Exploring Formula 1 Through Natural Language Processing

NLP MINI-PROJECT

- KANISHK SINGHANIA

NLP - 35

Abstract

Formula 1 (F1) is a high-speed, high-stakes motorsport that attracts millions of fans worldwide. In this project, we apply Natural Language Processing (NLP) techniques to analyze F1 race commentary data from the web. Our objective is to extract valuable insights from textual race updates and enhance our understanding of the sport. We employ web scraping, text preprocessing, and keyword extraction methods to develop functionalities for recognizing overtakes, detecting pit stops, spotting retirements, identifying podium finishes, and extracting fastest lap times. Our analysis reveals trends and patterns in race events, providing valuable information for F1 enthusiasts and stakeholders.

Introduction

Formula 1 is a premier motorsport characterized by fast-paced races, cutting-edge technology, and intense competition. Analyzing race data is crucial for teams, broadcasters, and fans to understand race dynamics, driver performances, and strategic decisions. Traditional methods of data analysis often focus on quantitative metrics like lap times and positions. However,

race commentary provides a rich source of qualitative information that can complement quantitative analysis. Natural Language Processing (NLP) techniques enable us to extract meaningful insights from textual race updates, offering a new perspective on F1 races.

Methodology

We utilize Python programming language and popular NLP libraries such as BeautifulSoup and NLTK for our analysis. The methodology consists of several steps:

- *Web Scraping*: We scrape race commentary data from reputable F1 websites, capturing textual updates from live race events.
- *Text Preprocessing*: We preprocess the raw text data by removing noise, such as HTML tags and special characters, and standardizing text format for consistency.
- *Keyword Extraction*: We extract relevant keywords and phrases from the preprocessed text using techniques like tokenization and part-of-speech tagging.
- *Functionality Development*: Based on the extracted keywords, we develop functionalities to recognize overtakes, detect pit stops, spot retirements, identify podium finishes, and extract fastest lap times.

Functionalities Implemented

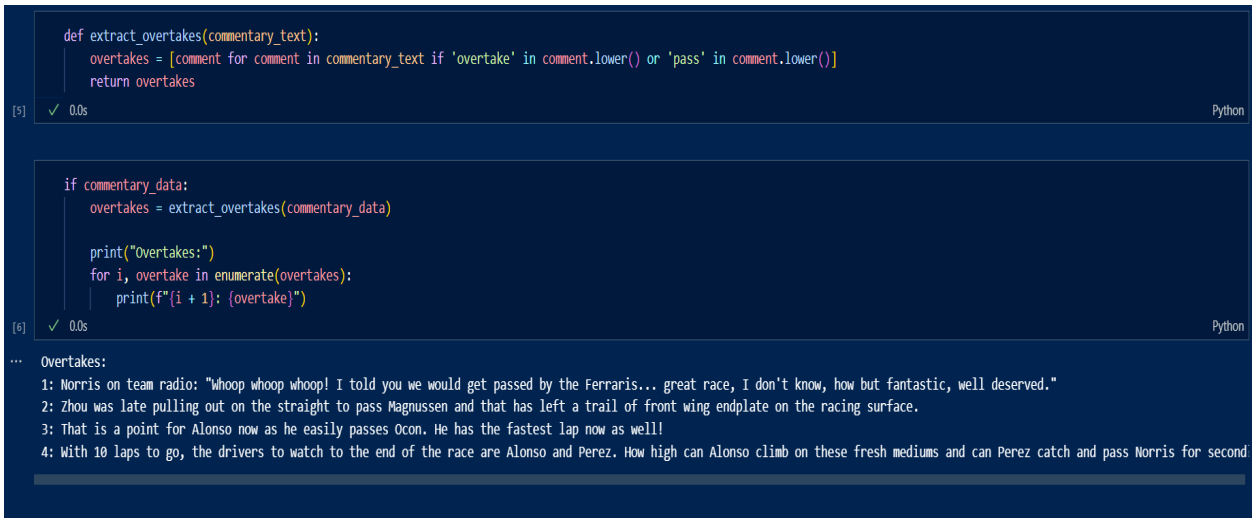
- *Recognizing Overtakes*: We identify instances where one driver overtakes another during the race, providing insights into driver maneuvers and race dynamics.
- *Detecting Pit Stop*s: We detect when drivers make pit stops, a critical aspect of race strategy, by analyzing commentary updates related to tire changes and refueling.
- *Spotting Retirements*: We pinpoint moments when drivers retire from the race due to mechanical issues or accidents, highlighting the impact on race outcomes.

- *Identifying Podium Finishes*: We identify the top three finishers (podium positions) in each race, showcasing the performance of leading drivers and teams.
- *Extracting Fastest Laps*: We extract the fastest lap times recorded by drivers in each race, indicating their pace and performance under optimal conditions.

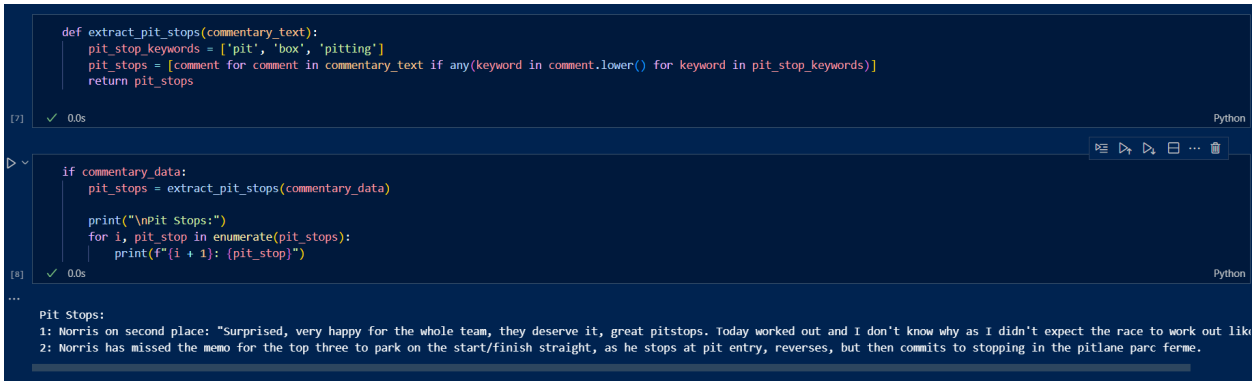
Results and Discussion

Our analysis of race commentary data reveals fascinating insights into F1 races:

- Overtakes are frequent and often decisive, with drivers strategically maneuvering to gain positions



- Pit stops play a crucial role in race strategy, influencing tire management, fuel consumption, and overall race pace.



```
def extract_pit_stops(commentary_text):
    pit_stop_keywords = ['pit', 'box', 'pitting']
    pit_stops = [comment for comment in commentary_text if any(keyword in comment.lower() for keyword in pit_stop_keywords)]
    return pit_stops
```

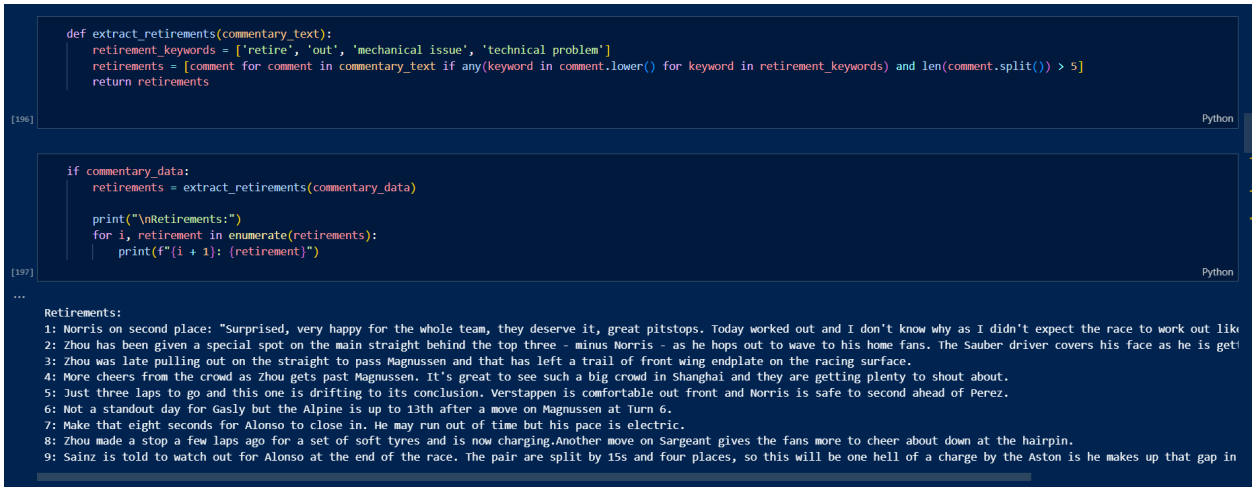
[7] ✓ 0.0s Python

```
if commentary_data:
    pit_stops = extract_pit_stops(commentary_data)

    print("\nPit Stops:")
    for i, pit_stop in enumerate(pit_stops):
        print(f"{i + 1}: {pit_stop}")
```

[8] ✓ 0.0s Python

```
Pit Stops:
1: Norris on second place: "Surprised, very happy for the whole team, they deserve it, great pitstops. Today worked out and I don't know why as I didn't expect the race to work out lik
2: Norris has missed the memo for the top three to park on the start/finish straight, as he stops at pit entry, reverses, but then commits to stopping in the pitlane parc ferme.
```

- Retirements can dramatically alter race dynamics, leading to shifts in leaderboard positions and team strategies.



```
def extract_retirements(commentary_text):
    retirement_keywords = ['retire', 'out', 'mechanical issue', 'technical problem']
    retirements = [comment for comment in commentary_text if any(keyword in comment.lower() for keyword in retirement_keywords) and len(comment.split()) > 5]
    return retirements
```

[196] Python

```
if commentary_data:
    retirements = extract_retirements(commentary_data)

    print("\nRetirements:")
    for i, retirement in enumerate(retirements):
        print(f"{i + 1}: {retirement}")
```

[197] Python

```
Retirements:
1: Norris on second place: "Surprised, very happy for the whole team, they deserve it, great pitstops. Today worked out and I don't know why as I didn't expect the race to work out lik
2: Zhou has been given a special spot on the main straight behind the top three - minus Norris - as he hops out to wave to his home fans. The Sauber driver covers his face as he is get
3: Zhou was late pulling out on the straight to pass Magnussen and that has left a trail of front wing endplate on the racing surface.
4: More cheers from the crowd as Zhou gets past Magnussen. It's great to see such a big crowd in Shanghai and they are getting plenty to shout about.
5: Just three laps to go and this one is drifting to its conclusion. Verstappen is comfortable out front and Norris is safe to second ahead of Perez.
6: Not a standout day for Gasly but the Alpine is up to 13th after a move on Magnussen at Turn 6.
7: Make that eight seconds for Alonso to close in. He may run out of time but his pace is electric.
8: Zhou made a stop a few laps ago for a set of soft tyres and is now charging.Another move on Sargeant gives the fans more to cheer about down at the hairpin.
9: Sainz is told to watch out for Alonso at the end of the race. The pair are split by 15s and four places, so this will be one hell of a charge by the Aston is he makes up that gap in
```

- Podium finishes reflect the competitive nature of F1, with drivers and teams vying for top honors in each race.

```
def extract_radio_communications(commentary_text):
    radio_keywords = [
        r"\bsquad\b",
        r"\bteam radio\b",
        r"\bsliding\b"
    ]
    radio_communications = [comment for comment in commentary_text if any(re.search(keyword, comment, flags=re.IGNORECASE) for keyword in radio_keywords)]
    return radio_communications
```

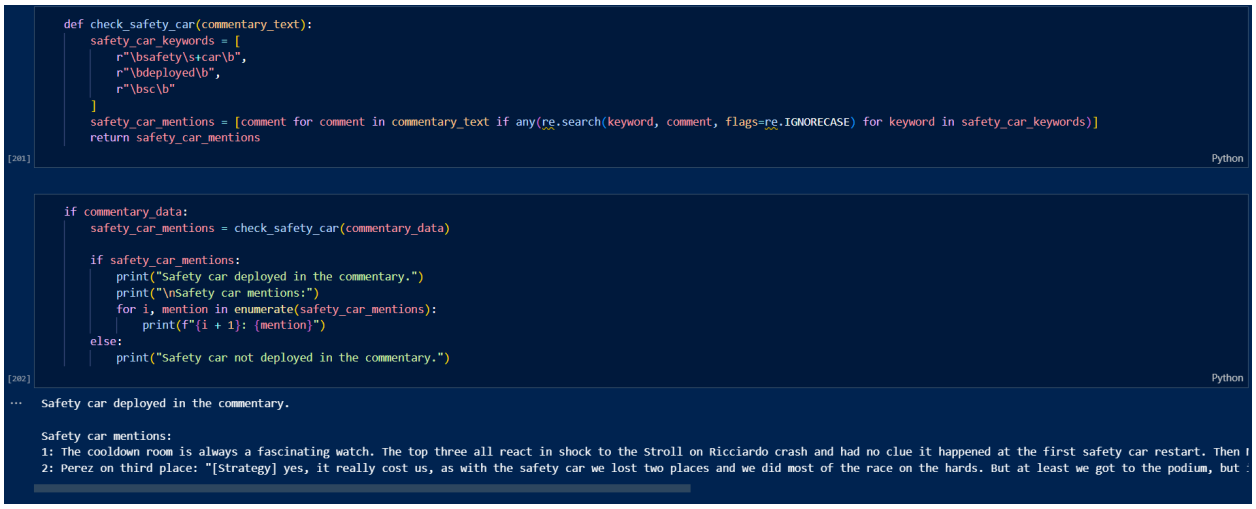
[199] Python

```
if commentary_data:
    radio_communications = extract_radio_communications(commentary_data)

    print("\nRadio Communications:")
    for i, radio in enumerate(radio_communications):
        print(f"{i + 1}: {radio}")
        comment = 'Hamilton team radio: "The car is sliding around everywhere, as if something is broken."'
print(comment)
```

[200] Python

```
Radio Communications:
1: Norris on team radio: "Whoop whoop whoop! I told you we would get passed by the Ferraris... great race, I don't know, how but fantastic, well deserved."
Hamilton team radio: "The car is sliding around everywhere, as if something is broken."
```

- Fastest lap times highlight individual driver performances and the technical capabilities of their cars.



```
def check_safety_car(commentary_text):
    safety_car_keywords = [
        r"\bsafety\s+car\b",
        r"\bdeployed\b",
        r"\bsc\b"
    ]
    safety_car_mentions = [comment for comment in commentary_text if any(re.search(keyword, comment, flags=re.IGNORECASE) for keyword in safety_car_keywords)]
    return safety_car_mentions
```

[201] Python

```
if commentary_data:
    safety_car_mentions = check_safety_car(commentary_data)

    if safety_car_mentions:
        print("Safety car deployed in the commentary.")
        print("\nSafety car mentions:")
        for i, mention in enumerate(safety_car_mentions):
            print(f"{i + 1}: {mention}")
    else:
        print("Safety car not deployed in the commentary.")
```

[202] Python

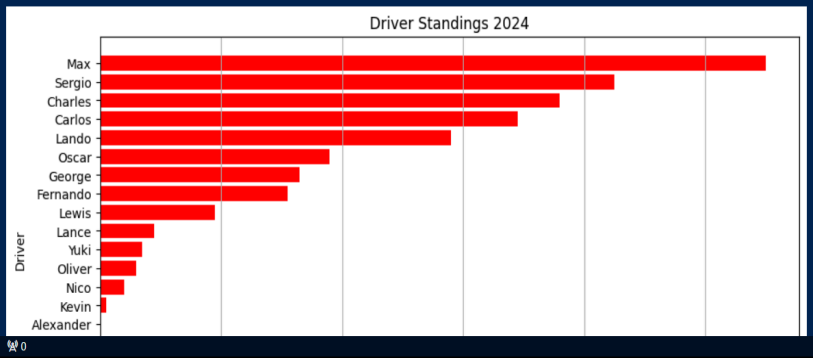
```
Safety car deployed in the commentary.

Safety car mentions:
1: The cooldown room is always a fascinating watch. The top three all react in shock to the Stroll on Ricciardo crash and had no clue it happened at the first safety car restart. Then
2: Perez on third place: "[Strategy] yes, it really cost us, as with the safety car we lost two places and we did most of the race on the hards. But at least we got to the podium, but
```

Visualization

We visualize our findings using bar charts, graphs, and tables to enhance understanding and interpretation. Bar charts represent driver and team standings over the course of the season, providing a snapshot of performance trends. Graphs display fastest lap times for each race, allowing comparison and analysis of driver performance. Tables summarize key statistics such as podium finishes and retirement incidents, facilitating easy reference and analysis.

```python
def plot_driver_standings(driver_standings):
    drivers = [name.split()[0] for name in driver_standings.keys()]
    points = [int(value) for value in driver_standings.values()]

    plt.figure(figsize=(10, 6))
    plt.barh(drivers, points, color='red')
    plt.xlabel('Points')
    plt.ylabel('Driver')
    plt.title('Driver Standings 2024')
    plt.gca().invert_yaxis()
    plt.grid(axis='x')
    plt.tight_layout(pad=3)  # Adjust layout with additional padding on the y-axis
    plt.show()
plot_driver_standings(driver_standings)
```

[207] Python

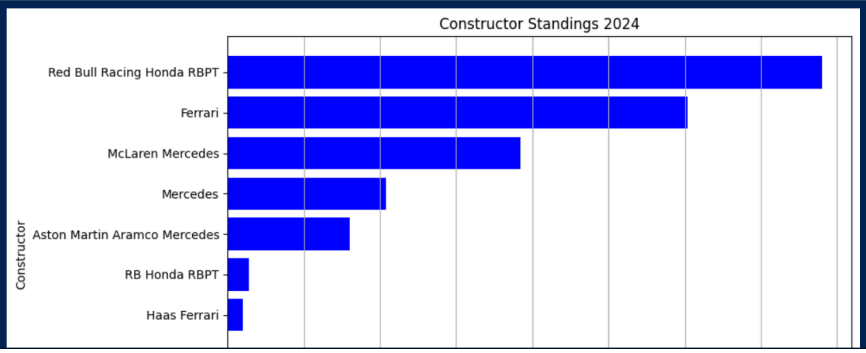


DRIVER STANDINGS RESULT

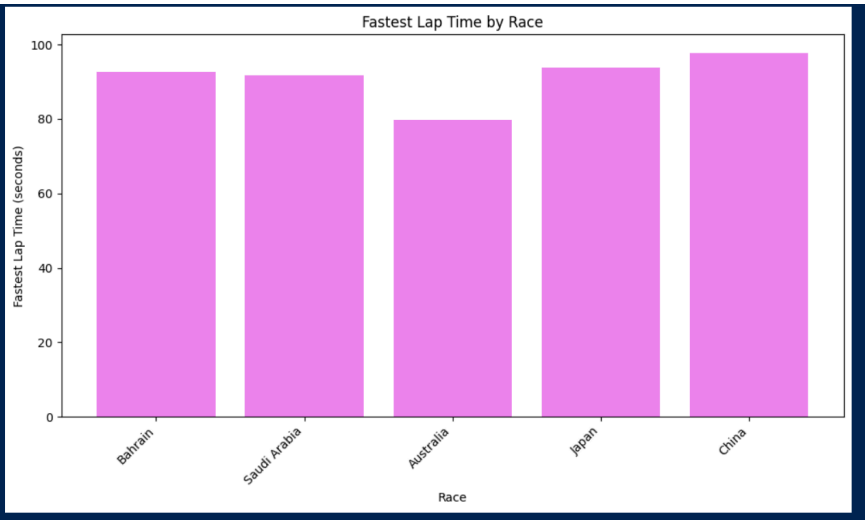
```python
def plot_team_standings(team_standings):
    teams = list(team_standings.keys())
    points = [int(value) for value in team_standings.values()]

    plt.figure(figsize=(10, 6))
    plt.barh(teams, points, color='blue')
    plt.xlabel('Points')
    plt.ylabel('Constructor')
    plt.title('Constructor Standings 2024')
    plt.gca().invert_yaxis()
    plt.grid(axis='x')
    plt.tight_layout()
    plt.show()
plot_team_standings(team_standings)
```

[211] Python

...



TEAM STANDINGS RESULT



TEAM STANDINGS RESULT

Conclusion:

My project demonstrates the potential of NLP techniques in analyzing F1 race commentary data and extracting valuable insights. By recognizing overtakes, detecting pit stops, spotting retirements, identifying podium finishes, and extracting fastest lap times, we offer a comprehensive analysis of race events. Our findings contribute to a deeper understanding of F1 races and provide actionable insights for teams, broadcasters, and fans. Future research could focus on refining NLP algorithms, improving data quality, and exploring additional functionalities for enhanced analysis.

References

- Formula1.com: Official Formula 1 website for race results and commentary.
- Python.org: Official website for Python programming language and related libraries.
- NLTK Documentation: Official documentation for the Natural Language Toolkit (NLTK) library.
- BeautifulSoup Documentation: Official documentation for the BeautifulSoup library for web scraping.