A Comprehensive Study of Mahabharat using Semantic and Sentiment Analysis

Srijeyarankesh J S, Aishwarya K, Nithyasri L and **Shanmuga priya M**
Department of Information Science and Technology
College of Engineering Guindy, Anna University
{2018115109, 2018115008, 2018115063,2018115103} @ auist.net

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 depicted. Sentimental analysis is prevailing in various domains such as social media monitoring (Rodríguez et al., 2019), customer support management (Pankaj et al., 2019), and analysing customer feedback (Capuano et al., 2021).

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 goal of the project is to provide a comprehensive analysis of this rich ancient text to allow for ease of understanding of the same. We are proposing a completely automated technique using semantic and sentiment analysis.

The Mahabharat Ganguli translation is used for conducting such an analysis. Entity Analysis involves named entity recognition which helps in discovering many unfamiliar characters present in Mahabharat. Semantic Analysis is used to understand and interpret the insights of the story 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 Organization of the paper

The paper is organized into 5 different sections. Introduction gives a general idea of the goal of the paper and the motivation behind the proposal of the idea. Related Works provides a summary of works which are similar in nature and how those works can be further extended to meet our goals. The next section discusses the different approaches used and their role in achieving the goal of the project. Performance Analysis depicts the metrics used for comparing the results obtained the final output of the proposed idea. Conclusion acts as a footnote to encourage readers to utilize this paper to supplement their future works.

3 Related Works

Mahabharat is an epic with valuable lessons on life and values. An epic typically has a plot which impart life lessons to the readers. Plots of an epic have various constituents i.e. happiness, sadness, suffering, reversal, recognition of latest knowledge, surprise etc. 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.

(Das et al., 2016) have 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(Priyatno, 2019) 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 (Mohammad et al., 2020). 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.

(Chandra et al., 2022) have discussed the results of semantic and sentiment analysis on Bhagavad gita which revolves around two characters i.e Arjuna and Krishna. The sentiments discussed in the paper are classified into optimistic and pessimistic sentiments. There is no discussion about different types of emotions which can shed light on the change in tone as the story transpires. The semantic analysis of Bhagavad Gita discusses about the similarity between verses of three authors and not on the overall text. Using the discussed paper as a motivation, this paper emphasizes on performing a detailed study by considering 28 different emotions on Mahabharat text and discuss about the different characters involved in the epic, important events which transpire and an interactive question answering model.

Named Entity Recognition is identifying propernouns 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 is done by Yanyao Shen et.al (Shen et al., 2017) where a CNN-CNN-LSTM model is built for NER, in an iterative approach. They have used various selection strategies for NER such as least confidence and 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) Bidirectional Encoder Representations from Transformers. BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. BERT is 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.

4 Methods

This section describes the idea being proposed with the help of an overall system architecture represented in Figure 1. Using NLP techniques such as co-referencing, relationship extraction, analysis on events, 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 utilizes BART for summarizing and whole word masking 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 Mahabharat is used to draw the character sketch. The emotion sketch is derived from using BERT model by using emotions from Go-Emotion dataset (Demszky et al., 2020) . The generated summary along with emotions extracted in every parva is passed through a text generation model for generating an emotion sketch.

4.1 Raw Dataset

Kisari Mohan Ganguli's translation of the Sanskrit epic Mahabharat is the raw dataset. The raw data consists of eighteen books. They are Adi Parva, Sabha Parva, Vana Parva, Virata Parva, Udyoga

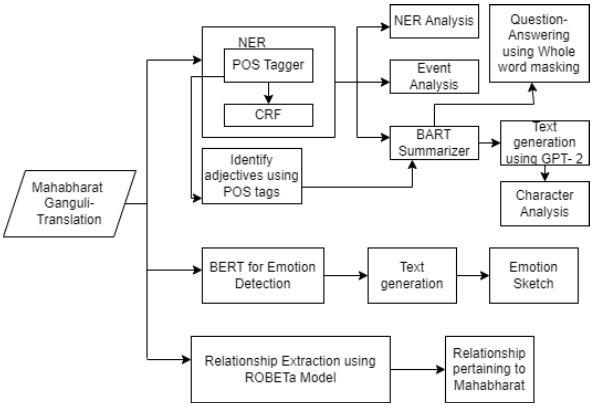

NER
POS Tagger
CRF
NER Analysis
Event Analysis
Question-Answering using Whole word masking
BART Summarizer
Text generation using GPT- 2
Character Analysis
Identify adjectives using POS tags
Mahabharat Ganguli-Translation
BERT for Emotion Detection
Text generation
Emotion Sketch
Relationship Extraction using ROBETa Model
Relationship pertaining to Mahabharat

Figure 1: Overall 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

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.

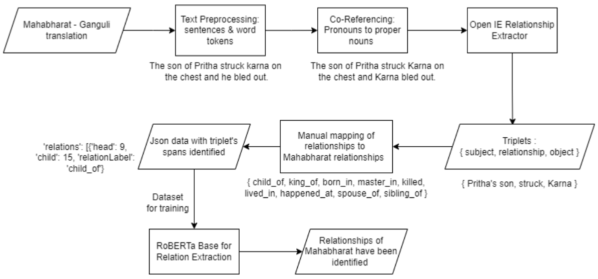

Mahabharat - Ganguli translation
Text Preprocessing: sentences & word tokens
The son of Pritha struck karna on the chest and he bled out.
Co-Referencing: Pronouns to proper nouns
The son of Pritha struck Karna on the chest and Karna bled out.
Open IE Relationship Extractor
Triplets : { subject, relationship,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

Figure 2: Relationship Extraction Architecture Diagram

4.2 Named Entity Recognition

In order to identify and extract the entities in the text, we curated a list of potential entities specific to Mahabharat which have been listed in the Table 1. To automatically identify these entities from the text, we trained a Conditional Random Field(CRF) model on the Mahabharat dataset. The CRF 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:

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

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 CRF model has been implemented through the Sklearn-CRFSuite (Pedregosa et al., 2011). It has been modified based on the features for Mahabharat text.

4.3 Relationship Extraction

The Relationship Extraction architecture is represented in Figure 2 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 (Mausam et al., 2012) 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

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

dataset.

4.4 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.

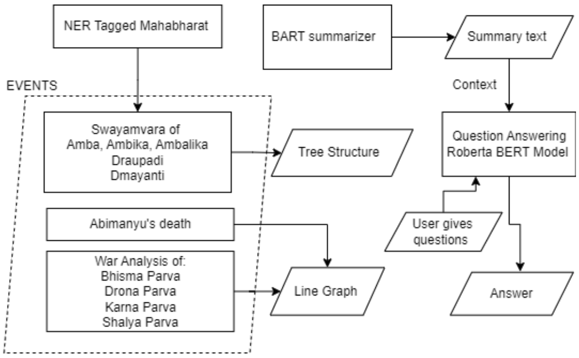


Figure 3: Event Analysis Architecture Diagram

Summary Generation :

After the Mahabharat dataset has been tagged 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.

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

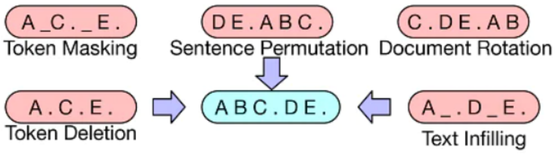


Figure 4: BART sentence masking

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 section 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 has been done using the SQuAD(Stanford Question Answering Dataset).

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.

4.5 Character Analysis

Character Analysis presents a holistic view of the character in perspective of Mahabharat. 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.

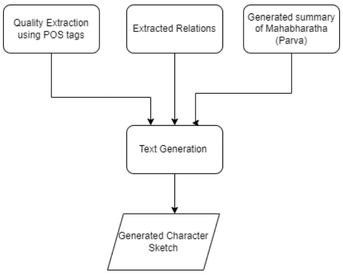
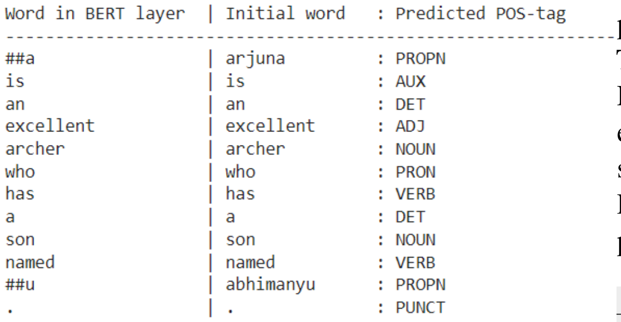

Quality Extraction using POS tags
Extracted Relations
Generated summary of Mahabharatha (Parva)
Text Generation
Generated Character Sketch

Figure 5: Character Sketch Diagram

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 with the help of BERT model. The BERT model is fine-tuned on the UPenn-Treebank dataset(Taylor et al., 2003) 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.



```
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 6: An Example of a sentence with POS tags

The Figure 6 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.

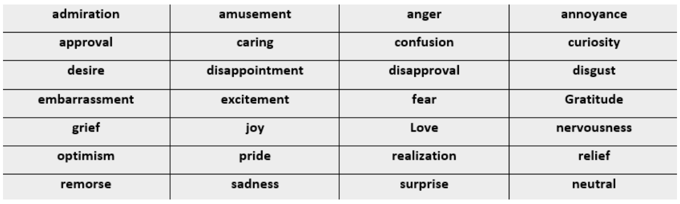
The summary of character analysis is generated using the BART model built on top of a base BERT model. The input is given parva wise to the summary generation model so that necessary information is captures which can be later used for any generation tasks.

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.

4.6 Emotion Analysis

Mahabharat 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 28 different emotions as per the Go-Emotions dataset. 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 28 emotions as depicted in Figure 7 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 Mahabharat 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.



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 7: Emotions used in the paper

The GPT-2 model takes in output from emotion detection phase and generated summary of Mahabharat 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.

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

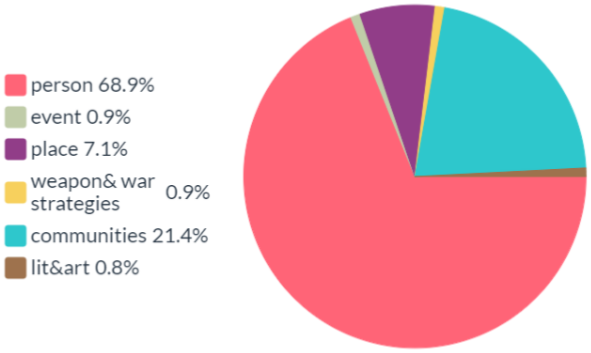


Figure 8: Distribution of Entities in the Text

The text is presented to the user in the form of parapraghs.

5 Analysis

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

5.1 Entity Analysis

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

Following this, a Relations Extraction model has been 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 have been 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 8.

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

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, 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 represented as a graph in Figure 9.

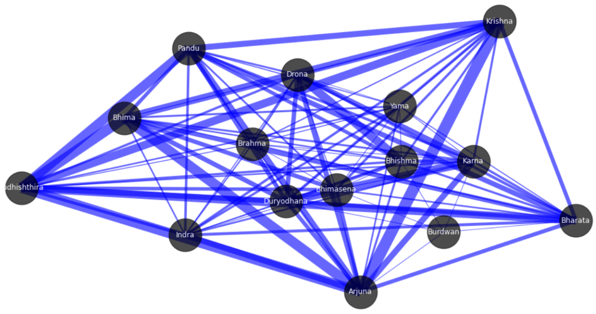


Figure 9: Interrelationships graph based on closeness centrality

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.

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.

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

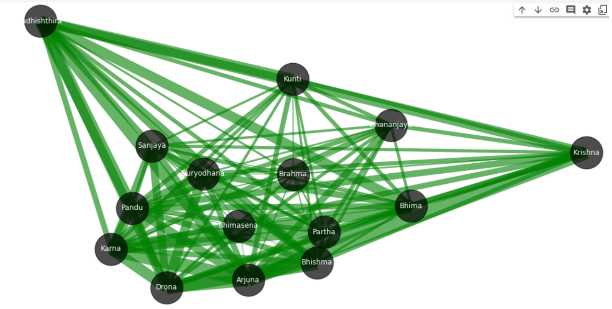
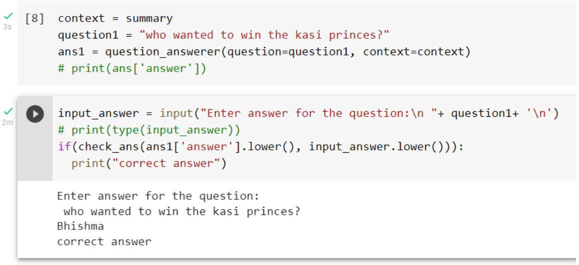


Figure 10: Interrelationships graph based on Frequency



```
[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 Question-Answering model

and Kunti. This shows that these two characters occur frequently in contexts outside of interactions.

5.2 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.

5.2.1 Swayamvara

The Swayamvara event has been 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.

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.

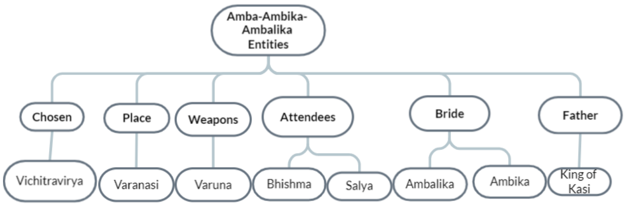


Figure 12: Semantic Graph depicting Amba's Swayamvara

5.3 War Analysis

The war events are distributed across four parvas: 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 13 and 14 depicts trends in two different war parvas.

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.

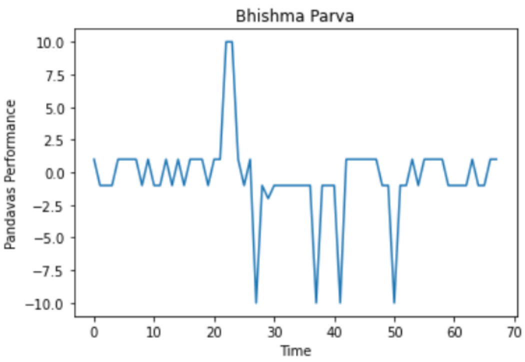


Figure 13: Line graph showing the performance of the Pandavas in Bhishma Parva

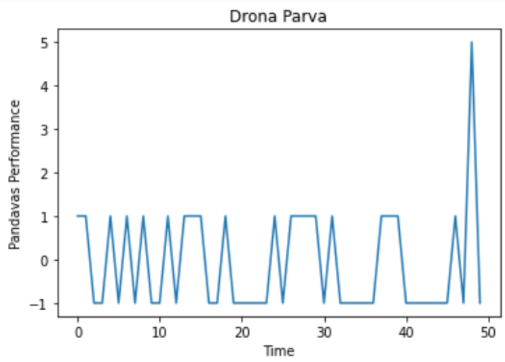


Figure 14: Line graph showing the performance of the Pandavas in Drona Parva

5.4 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 15. The graph shown in Figure 15 demonstrates how well Abhimanyu fought before the time of his death.

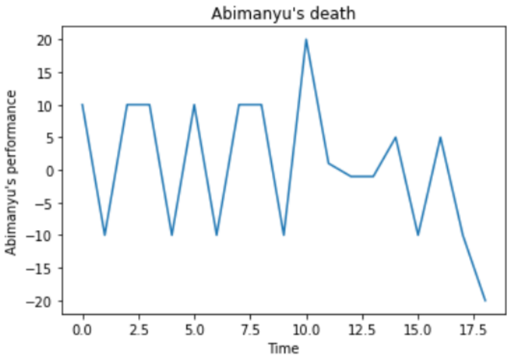
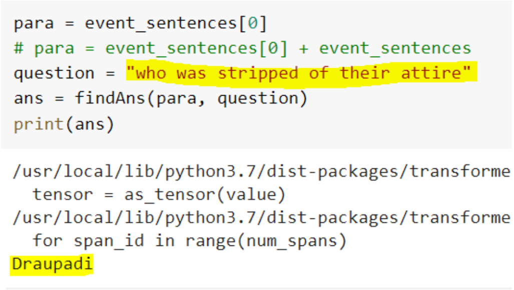


Figure 15: Line graph showing the performance of the Abhimanyu at the time of his death

5.5 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 16.



```
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 16: Q and A for disrobing event

5.6 Character Analysis

Character Analysis is performed by combining POS tagging model, BART model to generate summary and finally GPT-2 for generating text.The user has to give the Character's name as the input for the model to generate a summary of the character. The below character sketch shows a part of Arjuna's character sketch.

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. Pandu raised Arjuna as his child. He was the third eldest among the Pandavas. Arjuna's brothers are Yudhisthira, Nakula, Sahadeva and Bhima.

Arjuna married four wives. Arjuna married Draupadi in a swayamvar in Panchala. Arjuna hit the rotating target in the ceiling by looking at the water reflection below and married Draupadi. Their son was Shrutakarma. Arjuna married Balarama's sister Subhadra while he was in exile. Their son was Abhimanyu. Arjuna married Ulupi, Naga princess whilst in forest preparing for war. Their son was Iravan. Arjuna married Chitrangadha, princess of Manipura. Their son was Babhruvahana.

Arjuna is dark-skinned with a jubilant demeanour. He is an excellent archer and a brave warrior and a favourite of Dronacharya. His courage and proficiency was noted among his brothers during the Kurukshetra war. He is depicted to be handsome and furious. He emerged victorious in his battle against Karna who was equally good in archery. He was righteous and fought for the right cause.He was loved for his humility. Arjuna was killed by his *son Babhruvahana in a battle by Vasus curse*

5.7 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 below paragraph shows a part of emotion sketch output of Drona Parva.

The Kauravas ***grief*** *over the fall of Bhishma and are very* ***sad****. The* ***desire*** *for revenge is very high.*
Kauravas were ***optimistic*** *and Duryodhana made Drona the commander-in-chief of the Kuru. The* ***pride*** *of Kuru thus rested upon him. The Pandas were* ***surprised*** *to fight their teacher as leader of Kauravas and were* ***nervous****.*
Krishna asks Arjuna to be ***optimistic*** *about the war. There's confusion among the warriors due to* ***nervousness*** *of Pandavas. Krishna's words provided* ***relief*** *and* ***joy*** *for the Pandavas.*
Yudhishthira fought ***bravely*** *and killed a lot of warriors on Kaurava side. Duryodhana was* ***angry*** *over Bhishma's fall and plotted Yudhishthira's death, to isolate and kill him. Arjuna* ***consoles*** *his brother. Day ends with Pandavas having a upper hand.*
The following day, Dronacharya killed Satyajit. Arjuna killed Sampathakas bringing ***joy*** *and* ***admiration*** *for him among war.*

6 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 CRF model 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. Since Mahabharat is an extensive epic and the focus was rather on important events, interactive question answering model, analysis of characters and the overall sentiment expressed throughout different sections of the text, coherent text generation using large language models wasn't discussed in a larger context. This study can be used to discuss generation of coherent text for other literature using above mentioned techniques.

Aspect based sentiment analysis provides a deeper understanding on underlying emotions especially in a complex and intertwined epic like Mahabharat. However, since the general idea is to present to the reader an overall view of how emotions transpire through major events of the epic, paying too much focus on different sets of conflicting emotions within sentences strays way from the goal. This study can be used as a motivation for further diving deep into aspect-based sentiment analysis with compound analysis of emotions as the major task.

Large language models like GPT-2 by OpenAI can be used for generating coherent text. GPT-2 uses input text to set the initial context for further text generation. Mahabharatha is an extensive epic and the focus was rather on important events, interactive question answering model, analysis of characters and the overall sentiment expressed throughout different sections of the text, coherent text generation using large language models wasn't discussed in a larger context. This study can be used to discuss generation of coherent text for other literature using discussed techniques.

Interested readers can utilize the obtained results from this paper as an incentive for any additional work.

References

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

Nicola Capuano, Luca Greco, Pierluigi Ritrovato, and Mario Vento. 2021. Sentiment analysis for customer relationship management: an incremental learning approach. *Applied Intelligence*, 51(6):3339–3352.

Rohitash Chandra, Venkatesh Kulkarni, et al. 2022. Semantic and sentiment analysis of selected bhagavad gita translations using bert-based language framework. *IEEE Access*, 10:21291–21315.

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 Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

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.

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.

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)*.

Saif M. Mohammad et al. 2020. Practical and ethical considerations in the effective use of emotion and sentiment lexicons.

Pankaj, Prashant Pandey, Muskan, and Nitasha Soni. 2019. Sentiment analysis on customer feedback data: Amazon product reviews. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 320–322.

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.

Bijaksana Moch Arif Priyatno, Jentrisi. 2019. Clustering synonym sets in english wordnet. In *2019 7th International Conference on Information and Communication Technology (ICoICT)*, pages 1–4.

Axel Rodríguez, Carlos Argueta, and Yi-Ling Chen. 2019. Automatic detection of hate speech on facebook using sentiment and emotion analysis. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 169–174.

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.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The penn treebank: An overview. In *Treebanks*, pages 5–22. Springer Netherlands.