**Project - 1**

**Sentiment and Trend Analysis of Historical Literature using Hadoop and Hive**

**TEAM - 10**

**SRIJA BANDA**

**LOKESH CHOWDARY JASTI**

**VENKATA REVANTH REDDY PALAKONDA**

**TARUN KUMAR KANAKALA**

**MANISHA CHOWDARY AEINEDI**

**Introduction and Background**

The digitization of classic literature through initiatives like Project Gutenberg has made centuries-old texts accessible to modern data-driven analysis. With the rise of distributed computing frameworks, we now have the ability to computationally extract, analyze, and interpret the emotional and linguistic evolution embedded in literary masterpieces.

This project leverages Apache Hadoop and Hive to build a scalable sentiment and trend analysis pipeline. The goal is to examine five influential works—The Prince, Alice’s Adventures in Wonderland, Pride and Prejudice, Great Expectations, and Beowulf—to identify sentiment patterns, thematic word usage, and recurring phrase structures across literary genres and centuries.

**Project Objective**

The core objective of this project is to design and execute a distributed text analytics pipeline that can:

* Clean and preprocess large volumes of unstructured literary text
* Perform lemmatized word frequency analysis
* Score sentiment using the AFINN lexicon
* Track sentiment evolution across publication decades
* Extract common bigrams using a custom Hive UDF to reveal linguistic patterns

This workflow enables both macro-level trend detection (e.g., emotional arcs over time) and micro-level linguistic insights (e.g., key themes and expressions).

**Dataset and Technology Stack**

Corpus: Five public domain books sourced from Project Gutenberg, spanning different time periods and writing styles:

1. *The Prince* by Niccolò Machiavelli (1860 release)
2. *Alice’s Adventures in Wonderland* by Lewis Carroll (1838 release)
3. *Pride and Prejudice* by Jane Austen (1882 release)
4. *Great Expectations* by Charles Dickens (1802 release)
5. *Beowulf* translated by Lesslie Hall (1840 release)

**Tools and Frameworks:**

* Apache Hadoop (MapReduce): Parallel processing of large text data
* Apache Hive: Structured query execution and bigram UDF integration
* Stanford CoreNLP: Lemmatization of tokens
* AFINN-111 Lexicon: Lexicon-based sentiment scoring
* Java: Used for both MapReduce jobs and custom Hive UDF
* Maven: Project build and dependency management

**Task-by-Task Breakdown**

Task 1: Preprocessing and Tokenization

Text files were cleaned to remove metadata, headers, and footers using regex-based logic tailored to Project Gutenberg formats. Each file was processed using a Mapper to extract metadata (book title, publication year), lowercase the content, and remove punctuation and stopwords.

Exampl Output:  
(Beowulf, 1840) warrior hall dragon death gold

Task 2: Lemmatized Word Frequency Analysis

Using Stanford CoreNLP, the tokens were lemmatized to unify variations (e.g., “running” → “run”). The MapReduce job emitted (bookID, lemma, year) as key-value pairs, counting occurrences across texts. This revealed dominant themes:

* *The Prince:* “power,” “state,” “fear,” “rule”
* *Alice in Wonderland:* “rabbit,” “queen,” “dream,” “eat”
* *Pride and Prejudice:* “pride,” “marriage,” “letter,” “darcy”
* *Great Expectations:* “pip,” “prison,” “fear,” “expectation”
* *Beowulf:* “fight,” “dragon,” “warrior,” “glory”

Task 3: Sentiment Scoring

Sentiment analysis was applied to the lemmatized words using AFINN-111. Words with known sentiment scores were aggregated per book and year.

Results:

* *The Prince* scored moderately negative, reflecting themes of power, deception, and control.
* *Alice’s Adventures in Wonderland* had high sentiment variance, swinging between whimsical and dark tones.
* *Pride and Prejudice* held a predominantly positive tone, especially in character interactions.
* *Great Expectations* displayed emotional depth, with many negative phrases during Pip’s early years.
* *Beowulf* had neutral to mildly positive sentiment, showcasing themes of valor, heroism, and legacy.

Task 4: Temporal Sentiment Trend Analysis

Publication years were bucketed into decades (e.g., 1802 → 1800s). Sentiment scores were averaged per decade to detect long-term emotional trends.

Observations:

* 1800s literature (Dickens, Austen) leaned toward emotional introspection and social critique.
* 1830s and 1840s included more fantasy and mythic storytelling, contributing to fluctuating sentiment.
* *The Prince* (1860s) brought political realism and cynical tones, showing a sentiment dip.
* Earlier translated epics like *Beowulf* (1840) maintained a stoic emotional register, with less variance in tone.

Task 5: Bigram Analysis via Hive UDF

A Hive UDF was implemented in Java to generate bigrams from the cleaned and lemmatized text. Using LATERAL VIEW explode(), the bigrams were extracted and analyzed.

Frequent Bigrams:

* *The Prince:* “new principality,” “civil war,” “human nature”
* *Alice in Wonderland:* “white rabbit,” “drink me,” “mad hatter”
* *Pride and Prejudice:* “mr darcy,” “miss bennet,” “pride prejudice”
* *Great Expectations:* “young man,” “mr pip,” “blacksmith joe”
* *Beowulf:* “great hall,” “dragon fight,” “brave warrior”

These reflected the defining themes and memorable motifs of each book.

**Challenges and Solutions**

During the course of this project, several technical challenges were encountered and resolved through a combination of optimization techniques and custom development. One major issue was the inconsistent formatting of raw texts from Project Gutenberg. Since each book had varying metadata, headers, and footers, a universal cleaning approach was ineffective. To address this, a custom regex-based cleaner was developed for each text file, ensuring standardized and clean input for analysis.

Another performance bottleneck arose during sentiment lookup operations. Since sentiment scoring relies on mapping each lemma to its AFINN score, repeated file reads led to latency. This was optimized by loading the AFINN lexicon into Hadoop’s Distributed Cache, allowing fast and parallel access across all nodes during MapReduce processing.

The lemmatization process, powered by Stanford CoreNLP, initially consumed significant memory and CPU resources, especially when processing large texts. This was mitigated by reusing CoreNLP pipeline instances across batches and tuning JVM memory settings, which drastically reduced overhead and stabilized performance during text processing.

For bigram extraction, early results were flooded with irrelevant or meaningless word pairs, often due to the presence of stopwords in both positions. A refined solution was implemented by applying strong stopword filters to both tokens in each bigram, ensuring the results highlighted only meaningful phrase structures relevant to each book.

When developing and deploying the custom Hive UDF for bigram generation, testing proved challenging, especially without a live Hive environment. To overcome this, a local Hive testing scaffold was created with representative data, allowing the UDF to be validated efficiently before integration into production workflows.

Finally, limitations inherent to Hive’s User-Defined Table-Generating Functions (UDTFs) surfaced, especially when attempting to use them in WHERE or GROUP BY clauses. This was resolved by employing LATERAL VIEW in combination with subqueries, enabling smooth expansion and filtering of bigram outputs within Hive queries.

**Results and Insights**

The project successfully extracted and analyzed sentiment trends, word frequencies, and linguistic patterns from five foundational literary texts:

* Each book's sentiment signature aligned with its genre and narrative style.
* Lemmatization improved term unification and accuracy of word-based analysis.
* Bigram analysis revealed recurring motifs—critical for understanding authorial style and text themes.
* The modular pipeline design ensures scalability for larger corpora or real-time text streams.

**Conclusion**

This project highlights the utility of distributed computing for literary analysis. By leveraging Hadoop MapReduce and Hive, we efficiently processed and extracted insights from large-scale historical text. This hybrid of computational linguistics and big data offers new ways to interpret classic literature, understand emotional tone over time, and reveal stylistic patterns of iconic authors.

The methodology developed here can be scaled to broader digital humanities applications, such as analyzing entire genres, comparing literary movements, or creating sentiment-based recommender systems for readers.