A Comprehensive Study of Mahabharat using Semantic and Sentiment Analysis

Ranjani Parthasarathi, Professor and
Aishwarya Kumaran and Nithyasri and Shanmuga Priya and Srijeyarankesh J S
College of Engineering Guindy
Anna University, Chennai- 600 025

Abstract

Indian epics have not been analyzed computationally to the extent that Greek epics have. In this paper, we show how interesting insights can be derived from the ancient epic Mahabharata by applying a variety of analytical techniques based on a combination of natural language processing methods like semantic analysis, sentiment analysis and Named Entity Recognition (NER). The key findings include the analysis of events and their importance in shaping the story, character's life and their actions leading to consequences and change of emotions across the eighteen parvas of the story.

1 Introduction

Semantic analysis is the study of the meaning of language, whereas sentiment analysis is the study of emotions that has been depicted. Sentimental analysis is prevailing in various domains such as social media monitoring, customer support management, and analysing customer feedback.

Mahabharat is a tightly interwoven story with intricate characters traversing various incidents resulting in many course of actions. This makes Mahabharat an interesting study for analysing such characters and incidents using the various NLP techniques. The Mahabharat Ganguli translation is used for conducting such an analysis. Entity Analysis involves named entity recognition which helped discover many unfamiliar characters present in Mahabharat. Semantic Analysis is used to analyze sentence structure to highlight the events and their resulting actions whereas sentiment analysis is used to analyze the flow of emotions as the story progresses. Character analysis describes the character's life, the trails and tribulations the character has been through and his/her characteristics. The paper presents a unified technique to achieve the above as stated.

2 Related Works

Mahabharatha is an epic with valuable lessons on life and values .Epics like Mahabharata are a kind of tragedy and are built around noble men within the form of narratives. A tragedy typically has a plot with a beginning, a middle and an end and other constituents of the text are secondary to the plot. The start of the plot typically is a scenario of stability which gets disturbed by some events. Plots of tragedy have various constituents i.e. suffering, reversal, recognition of latest knowledge, surprise. An epic is different from a newer literary genre like a novel and will have lot of negative sentiment across its breadth but in spite of that conveys a noble theme in the minds of its audience.

Debabrati et al. (Das et al., 2016) has proposed the usage of NLP techniques such as sentiment analysis and characterization of important characters with respect to their emotion. Mabhabharatha text is tokenized using standard NLP techniques. - The tokens are POS (parts of speech) tagged and tagged tokens are mapped to synsets in Wordnet in a word sense disambiguation process. - The sentiment scores are picked up from SentiWordnet for each synset. - Overall sentiment of the parva is derived from these values by summing the constituent sentiment scores. Emotion analysis for the full text and each of the protagonists is done with the help of NRC word-emotion association lexicon. After extracting the relevant part of the corpus,the score is calculated for each POS (part of speech) tagged token for each emotion and finally summed up. However, by this approach one cannot get an overall view of the character in terms of their life, relations and actions but only about their emotions. The usage of lexicon based approach limits the ability of the model to learn new vocabulary. The proposed idea in this paper aims to remove these two limitations.

Named Entity Recognition is identifying proper-

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 308 - 317
December 15-18, 2022 ©2022 Association for Computational Linguistics

nouns in the text. The biggest challenge in Named Entity Recognition is the lack of sufficient labelled data. This poses a challenge for NER in Mahabharat as the standard tagged datasets are different in comparison. Active Learning is an efficient option as it helps identify samples that will be the most informative to the model(Li et al., 2022), discuss active learning technique for Named Entity Recognition. Further work was done by Yanyao Shen et.al (Shen et al., 2017) where a CNN-CNN-LSTM model was built for NER, in an iterative approach. They used the various selection strategies for NER such as least confidence, Maximum Normalized Log-Probabilities.

Named Entity Recognition is a sequence labelling task.(Akhundov et al., 2018) discusses the merits of using Bidirectional Long Short Term (BiLSTM) models for sequence labelling tasks. For any sequence labelling task the model is required to take into consideration the context of the entire sentence.

(Devlin et al., 2019) introduced a model called BERT. BERT was trained on two tasks - masked word prediction and next sentence prediction. These tasks can make use of data that requires no labelling and is widely available.

Conditional random field is a popular probabilistic method for structured prediction.(Sutton and McCallum, 2010) discussed the problem of classification by predicting a single discrete class variable y given a vector of features.

In co-referencing resolution, training recurrent neural networks to model long term dependencies is an issue faced.(Dhingra et al., 2017) had proposed to use external linguistic knowledge as an explicit signal to inform the model which memories it should utilize.

3 Methods

This section describes the design and implementation of the system being proposed with the help of overall system architecture represented in Figure 1. In this section the proposed methodology is discussed. Using Natural Language Processing techniques such as co-referencing, relationship extraction, analysis on events and many other functions are performed like automated question-answering, graphical representations and identifying relationships of different entities in the Mahabharat dataset. Relationship extraction is a key task done with the help of co-referencing. Event analysis with the help of BART for summarization and BERT for question answering. The character sketch is drawn from using adjective extraction model using BERT and POS tags. The POS tags along with generated summary of each parva in Mahabharatha is used to draw the character sketch. The emotion sketch is derived from using BERT model by using emotions from Go-Emotion dataset. The generated summary along with emotions extracted in every parva is passed through a text generation model for generating an emotion sketch.

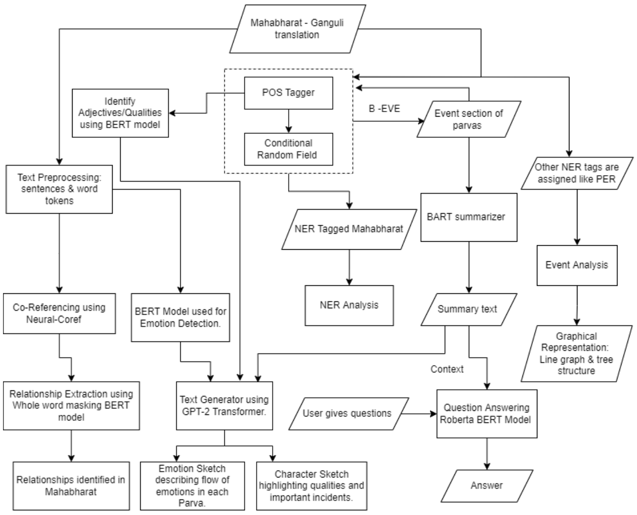

Mahabharat - Ganguli translation
Identify Adjectives/Qualities using BERT model
POS Tagger
Conditional Random Field
B -EVE
Event section of parvas
Other NER tags are assigned like PER
Text Preprocessing: sentences & word tokens
NER Tagged Mahabharat
BART summarizer
Event Analysis
NER Analysis
Co-Referencing using Neural-Coref
BERT Model used for Emotion Detection.
Summary text
Graphical Representation: Line graph & tree structure
Context
Relationship Extraction using Whole word masking BERT model
Text Generator using GPT-2 Transformer.
User gives questions
Question Answering Roberta BERT Model
Relationships identified in Mahabharat
Emotion Sketch describing flow of emotions in each Parva.
Character Sketch highlighting qualities and important incidents.
Answer

Figure 1: Overall Architecture Diagram

3.1 Raw Dataset

Kisari Mohan Ganguli's translation of the Sanskrit epic Mahabharat is the raw data acquired. The raw data consists of eighteen books. They are Adi Parva, Sabha Parva, Vana Parva, Virata Parva, Udyoga Parva, Bhishma Parva, Karna Parva, Shalya Parva, Sauptika Parva, Stri Parva, Santi Parva, Anusasana Parva, Aswamedha Parva, Asramavaisika Parva, Mausala Parva, Mahaprasthanika Parva and Svargarohanika Parva. The entire dataset has 1,35,850 sentences.

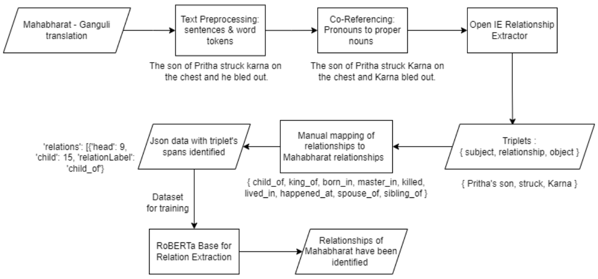

Mahabharat - Ganguli translation
Text Preprocessing: sentences & word tokens
Co-Referencing: Pronouns to proper nouns
Open IE Relationship Extractor
The son of Pritha struck karna on the chest and he bled out.
The son of Pritha struck Karna on the chest and Karna bled out.
'relations': [{'head': 9, 'child': 15, 'relationLabel': 'child_of'}
Json data with triplet's spans identified
Manual mapping of relationships to Mahabharat relationships
Triplets : { subject, relationship, object }
{ child_of, king_of, born_in, master_in, killed, lived_in, happened_at, spouse_of, sibling_of }
{ Pritha's son, struck, Karna }
Dataset for training
RoBERTa Base for Relation Extraction
Relationships of Mahabharat have been identified

Figure 2: Relationship Extraction Architecture Diagram

TAG	EXPANSION
B-PER	Beginning of Person entity
I-PER	Inside a Person entity
B-PLACE	Beginning of Place entity
I-PLACE	Inside a Place entity
B-EVE	Beginning of Event entity
I-EVE	Inside a Event entity
B-WEAPON & WAR STRATEGY	Beginning of Weapons and War Strategy entity
I-WEAPON & WAR STRATEGY	Inside a Weapons and War Strategy entity
B-COMMUNITY	Beginning of Communities entity
I-COMMUNITY	Inside a Communities entity
B-LIT and ART	Beginning of Literature and Art entity
I-LIT and ART	Inside a Literature and Art entity

Table 1: NER tags and their Expansions

Relation	Subject, Object
Child of	Person, Person
King of	Person, Place
Born in	Person, Place
Master of	Person Literature, Artifact
Killed	Person, Person/Place/Weapon
Lived in	Person, Place
Happened in	Event, Place
Spouse of	Person, Person
Sibling of	Person, Person
Friend of	Person, Person
Leader of	Person, Community/Place
Guardian of	Person, Person/Community
Belongs to	Person, Community/Place

Table 2: NER tags and their Expansions

3.2 Tool Used

The tools, libraries and environments used include (pandas development team, 2020), (Abadi et al., 2015), (Akbik et al., 2019), (Hunter, 2007), (Mausam et al., 2012), (Harris et al., 2020), (Pedregosa et al., 2011) and (Loper and Bird, 2002).

3.3 Named Entity Recognition

In this process, entities pertaining to Mahabharat have been identified as listed in the Table 1. To automatically identify these entities from the text, we trained a CRF model on the Mahabharat dataset. The Conditional Random Field model considers the semantics of the given text where, given a sequence of input words we obtain the sequence of output labels. Training set $\left\{\left(\mathbf{X}^{(t)}, \mathbf{y}^{(t)}\right)\right\}$ is a set of input and target sequences pairs:

input words are $\mathbf{X}^{(t)} = \left[\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_{K_t}^{(t)}\right]$
target labels are $\mathbf{y}^{(t)} = \left[y_1^{(t)}, \ldots, y_{K_t}^{(t)}\right]$

K_t is the length of the t^{th} sequence.
A set of features from the Mahabharat dataset has been crafted which is provided to the CRF model. The features of the sentence given to the model include the case of the word, the last few letters of the word. The implementation of the Conditional Random Field model has been motivated from the Sklearn-CRFSuite (Pedregosa et al., 2011). It has been modified based on the features for Mahabharatha text.

3.4 Relationship Extraction

The Relationship Extraction architecture is represented in Figure 2. which involves co-referencing and RoBERTa for relationship extraction.The process of co-referencing involves replacing the pronouns by their respective proper nouns in the sentence. For each mention or a pair of mentions a set of features are crafted. The most likely antecedent is mapped to its corresponding mention.After the co-referencing phase, the text has the proper noun in place of the pronoun referencing it. Coreferrence Resolution has been implemented through Neural Coref model. This co-referenced data is sent to the OpenIE model which finds all the relationships in the data. The output is given as a triplet of entities and the relationship identified. The relationship triplets identified here has to be filtered according to the relationships mentioned in Mahabharat. The dataset is analysed to identify fourteen relationships as listed in Table 2 between the entities identified in the Mahabharat text. The dataset with entities, relationship labels and its tokens are given to the RoBERTa base Model. The relationship extractor is thus trained on the given dataset.

3.5 Event Analysis

The event analysis architecture is represented in Figure 3. The important tasks involved in event analysis are summary generation, question - answering and graphical representation of the insights obtained.

Summary Generation :

After the Mahabharat dataset has been tagged

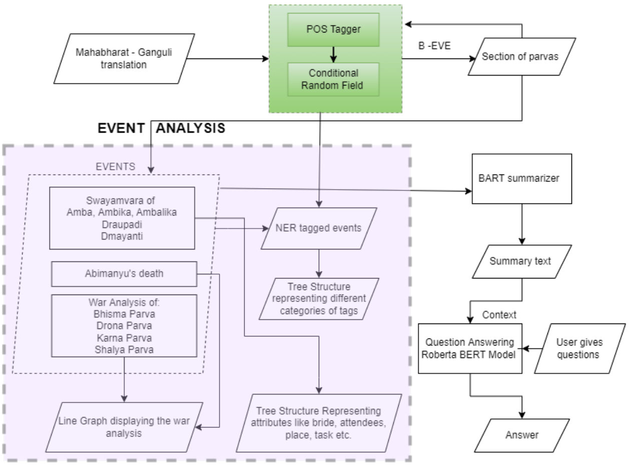

Mahabharat - Ganguli translation
POS Tagger
Conditional Random Field
B -EVE
Section of parvas
EVENT ANALYSIS
EVENTS
Swayamvara of Amba, Ambika, Ambalika Draupadi Dmayanti
Abimanyu's death
War Analysis of: Bhisma Parva Drona Parva Karna Parva Shalya Parva
NER tagged events
Tree Structure representing different categories of tags
Line Graph displaying the war analysis
Tree Structure Representing attributes like bride, attendees, place, task etc.
BART summarizer
Summary text
Context
Question Answering Roberta BERT Model
User gives questions
Answer

Figure 3: Event Analysis Architecture Diagram

by the NER tagger the section of the document describing the events are identified by the B-EVE and I-EVE tags. This sections of Parvas are given to the summary model to extract the summary of each event identified. The events identified include swayamvaras of Amba, Ambalika, Ambika, Draupadi and Damayanti, Abhimanyu's death and war analysis on different parvas. Different kinds of analysis are performed on the events and represented in graphs. The BART model generates summary. The embedding in a BART model is built on top of BERT. For every text sequence in its input, the BERT encoder outputs an embedding vector for each token in the sequence as well as an additional vector containing sentence-level information. The pre-training is done using the masked sequences. BART uses additional masking mechanisms as shown in Figure 4.

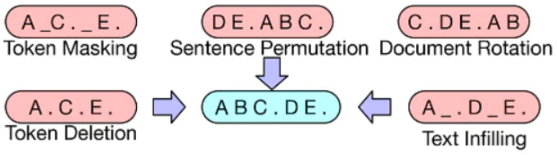


Figure 4: BART sentence masking

Question-Answering : The question-answering task is performed by whole word masking BERT model where the model gives the answer for the given question from the context. BERT model predicts the probability of each word being the starting and ending index of the answer span.
The BART model discussed in the previous phase outputs the summary of the event. The summary of the event is given to the question-answering model which identifies the answer span of data from the context for the specific question given by the user. The fine tuning of the question and answering model was done using the SQuAD(Stanford Question Answering Dataset).

TAG	EXPANSION
Attendees	People present at the event
Chosen one	The groom
Bride	The one who chooses
Father of Bride	King who organized the event
Place it was held	The kingdom
Weapon used	Weapons used in the event

Table 3: Template for the Swayamvara graph

Visualization : The insights of events of Mahabharat obtained on characters involved, place of the event etc. is represented by a tree structure as shown in Figure 12 which helps in comparing the event. Table 3 shows the entities of the template.

3.6 Character Analysis

Character Analysis is done so as to present a holistic view of the character in perspective of Mahabharatha. It includes the qualities of the character, their relationships, trials and tribulations they have been through and consequences of their actions. The Figure 5 depicts the flow of execution in performing this task. The Qualities are extracted using the 11 POS tags i.e [ADJ],[PUNCT], [ADV], [INTJ], [NOUN], [PROPN], [VERB], [CCONJ], [NUM],[PART],[AUX]. The extracted relations and the generated summary are used to create the character sketch with the help of a text generation model.

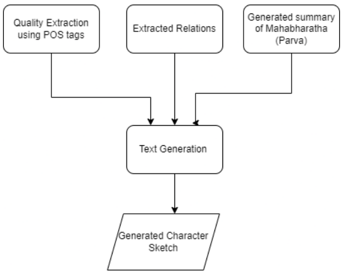

Quality Extraction using POS tags
Extracted Relations
Generated summary of Mahabharatha (Parva)
Text Generation
Generated Character Sketch

Figure 5: Character Sketch Diagram

Quality Extraction : The Qualities of a person define who he/she is in the story. These are exhibited using adjectives in the story. The adjectives have to be extracted using POS tags using BERT model. The BERT model is already fine-tuned on the UPenn-Treebank dataset with an accuracy of about 97.25%. The top 15 adjectives are chosen by frequency corresponding to the character as they distinctly represent the character's qualities.

Summary Generation : The summary is generated using the BART model built on top of the BERT model. The input is given parva wise to the summary generation model so that necessary information is captures which can be later used for any generation tasks.

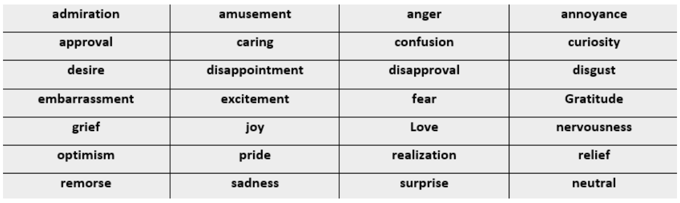
Text Generation : OpenAI's GPT-2 model is used for text generation. The GPT-2 transformer takes in a sequence of input tokens and then tries to generate multiple sequences of tokens in some chronological order so they form a meaningful sequence. The sequence of tokens generated are appended together to form a text. The Mahabharath summary alongwith the set of adjectives are taken as input collectively with some keywords such as "marriage", "parents", "born" etc. The model tries to decipher information related to these keywords and incorporates into the final text. Thus a character sketch is generated.

3.7 Emotion Analysis

Mahabharatha being an epic, contains a myraid of emotions throughout. It is important to identify these emotions and present them to the user in the most concise way possible without losing out information being captured. This is done by employing a emotion detection mechanism initially using a BERT model. This paper uses 26 different emotions as per the Go-Emotions dataset by Demsky et al. (Demszky et al., 2020). The extracted emotions are then fed to the text genration model collectively with the summarized text of Mahabharath (Parva-wise).

Emotion Detection : The Go-emotions dataset employs these 26 emotions as the necessary ones that can accurately capture different emotions while also not losing out on the context. BERT model is initially trained on an annotated parva of Mahabharatha with these set of emotions. The model is then deployed in the other 17 Parvas. Every sentence is attributed with some dominant emotion and the emotion which is dominant in one section of the parva is chosen as the right emotion. Every Parva contains about 100 sections and this procedure is followed for every Parva.

Text Generation : OpenAI's GPT-2 model is used for this text generation phase. The GPT-2 model takes in output from emotion detection phase and generated summary of Mahabharatha parva-wise. The keywords such as "feelings", "tension", "dilemma" are given as inputs alongwith the model so that the generated text is able to capture related incidents pertaining to those keywords. The text is presented to the user in the form of parapraghs.



admiration	amusement	anger	annoyance
approval	caring	confusion	curiosity
desire	disappointment	disapproval	disgust
embarrassment	excitement	fear	Gratitude
grief	joy	Love	nervousness
optimism	pride	realization	relief
remorse	sadness	surprise	neutral

Figure 6: Emotions used in the paper

TAG	COUNT
PERSON	1689
PLACE	173
EVENT	20
WEAPON and WAR STRATEGY	22
COMMUNITY	524
LIT and ART	23

Table 4: NER tags and their Expansions

4 Results

This section explains the results of semantic analysis tasks on Mahabharat like NER, Relationship extraction, Summary and Question Answering modules, Character Analysis and Emotion Analysis.

4.1 Entity Analysis

The entire text of Mahabharat has been annotated using a Conditional Random field model tuned for Named Entity Recognition.

Following this, a Relations Extraction model was built and the dataset for this model consists of parvas 5,6 of the Mahabharat text. The size of the training data is 2164 sentences. The model is capable of recognizing the relationships between the entities as one of the 14 categories as shown in Table 1. From the annotated data, the following inferences were made.

The number of unique entities in each category identified in the text are shown in Table 4. The frequency distribution of unique occurrences of each entity type is calculated and visualized as a pie chart as shown in Figure 7.

The "person" entities are then paired with each other based on their occurrences in the text. Two

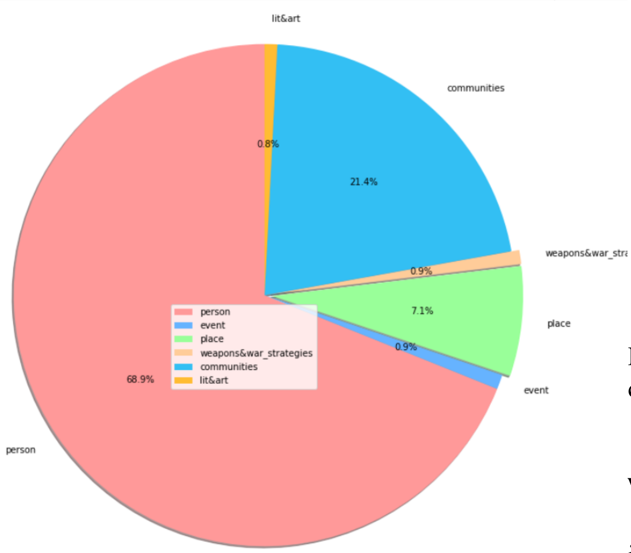
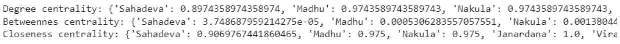


Figure 7: Distribution of Entities in the Text

person entities are said to be connected if they occur together in a span of 30 words. A network graph is thus constructed. The weight of the edges are assigned based the frequency of the particular pair. From this graph three different centralities are identified as shown in Figure 8.



```
Degree centrality: {'Sahadeva': 0.8974358974358974, 'Madhu': 0.9743589743589743, 'Nakula': 0.9743589743589743,
Betweennes centrality: {'Sahadeva': 3.748687959214275e-05, 'Madhu': 0.0005306283557057551, 'Nakula': 0.00138044
Closeness centrality: {'Sahadeva': 0.9069767441860465, 'Madhu': 0.975, 'Nakula': 0.975, 'Janardana': 1.0, 'Vira
```

Figure 8: Centralities

Each centrality represents a different kind of information about the entity.

Degree centrality is a measure of the number of other people a person is connected to. Higher the degree centrality, the more the person is connected.

Betweenness centrality is a measure of the popularity of the person. It is a measure of how many nodes are connected to others through this node.

Closeness centrality is a measure of the weightage of each of the connections in the graph. As the name indicates it shows how close each entity is to its neighbour.

The top fifteen entities with the highest closeness centralities are identified and a graph is plotted in the same manner as before, to show their interrelationships. Figure 9 shows that graph.

The entities included in this graph are those that have closest relationships with others. Arjuna has some of the most highly weighted edges implying that he is one of the most strongly connected character in the book. In addition this graph also shows the strong relationship between the five pandavas, with eachother and with Krishna.

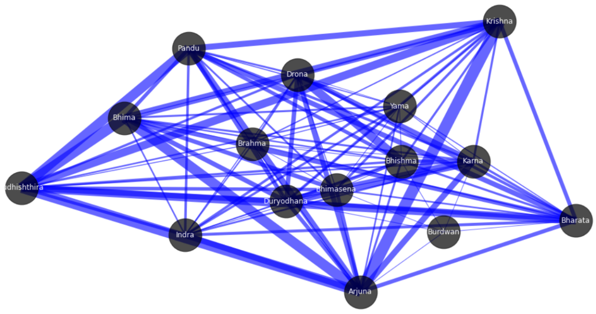


Figure 9: Interrelationships graph based on closeness centrality

In addition, the top fifteen entities based on their individual frequencies were identified and their interrelationships are represented in a network graph as shown in Figure 10.

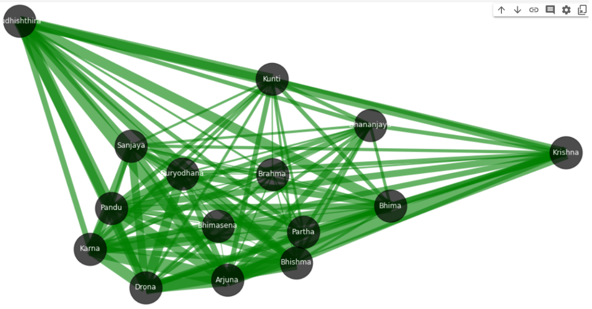


Figure 10: Interrelationships graph based on Frequency

This graph shows the most frequently occurring characters in the text. On comparison with the graph based on characters with highest centrality, two additional characters are identified - Sanjaya and Kunti. This shows that these two characters occur frequently in contexts outside of interactions.

4.2 Event Analysis

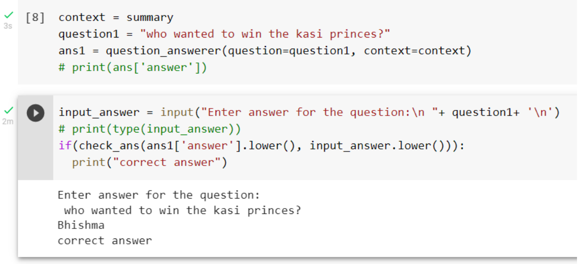
The entities tagged as events are identified and areas of the text where they are clustered are inferred to be the major events. These include Swayamvara, War parvas, Abimanyu's Death, dice game and disrobing. These events are analysed using Summary generation model, Question Answering models and through graphs.

4.2.1 Swayamvara

The Swayamvara event was analysed using the following tasks.

Summary Generation: The text pertaining to the event are fed as input to the summary generation model as a sequence of paragraphs. This model gives a 3-4 line output for the given input sample. The summary model is able to retain all important entity information and conveys the overall sequence of events in a succinct way.

Question Answering model for Quiz App: The output of the summary model is used as the context for the Question Answering model. The Q and A model has been used to build a quiz application, where the user is presented with an event and a set of questions pertaining to the question. The model identifies the answer from the context summary, and compares the answer it to the one given by the user. This has been demonstrated in Figure 11.



```
[8] context = summary
    question1 = "who wanted to win the kasi princes?"
    ans1 = question_answerer(question=question1, context=context)
    # print(ans['answer'])
```

```
input_answer = input("Enter answer for the question:\n "+ question1+ '\n')
# print(type(input_answer))
if(check_ans(ans1['answer'].lower(), input_answer.lower())):
  print("correct answer")
```

```
Enter answer for the question:
 who wanted to win the kasi princes?
Bhishma
correct answer
```

Figure 11: Snippet of the Quiz Application

In addition, there are provisions for the users to give their own questions to the model about each event.

Analysis and Graphical Representation: The event swayamvara is analysed using semantic graphs and a quiz app. A semantic graph with a fixed set of fields is defined for the events. By using the relationships identified in the event context, the values for the fields are filled.The template for the semantic graph of the event Swayamvara consists of the entity types mentioned in Table 3.This graph allows a comparison between the events. Figure 12 depicts the semantic graph for the Swayamvara of Amba, Ambika and Ambalika. This graph displays that the ceremony was held for three people together.

Figure 13 depicts the semantic graph for the Swayamvara of Panchali. The large number of attendess shows that a lot of important people took part in the competition.

4.3 War Analysis

The war events are distributed across four parvas such as Bhishma Parva, Drona Parva, Karna Parva, Shalya Parva. The performance of the Pandavas in each of these Parvas is plotted in a graph. If a member of the Pandava army is pierced or struck there is a small dip in the graph, if they are slayed a slightly bigger dip is shown and an even bigger dip is shown when they are slaughtered in bigger numbers. Similarly peaks of sizes proportional to the defeat of the Kouravas can be seen. Figure 14 and 15 show the graphs for the war parvas.

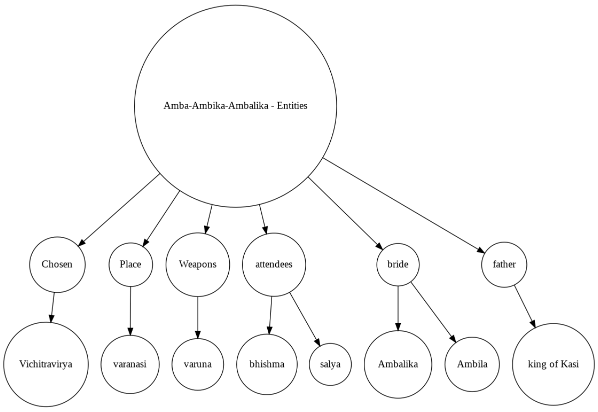


Figure 12: Semantic Graph depicting Amba's Swayamvara

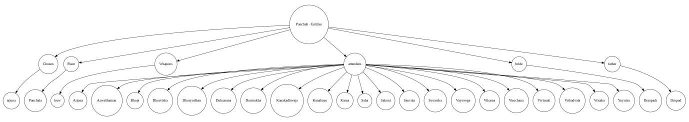


Figure 13: Semantic Graph depicting Panchali's Swayamvara

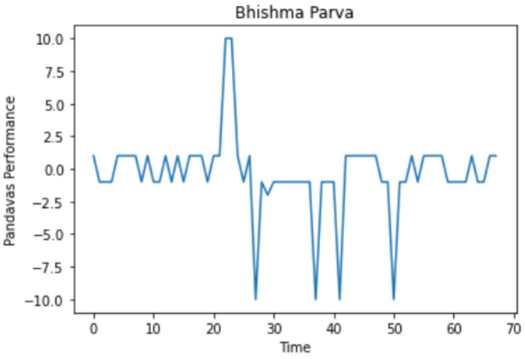


Figure 14: Line graph showing the performance of the Pandavas in Bhishma Parva

These graphs also allow us to track the battle sequence. The Kouravas saw major victories in Bhishma and Drona Parva, which is demonstrated by the major dips in the corresponding graphs. The victory of the Pandavas is shown in the final peak in the final graph.

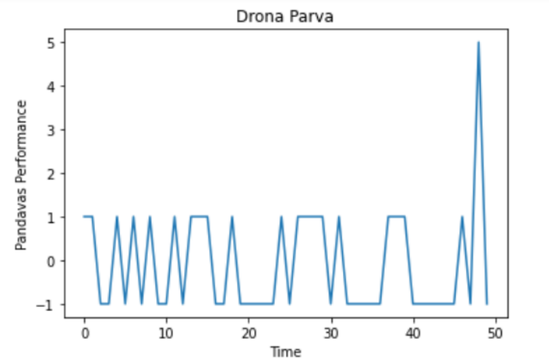


Figure 15: Line graph showing the performance of the Pandavas in Drona Parva

4.4 Abhimanyu’s Death

The events surrounding Abhimanyu’s death mark a turbulent battle between him and the kouravas. Abimanyu’s efforts and performance at the time of his death are plotted in Figure 16. The graph shown in Figure 16 demonstrates how well Abhimanyu fought before the time of his death.

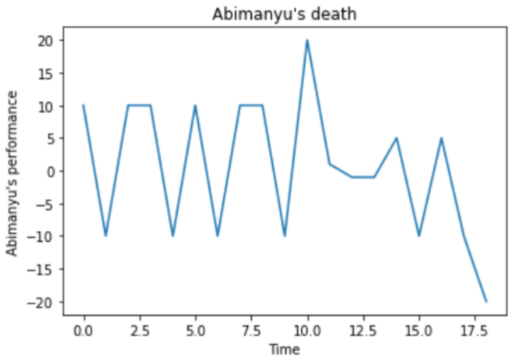
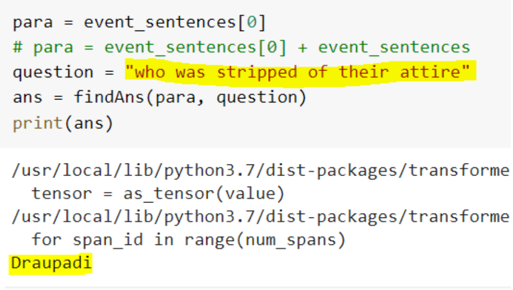


Figure 16: Line graph showing the performance of the Abhimanyu at the time of his death

4.5 Dice game and Disrobing

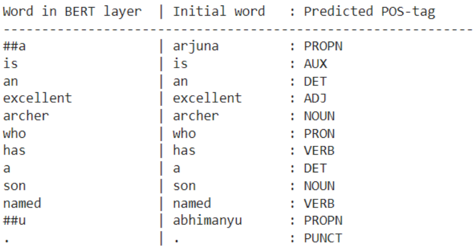
The dice game and disrobing event is analysed through summary generation and the q and a model and a quiz app is built for the same. Further there is also a provision for the users to ask questions related to a particular event as shown in Figure 17.



```
para = event_sentences[0]
# para = event_sentences[0] + event_sentences
question = "who was stripped of their attire"
ans = findAns(para, question)
print(ans)
```

```
/usr/local/lib/python3.7/dist-packages/transforme
  tensor = as_tensor(value)
/usr/local/lib/python3.7/dist-packages/transforme
  for span_id in range(num_spans)
Draupadi
```

Figure 17: Q and A for disrobing event

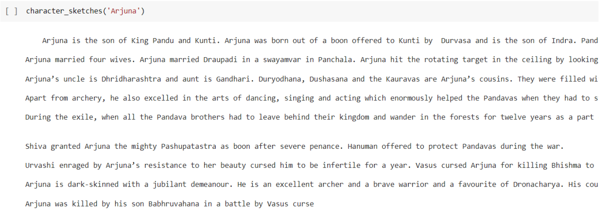


```
Word in BERT layer  | Initial word   : Predicted POS-tag
-------------------------------------------------------------
##a                 | arjuna         : PROPN
is                  | is             : AUX
an                  | an             : DET
excellent           | excellent      : ADJ
archer              | archer         : NOUN
who                 | who            : PRON
has                 | has            : VERB
a                   | a              : DET
son                 | son            : NOUN
named               | named          : VERB
##u                 | abhimanyu      : PROPN
.                   | .              : PUNCT
```

Figure 18: An Example of a sentence with POS tags

4.6 Character Analysis

The Figure 18 depicts an example of how BERT identifies POS tags and extracts [ADJ] tags for adjectives. These adjectives are used alongwith summary to generate the character sketch. The user can utilise the character sketch to learn about the particular character’s life, qualities and consequences of his/her actions instead of reading the entire text. The Figure 19 shows the final output of character sketch.



```
[ ]  character_sketches('Arjuna')

   Arjuna is the son of King Pandu and Kunti. Arjuna was born out of a boon offered to Kunti by  Durvasa and is the son of Indra. Pand
  Arjuna married four wives. Arjuna married Draupadi in a swayamvar in Panchala. Arjuna hit the rotating target in the ceiling by looking
  Arjuna’s uncle is Dhridharashtra and aunt is Gandhari. Duryodhana, Dushasana and the Kauravas are Arjuna’s cousins. They were filled wi
  Apart from archery, he also excelled in the arts of dancing, singing and acting which enormously helped the Pandavas when they had to s
  During the exile, when all the Pandava brothers had to leave behind their kingdom and wander in the forests for twelve years as a part

  Shiva granted Arjuna the mighty Pashupatastra as boon after severe penance. Hanuman offered to protect Pandavas during the war.
  Urvashi enraged by Arjuna’s resistance to her beauty cursed him to be infertile for a year. Vasus cursed Arjuna for killing Bhishma to
  Arjuna is dark-skinned with a jubilant demeanour. He is an excellent archer and a brave warrior and a favourite of Dronacharya. His cou
  Arjuna was killed by his son Babhruvahana in a battle by Vasus curse
```

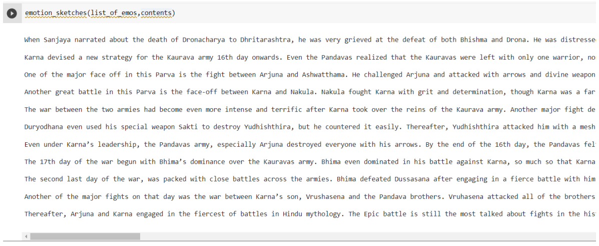
Figure 19: Character sketch of Arjuna

4.7 Emotion Analysis

The Emotion analysis analyzes the emotion sentence-wise and attributes the most occurring emotion to the section containing those sentences. The user can enter the Parva of choice for which the emotions are to be deduced. The emotions can be used as a basis for deriving any other analysis of the Mabhabaharatha text. The Figure 20 depicts the emotion sketch of Karna Parva.

5 Conclusion

The paper has discussed the different techniques used to analyze intricate events of Mahabharat and present them in a lucid and interesting manner to a user without prior knowledge of the text . Various entities present in the Mahabharat text were identified using a Conditional Random Field model



```
emotion_sketches(list_of_emos,contents)
```

```
When Sanjaya narrated about the death of Dronacharya to Dhritarashtra, he was very grieved at the defeat of both Bhishma and Drona. He was distresse
Karna devised a new strategy for the Kaurava army 16th day onwards. Even the Pandavas realized that the Kauravas were left with only one warrior, no
One of the major face off in this Parva is the fight between Arjuna and Ashwatthama. He challenged Arjuna and attacked with arrows and divine weapon
Another great battle in this Parva is the face-off between Karna and Nakula. Nakula fought Karna with grit and determination, though Karna was a far
The war between the two armies had become even more intense and terrific after Karna took over the reins of the Kaurava army. Another major fight de
Duryodhana even used his special weapon Sakti to destroy Yudhishthira, but he countered it easily. Thereafter, Yudhishthira attacked him with a mesh
Even under Karna’s leadership, the Pandavas army, especially Arjuna destroyed everyone with his arrows. By the end of the 16th day, the Pandavas fel
The 17th day of the war begun with Bhima’s dominance over the Kauravas army. Bhima even dominated in his battle against Karna, so much so that Karna
The second last day of the war, was packed with close battles across the armies. Bhima defeated Dussasana after engaging in a fierce battle with him
Another of the major fights on that day was the war between Karna’s son, Vrushasena and the Pandava brothers. Vruhasena attacked all of the brothers
Thereafter, Arjuna and Karna engaged in the fiercest of battles in Hindu mythology. The Epic battle is still the most talked about fights in the his
```

Figure 20: Emotion Sketch of Karna Parva

after a comparative analysis. Once the entities were identified, the observations and inferences based on their count, frequency distribution and interactions have been recorded. Various events in the text including Swayamvara and War have been analysed using summary generation models and question answering models. The character analysis provides a first hand impression of the character under consideration and the trails and tribulations which the character has gone through. Emotion analysis draws the flow of emotions and reactions of events described in Mahabharat to be presented in a concise manner to the user. Interested readers can utilize the obtained results from this paper as an incentive for any additional work.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

A. Akbik, Tanja Bergmann, Duncan A. J. Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL*.

Adnan Akhundov, Dietrich Trautmann, and Georg Groh. 2018. Sequence labeling: A practical approach.

Debarati Das, Bhaskarjyoti Das, and Kavi Mahesh. 2016. A computational analysis of Mahabharata. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 219–228, Varanasi, India. NLP Association of India.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. *CoRR*, abs/2005.00547.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Bhuwan Dhingra, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2017. Linguistic knowledge as memory for recurrent neural networks. *CoRR*, abs/1703.02620.

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585:357–362.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

J. Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34:50–70.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

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

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In

Proceedings of the 2nd Workshop on Representation Learning for NLP, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Charles Sutton and Andrew McCallum. 2010. An introduction to conditional random fields.