identification


```
[
          .........................................
          .........................................
 {
   "sentence": "రత్నవేలు కెమెరా వర్క్ ఆకట్టుకోగా, థమన్ బ్యాక్గ్రౌండ్ స్కోర్ పర్వాలేదనిపించింది.",
   "aspectTerms": [
    {
     "aspectTerm": "కెమెరా వర్క్",
     "start": 10,
     "end": 22,
     "category": "technical",
     "polarity": "positive"
    },
    {
     "aspectTerm": "బ్యాక్గ్రౌండ్ స్కోర్",
     "start": 40,
     "end": 60,
     "category": "music",
     "polarity": "negative"069
    }
   ]
  },
          .........................................
          .........................................
]
```


Figure 4: Dataset annotation structure

#Sentences		5027
#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 1: Dataset Statistics. #w denotes the count of w.

4. 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. 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(Alzaidy et al., 2019). The best performance was obtained when we used the Language Model - Long Short-Term Memory - Conditional Random Field (LM-LSTM-CRF) model(Liu et al., 2017). This model has been successful in solving sequence labeling tasks like POS tagging, Named Entity Recognition, etc.
Figure 5 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 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.

4.2. 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.

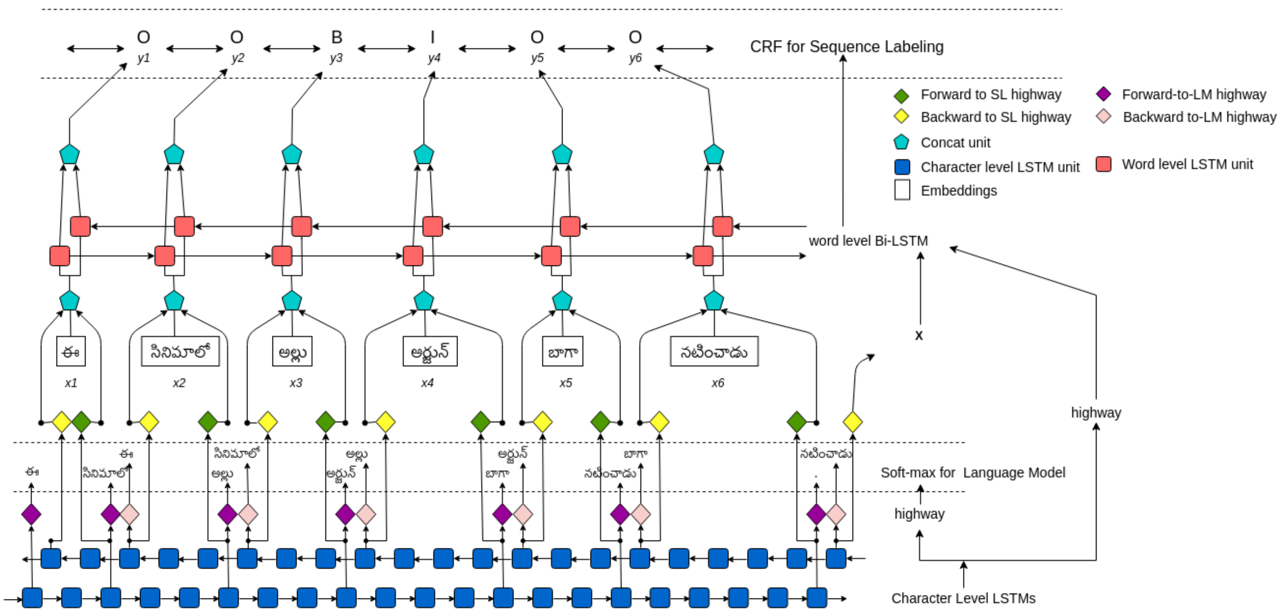

Figure 5: LM-LSTM-CRF architecture

4.2.1. 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(Cortes and Vapnik, 1995). 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.

4.2.2. 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 (Rish, 2001) to classify the aspect term.

4.2.3. LSTM

LSTM model architecture as in (Tang et al., 2015) 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.

4.2.4. Target-Dependent LSTM (TD-LSTM)

Target-Dependent LSTM model (Tang et al., 2015) 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 6 shows the architecture of the model.

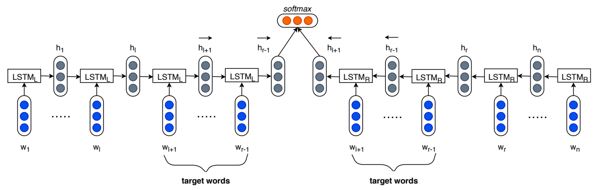

Figure 6: TD-LSTM model architecture

4.2.5. Target-Connection LSTM (TC-LSTM)

Target-Connection LSTM (Tang et al., 2015) 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 7 shows the architecture of the TC-LSTM model.

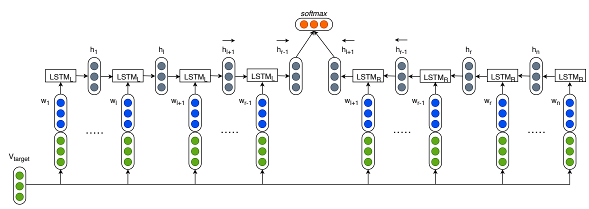

Figure 7: TC-LSTM model architecture

4.2.6. 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 8 shows the architecture of the ATAE-LSTM model.

(Wang et al., 2016)

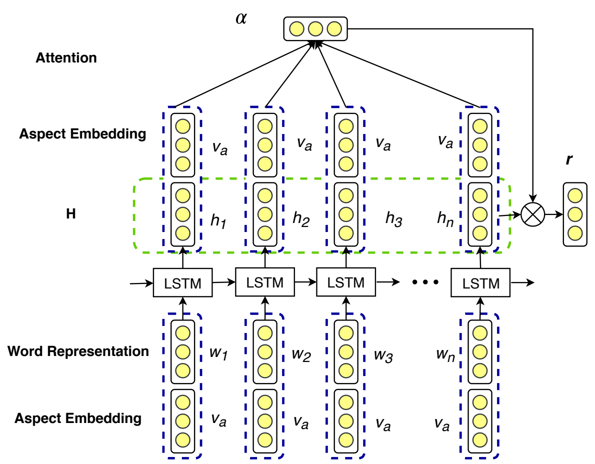

Figure 8: ATAE-LSTM model architecture

4.2.7. Interactive Attention Networks (IAN)

IAN (Ma et al., 2017) 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.

4.2.8. Deep Memory Networks

(Tang et al., 2016) 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 9 shows the architecture of the Deep Memory Network model.

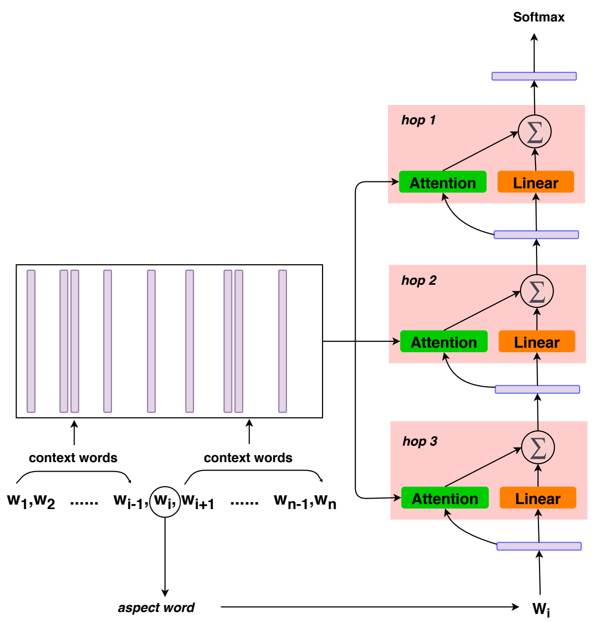

Figure 9: Deep Memory Network model architecture

5. Experiments and Evaluation

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. (Heinzerling and Strube, 2018)
- **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. (Bojanowski et al., 2017)

5.1. Model configuration and training

5.1.1. 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, 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.

5.1.2. 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.

5.2. Results and Analysis

Table 2 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

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

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 3 and 4 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 2: 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 3: Results of aspect categorisation

6. Conclusion

In this paper, we created a comprehensive dataset for Aspect Based Sentiment Analysis for Telugu movie reviews.

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: Results of aspect based sentiment analysis

We collected data from various movie review websites, preprocessed it and created an annotated corpus. The corpus consists of annotated data for aspect term identification, aspect polarity classification, and categorisation. We employed several deep learning methods for the three tasks. We established baselines on this dataset for all the three tasks thereby creating and evaluating a reliable dataset for a low resource language like Telugu. Our dataset is freely available for download[2] to encourage further exploration in this domain.

7. References

Abburi, H., Sai, E., Gangashetty, S., and Mamidi, R. (2017). Multimodal sentiment analysis of telugu songs. 12.

Akhtar, M. S., Ekbal, A., and Bhattacharyya, P. (2016). 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.

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

Alzaidy, R., Caragea, C., and Giles, C. L. (2019). 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. ACM.

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

Cheng, J., Zhao, S., Zhang, J., King, I., Zhang, X., and Wang, H. (2017). 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. ACM.

[2] http://tiny.cc/vdxugz

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

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

Gangula, R. R. R. and Mamidi, R. (2018). 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. European Language Resources Association (ELRA).

Heinzerling, B. and Strube, M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Hu, M. and Liu, B. (2004). 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. ACM.

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

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

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

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

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). 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, August. Association for Computational Linguistics.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., and Eryiğit, G. (2016). 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. Association for Computational Linguistics.

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

Rish, I. (2001). An empirical study of the naÃ¯ve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell*, 3, 01.

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

Tang, D., Qin, B., and Liu, T. (2016). 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, November. Association for Computational Linguistics.

Wang, Y., Huang, M., Zhu, X., and Zhao, L. (2016). 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, November. Association for Computational Linguistics.

Xue, W. and Li, T. (2018). 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. Association for Computational Linguistics.