1. Introduction

Employee reviews provide valuable insights into a company's work environment, culture, and overall job satisfaction. This project focuses on analysing reviews of Samsung Electronics employees across different job roles, locations, and departments. Using Natural Language Processing (NLP), we aim to extract sentiments from textual feedback, categorize employee opinions, and identify key strengths and concerns.

Employee sentiment is crucial for understanding workplace dynamics, improving policies, and enhancing job satisfaction. By analysing employee feedback, Samsung Electronics can make informed decisions to improve its working environment and retain talent.

2. Dataset Overview

The dataset used in this analysis contains structured and unstructured data regarding employee experiences at Samsung Electronics. The dataset columns are

- **Title**: Job title of the employee
- **Place**: Location of employment
- **Job Type**: Full-time, part-time, or contractual employment
- **Department**: The business unit or functional area the employee belongs to
- **Date**: Date of review submission
- **Overall Rating**: General sentiment rating (e.g., Excellent, Good, Average, Poor)
- **Ratings on Specific Aspects**: Includes work-life balance, skill development, salary & benefits, job security, and career growth
- **Work Satisfaction**: A metric representing overall job satisfaction
- **Likes**: Textual review describing what employees appreciate about Samsung Electronics
- **Dislikes**: Textual review describing concerns or drawbacks of working at Samsung Electronics

3. Aim of the Project

The primary objective of this project is to analyse employee sentiment and gain insights into workplace satisfaction at Samsung Electronics. The analysis focuses on identifying the key aspects that employees appreciate the most, as well as the common challenges they face. By evaluating sentiment scores across different job-related factors, we aim to determine which aspects contribute most to employee satisfaction and which require improvement.

Additionally, the project examines variations in job satisfaction across different departments and locations, providing a comparative analysis of employee experiences. Understanding these patterns will help Samsung Electronics enhance its work environment, address concerns, and strengthen employee engagement.

4. NLP Techniques Used

To analyse employee reviews, we implemented several NLP techniques to extract meaningful insights from textual feedback. These techniques helped in understanding sentiment trends, identifying key themes, and visualizing the distribution of opinions.

1. Sentiment Analysis

Sentiment analysis was performed to classify employee feedback as positive, negative, or neutral. We used two major sentiment analysis tools:

- **TextBlob**: A simple NLP tool that determines the polarity of text, categorizing it as positive, negative, or neutral.
- **VADER (Valence Aware Dictionary and Sentiment Reasoner)**: A lexicon-based approach specifically designed for analysing sentiments in short texts and social media-like reviews.

2. Tokenization

Tokenization was applied to the "Likes" and "Dislikes" columns, breaking down textual reviews into individual words or tokens. This pre-processing step was essential for sentiment analysis, frequency analysis, and topic modelling.

3. Keyword Extraction

Keyword extraction was used to identify the most frequently occurring and meaningful words in employee reviews. This helped in highlighting common themes in both positive and negative feedback. Techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) were applied to extract significant words that provide insights into employee perceptions.

4. Topic Modelling

To uncover hidden themes in employee feedback, topic modelling was performed using **Latent Dirichlet Allocation (LDA)**. This method grouped words into clusters representing different discussion topics, allowing us to identify key themes related to workplace satisfaction, career growth, management, and company policies.

5. Data Visualization

Graphical representations were used to present sentiment distribution and key insights from the textual data. The visualizations included:

- Sentiment distribution across different job-related aspects.
- The frequency of positive and negative words in employee reviews.
- Comparative analysis of job satisfaction across departments and job roles.
- Topic clusters highlighting key discussion points in employee feedback.

5. Insights from the Analysis

1. Dominance of Positive Sentiment

- Employees largely hold **positive** opinions about work-life balance, skill development, salary, and job security.
- Skill development and salary benefits received the highest positive ratings.
- Employees appreciate the **learning opportunities, compensation, and workplace culture** at Samsung Electronics.

2. Neutral Sentiment is Significant

- Many employees rated job security, career growth, and work-life balance as "Neutral."

- This suggests **mixed opinions or uncertainty** in these areas, meaning some employees do not have strong positive or negative views.

3. Career Growth and Job Security are Areas of Concern

- Negative sentiment was more frequent in career growth and job security categories.
- Some employees expressed concerns about limited career advancement opportunities, job promotions, and long-term stability**.**
- The negative sentiment for career growth suggests that Samsung Electronics may need to improve internal career progression policies.

4. Common Positive Themes from Employee Reviews

- Good Salary Packages
- Opportunities for skill enhancement
- Work culture and team collaboration
- Free transportation and facilities

5. Common Negative Themes from Employee Reviews

- Limited career growth opportunities
- Inconsistent management or leadership issues
- Long working hours and work-life balance concerns
- Job security concerns in certain roles

These insights indicate areas where Samsung Electronics can focus on improvements.

6. Conclusion

The sentiment analysis of Samsung Electronics employee reviews reveals that **most employees have a positive experience**, particularly in terms of salary, skill development, and workplace environment. However, there are areas that require attention:

- **Career Growth & Promotions**: Employees feel that promotions and internal career advancements need to be more structured.
- **Job Security**: Some employees are concerned about long-term stability in their roles.

- **Work-Life Balance**: While generally positive, some employees mentioned long working hours as a challenge.

Recommendations for Samsung Electronics:

Enhance Job Security Policies – Offer better job stability assurances and transparent policies.
Monitor Work-Life Balance – Implement flexible work policies to address concerns about long working hours.

Improve Career Growth Opportunities – Introduce clear promotion pathways and training programs to support career progression.

SAMSUNG ELECTRONICS

In [1]:

```
from textblob import TextBlob
from nltk.sentiment import SentimentIntensityAnalyzer
import nltk
```

In [2]:

```
nltk.download("vader_lexicon")
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

Out[2]:

```
True
```

In [3]:

```
sia = SentimentIntensityAnalyzer()
```

In [5]:

```
import pandas as pd
df = pd.read_csv("New.csv")
df.head()
```

Out[5]:

	Title	Place	Job_type	Department	Date	Overall_rating	work_life_balance	skill_development	salary_and_benefits
0	Professional Logistics	Chennai	Full Time	SCM & Logistics Department	5 Sep 2023	Average	Average	Poor	Average
1	Supervisor Instructor	Noida	Full Time	Production & Manufacturing Department	1 Sep 2023	Excellent	Excellent	Excellent	Excellent
2	Quality Inspector	Noida	Full Time	Quality Assurance and Testing Department	3 Aug 2023	Good	Good	Good	Good
3	Lead Engineer	Noida	Full Time	Software Development Department	6 Aug 2023	Good	Average	Good	Excellent
4	Zonal Sales Manager	Chennai	Full Time	Retail & B2C Sales Department	1 Aug 2023	Good	Below Average	Average	Average

In [6]:

```
def get_textblob_sentiment(text):
    return TextBlob(str(text)).sentiment.polarity  # Ranges from -1 to 1

# Function to compute VADER sentiment
def get_vader_sentiment(text):
    return sia.polarity_scores(str(text))["compound"]  # Ranges from -1 to 1

# Classify sentiment based on score
def classify_sentiment(score):
    if score > 0.05:
        return "Positive"
    elif score < -0.05:
        return "Negative"
    else:
        return "Neutral"
```

In [7]:
```
df["Likes_sentiment"] = df["Likes"].apply(get_vader_sentiment).apply(classify_sentiment)
df["Dislikes_sentiment"] = df["Dislikes"].apply(get_vader_sentiment).apply(classify_sentiment)
```

In [8]:
```
df[["Likes", "Likes_sentiment","Dislikes", "Dislikes_sentiment"]].head()
```

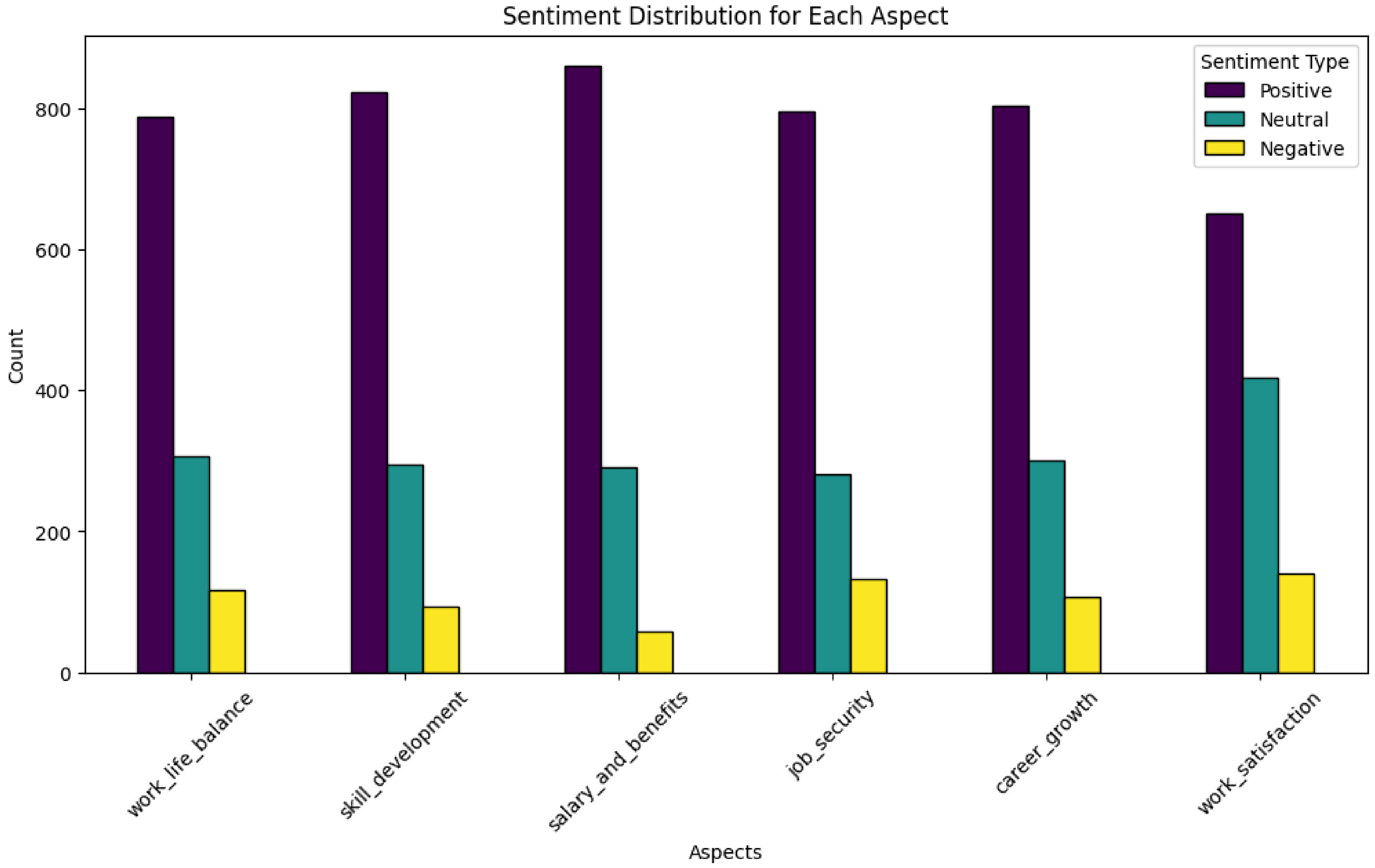
Out[8]:

	Likes	Likes_sentiment	Dislikes	Dislikes_sentiment
0	Company provide free of cost transportation\nG...	Positive	Montonous work\nManager don't assign responsib...	Negative
1	Samsung India electronics company bhut acchi h...	Neutral	Koi bhi problem nhi h company ki \nSab kuchh a...	Negative
2	Everything in Samsung India Pvt Ltd is done in...	Positive	Samsung private limited I eat discipline with ...	Positive
3	There is a lot of scope to learn.	Neutral	The rating system is not transparent. Maintain...	Positive
4	Wonderful data and tracking mechanism. Innovat...	Positive	At times we don't accept the market reality	Negative

In [9]:
```
# Define aspect columns for sentiment analysis
aspect_columns = ["work_life_balance", "skill_development", "salary_and_benefits", "job_security", "career_growth",

# Apply sentiment analysis and classification for each aspect
for col in aspect_columns:
    df[col + "_sentiment"] = df[col].apply(get_vader_sentiment).apply(classify_sentiment)

# Count positive, negative, and neutral sentiments for each aspect
aspect_sentiment_counts = {}
for col in aspect_columns:
    aspect_sentiment_counts[col] = df[col + "_sentiment"].value_counts()

# Convert to a DataFrame for better readability
aspect_sentiment_df = pd.DataFrame(aspect_sentiment_counts).T.fillna(0)  # Transpose for better structure
```

```python
# Display the sentiment counts for each aspect
print("\n=== Sentiment Distribution for Each Aspect ===\n")
print(aspect_sentiment_df)
import matplotlib.pyplot as plt
aspect_sentiment_df.plot(kind="bar", figsize=(12, 6), colormap="viridis", edgecolor="black")
plt.title("Sentiment Distribution for Each Aspect")
plt.xlabel("Aspects")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.legend(title="Sentiment Type")
plt.show()
```

```
=== Sentiment Distribution for Each Aspect ===

                     Positive  Neutral  Negative
work_life_balance         788      306       117
skill_development         823      295        93
salary_and_benefits       861      291        59
job_security              796      282       133
career_growth             804      300       107
work_satisfaction         651      419       141
```



Dominance of Positive Sentiment:

Across all categories (work-life balance, career development, salary & benefits, job security, career growth, and overall satisfaction), positive sentiment (dark purple) is significantly higher than neutral or negative sentiment. This suggests that employees generally have a favorable perception of these aspects.

Moderate Neutral Sentiment:

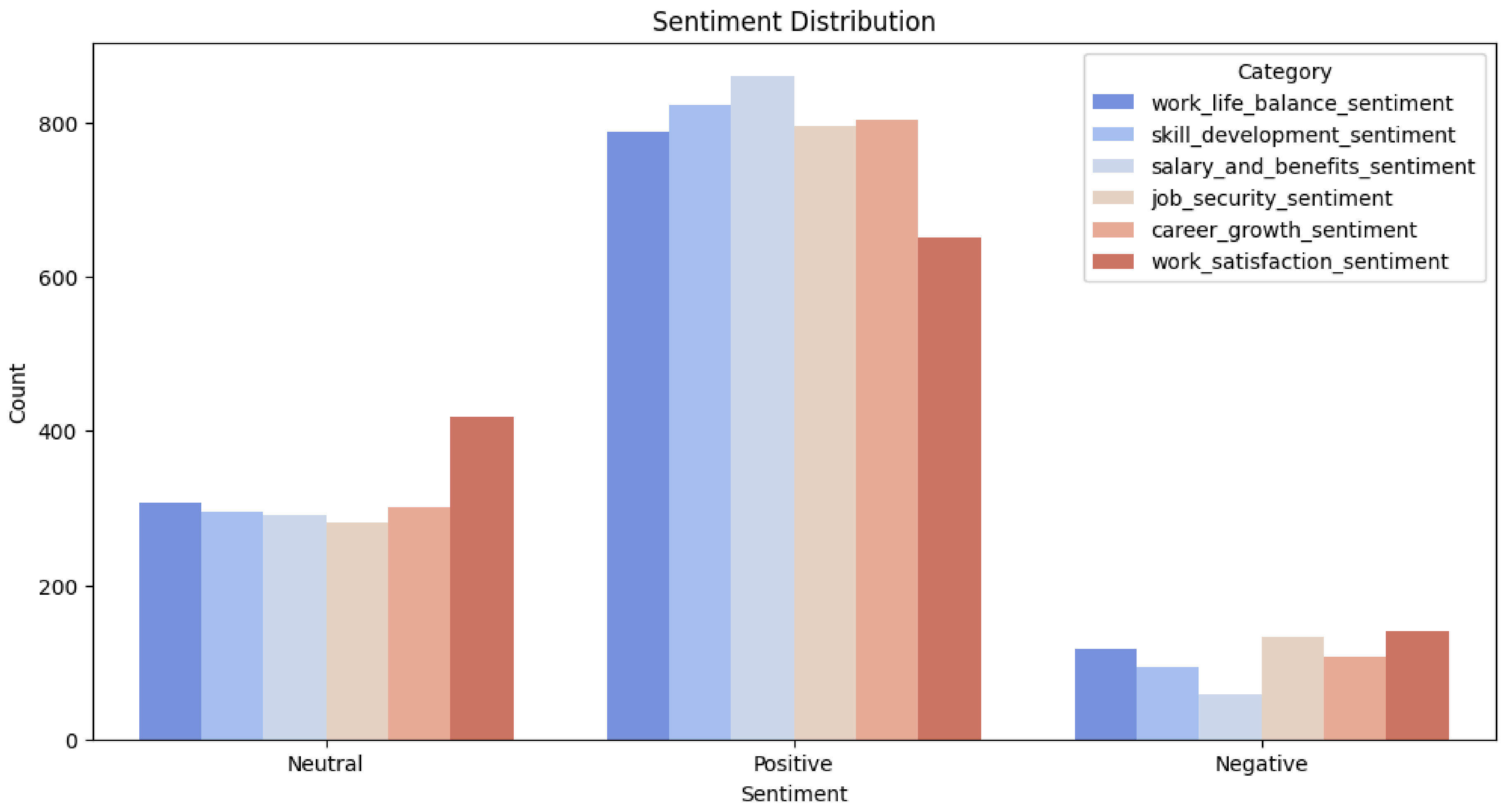
A noticeable portion of employees hold a neutral opinion (teal) on all key areas. This could indicate room for improvement or a lack of strong opinions on these aspects.

Least Prevalent Negative Sentiment:

Negative sentiment (yellow) is the least frequent across all categories, meaning dissatisfaction is present but relatively low. The job security and career growth categories have slightly higher negative sentiment compared to other factors, suggesting that some employees may be concerned about stability and advancement opportunities.

In [11]:

```python
import seaborn as sns
plt.figure(figsize=(12, 6))

# Count of each sentiment category
sentiment_counts = df[["work_life_balance_sentiment","skill_development_sentiment","salary_and_benefits_sentiment",
sns.countplot(x="value", hue="variable", data=sentiment_counts, palette="coolwarm")

plt.title("Sentiment Distribution")
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.legend(title="Category")
plt.show()
```



Overall Positive Sentiment is Dominant

Across all categories, positive sentiment has the highest count, suggesting that employees generally have a favorable experience in areas like work-life balance, skill development, salary, job security, career growth, and work satisfaction. Skill development and salary & benefits have the highest positive sentiment, indicating that employees appreciate opportunities for learning and compensation.

Neutral Sentiment is Considerable

Many responses fall into the neutral category, especially for work-life balance, skill development, and job security. This suggests that employees may have mixed or indifferent opinions in these areas, meaning companies might need to better define policies or improve communication to create stronger engagement.

Negative Sentiment is Relatively Low but Notable

While lower than positive sentiment, negative responses are more prominent in job security, career growth, and work satisfaction. Career growth and work satisfaction have the highest negative sentiment, which may indicate limited advancement opportunities or dissatisfaction with work conditions.

Key Takeaways:

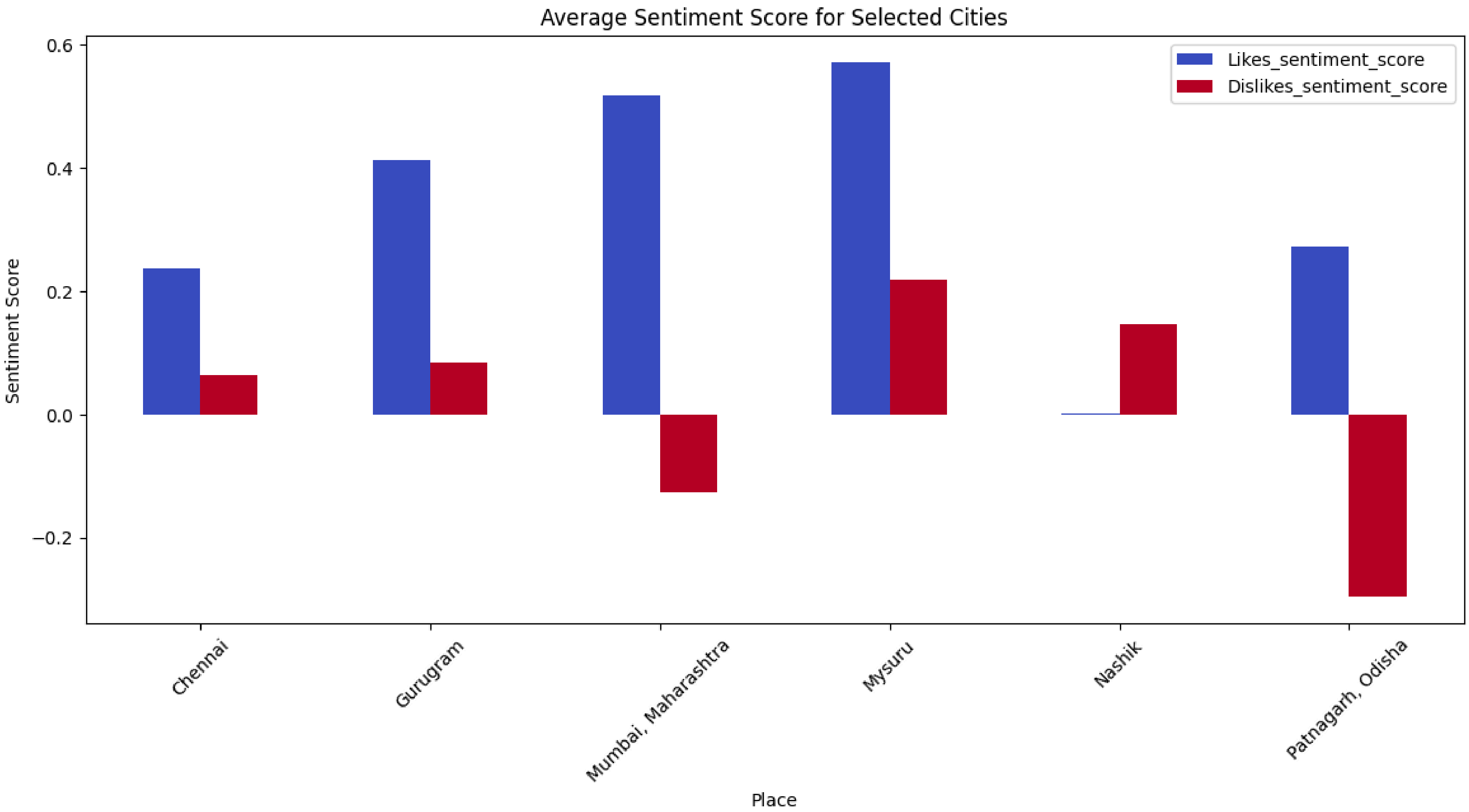
Strongest areas: Employees are highly positive about skill development and salary & benefits.

Areas for improvement: Career growth and job security show more negative sentiment, indicating potential concerns about promotions, long-term stability, or job prospects.

Neutral responses indicate room for engagement companies can work on clear policies and communication in work-life balance, job security, and skill development.

In [12]:
```
# Place-wise Sentiment Analysis (Filtered for Specific Cities)
selected_places = ["Chennai", "Mumbai, Maharashtra", "Gurugram", "Mysuru", "Nashik", "Patnagarh, Odisha"]

if "Place" in df.columns:
    # Compute sentiment scores for Likes and Dislikes
    df["Likes_sentiment_score"] = df["Likes"].apply(get_vader_sentiment)
    df["Dislikes_sentiment_score"] = df["Dislikes"].apply(get_vader_sentiment)

    # Filter and group by Place
    placewise_sentiment = df[df["Place"].isin(selected_places)].groupby("Place")[["Likes_sentiment_score", "Dislike

    # Print sentiment scores
    print("\n=== Place-wise Average Sentiment Scores ===\n")
    print(placewise_sentiment)

    # Plot place-wise sentiment scores
    plt.figure(figsize=(12, 6))
    placewise_sentiment.plot(kind="bar", figsize=(14, 6), colormap="coolwarm")
    plt.title("Average Sentiment Score for Selected Cities")
    plt.xlabel("Place")
    plt.ylabel("Sentiment Score")
    plt.xticks(rotation=45)
    plt.show()
else:
    print("No 'Place' column found in the dataset.")
```

```
=== Place-wise Average Sentiment Scores ===

                     Likes_sentiment_score  Dislikes_sentiment_score
Place
Chennai                           0.237795                  0.063585
Gurugram                          0.413580                  0.083260
Mumbai, Maharashtra               0.517157                 -0.125843
Mysuru                            0.571900                  0.217700
Nashik                            0.002433                  0.146800
Patnagarh, Odisha                 0.273200                 -0.296000
<Figure size 1200x600 with 0 Axes>
```



Overall Positive Sentiment Dominance:

Most cities have higher positive sentiment (blue bars) compared to negative sentiment (red bars). This indicates that employees generally have a favorable perception in these locations.

Cities with Strong Positive Sentiment:

Mysuru and Maharashtra have the highest likes sentiment scores, indicating a strong positive employee experience in these regions. Gurugram also shows significant positive sentiment, reflecting a good work environment.

Cities with Noticeable Negative Sentiment:

Maharashtra and Sambalpur, Odisha have higher dislike sentiment scores, meaning employees in these areas express more dissatisfaction than in other regions. Sambalpur, Odisha has the highest negative sentiment, suggesting possible work environment concerns.

Neutral to Mixed Sentiment Cities:

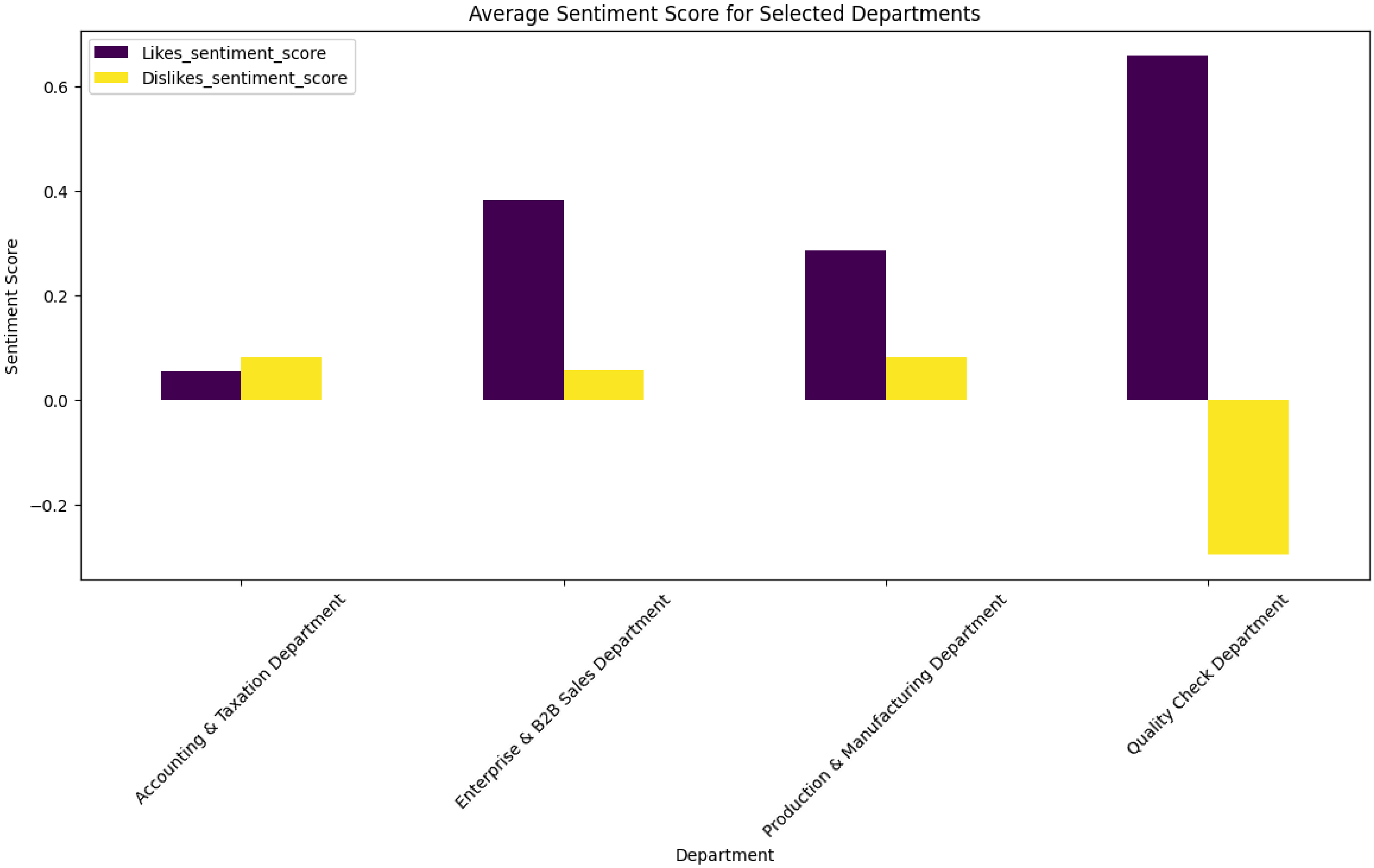
Chennai and Nashik have relatively balanced sentiment, with lower overall negativity but also moderate positive feedback.

Key Takeaways: Mysuru and Maharashtra stand out for having the most positive employee sentiment. Sambalpur, Odisha and Maharashtra show relatively higher dissatisfaction, indicating areas for potential improvement. Nashik and Chennai exhibit more neutral or mixed sentiment, meaning employee opinions are divided in these locations.

In [13]:

```python
# Select 5 departments for sentiment analysis
selected_departments = ["Quality Check Department", "Production & Manufacturing Department", "Enterprise & B2B Sale

# Check if 'Department' column exists
if "Department" in df.columns:
    # Apply VADER sentiment analysis to Likes and Dislikes
    df["Likes_sentiment_score"] = df["Likes"].apply(get_vader_sentiment)
    df["Dislikes_sentiment_score"] = df["Dislikes"].apply(get_vader_sentiment)

    # Filter for selected departments
    departmentwise_sentiment = df[df["Department"].isin(selected_departments)].groupby("Department")[["Likes_sentim

    # Plot sentiment scores for departments
    plt.figure(figsize=(12, 6))
    departmentwise_sentiment.plot(kind="bar", figsize=(14, 6), colormap="viridis")
    plt.title("Average Sentiment Score for Selected Departments")
    plt.xlabel("Department")
    plt.ylabel("Sentiment Score")
    plt.xticks(rotation=45)
    plt.show()
    print(departmentwise_sentiment)
else:
    print("No 'Department' column found in the dataset.")
```

```
<Figure size 1200x600 with 0 Axes>
```



```
                                       Likes_sentiment_score  \
Department
Accounting & Taxation Department                    0.052832
Enterprise & B2B Sales Department                   0.382216
Production & Manufacturing Department               0.285285
Quality Check Department                            0.659700

                                       Dislikes_sentiment_score
Department
Accounting & Taxation Department                       0.082200
Enterprise & B2B Sales Department                      0.056027
Production & Manufacturing Department                  0.080252
Quality Check Department                              -0.296000
```

Accounting & Taxation Department Likes Sentiment Score: 0.0528 (Neutral-Positive) Dislikes Sentiment Score: 0.0822 (Neutral-Positive)

Insight: Employees in this department have a neutral to slightly positive sentiment. The low but positive scores indicate that while there aren’t major complaints, there's no overwhelming enthusiasm either.

Enterprise & B2B Sales Department Likes Sentiment Score: 0.3822 (Moderately Positive) Dislikes Sentiment Score: 0.0560 (Neutral-Positive)

Insight: Employees in this department have a higher satisfaction level compared to others. The high likes sentiment suggests a positive work environment, while the neutral dislikes sentiment means there aren’t significant concerns.

Production & Manufacturing Department Likes Sentiment Score: 0.2853 (Moderately Positive) Dislikes Sentiment Score: 0.0803 (Neutral-Positive)

Insight: Employees in this department generally have a positive perception of their work. Dislikes sentiment is still neutral, meaning that while there may be some challenges, they aren’t major complaints.

Quality Check Department Likes Sentiment Score: 0.6597 (Highly Positive) Dislikes Sentiment Score: -0.2960 (Negative)

Insight: This department has the highest positive sentiment score (0.6597) in Likes, meaning employees love certain aspects of their work. However, Dislikes Sentiment is negative (-0.2960), meaning there are significant concerns despite liking the job overall.

Final Recommendations:

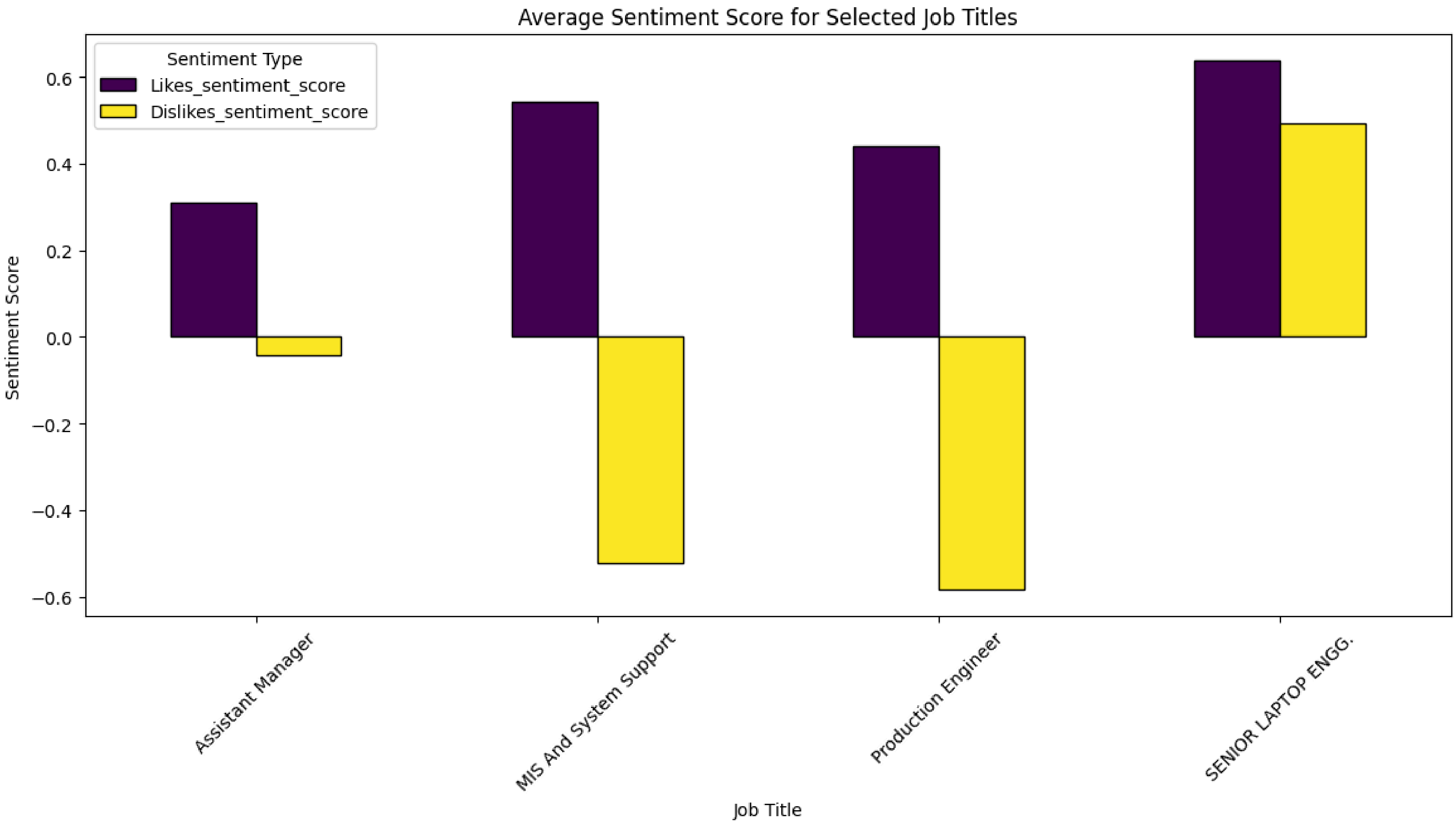
Quality Check needs focus on addressing employee dissatisfaction despite high job satisfaction. Production & Manufacturing could benefit from better work-life balance and team engagement. Sales & B2B is doing well; maintaining incentives and recognition programs will help sustain positive sentiment.

In [14]:

```
import matplotlib.pyplot as plt

# Define selected job titles
selected_title = [
    "Production Engineer",
    "SENIOR LAPTOP ENGG.",
    "MIS And System Support",
    "Assistant Manager",

]

if "Title" in df.columns:
    # Apply VADER sentiment analysis to Likes and Dislikes
    df["Likes_sentiment_score"] = df["Likes"].apply(get_vader_sentiment)
    df["Dislikes_sentiment_score"] = df["Dislikes"].apply(get_vader_sentiment)

    # Filter and compute mean sentiment scores for selected titles
    titlewise_sentiment = df[df["Title"].isin(selected_title)].groupby("Title")[["Likes_sentiment_score", "Dislikes

    # Print sentiment scores
    print("\n=== Average Sentiment Scores for Selected Titles ===\n")
    print(titlewise_sentiment)

    # Plot sentiment scores for job titles
    plt.figure(figsize=(12, 6))
    titlewise_sentiment.plot(kind="bar", figsize=(14, 6), colormap="viridis", edgecolor="black")
    plt.title("Average Sentiment Score for Selected Job Titles")
    plt.xlabel("Job Title")
    plt.ylabel("Sentiment Score")
    plt.xticks(rotation=45)
    plt.legend(title="Sentiment Type")
    plt.show()
```

```
else:
    print("No 'Title' column found in the dataset.")
```

```
=== Average Sentiment Scores for Selected Titles ===

                        Likes_sentiment_score  Dislikes_sentiment_score
Title
Assistant Manager                    0.310837                   -0.0424
MIS And System Support               0.542300                   -0.5216
Production Engineer                  0.440400                   -0.5849
SENIOR LAPTOP ENGG.                  0.636900                    0.4927
<Figure size 1200x600 with 0 Axes>
```



Assistant Manager

Moderate positive sentiment in Likes (0.3108) → Indicates that many employees in this role express satisfaction with certain aspects of their job. Almost neutral sentiment in Dislikes (-0.0424) → Shows that negative feedback is minimal and does not strongly impact overall satisfaction.

Interpretation: The role has an overall positive perception with very little dissatisfaction.

MIS And System Support

High positive sentiment in Likes (0.5423) → Employees appreciate many aspects of the job, possibly good work environment or career growth. Strong negative sentiment in Dislikes (-0.5216) → This indicates that, despite liking certain aspects, employees also express significant dissatisfaction about some factors.

Interpretation: The role has a mixed sentiment, where employees like key aspects but have serious concerns (e.g., workload, management, work-life balance).

Production Engineer

Fairly strong positive sentiment in Likes (0.4404) → Employees find the role engaging and rewarding in certain aspects.

Strong negative sentiment in Dislikes (-0.5849) → Indicates clear dissatisfaction, potentially due to working conditions, stress, or job security issues.

Interpretation: Employees appreciate aspects of the role but experience high dissatisfaction, making it a high-stress job with divided opinions.

Senior Laptop Engineer

Highest positive sentiment in Likes (0.6369) → Indicates that employees strongly appreciate their job, possibly due to good pay, work culture, or career growth.

Positive sentiment even in Dislikes (0.4927) → This is very unusual, suggesting that even negative feedback is expressed in a positive way.

Interpretation: Employees in this role have the highest job satisfaction, and even their concerns are not major complaints but rather constructive feedback.

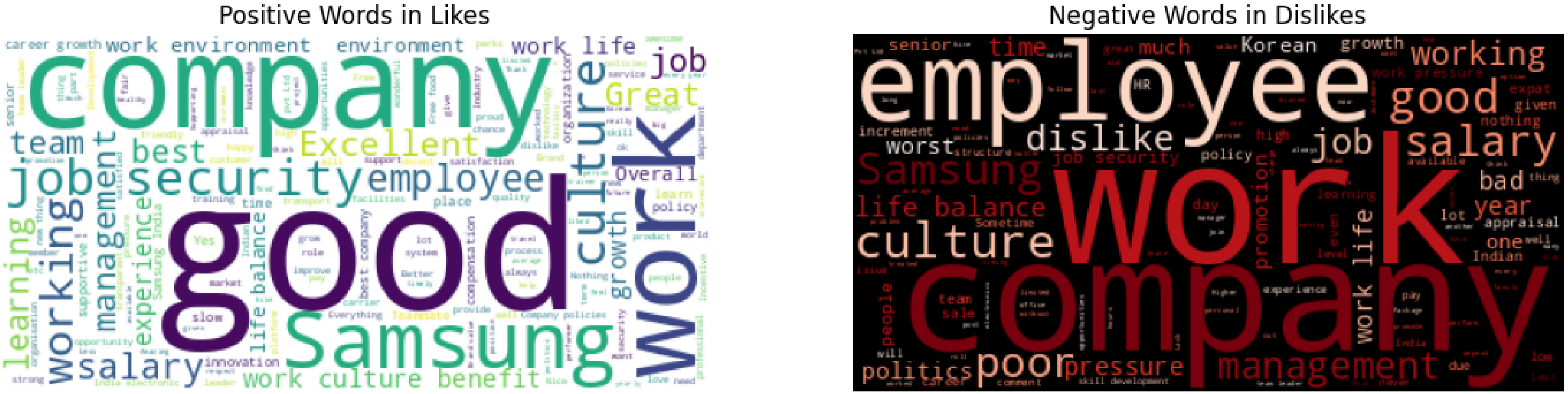
Best Perceived Job → Senior Laptop Engineer (high positive sentiment in both Likes & Dislikes).

Most Divided Sentiment → MIS And System Support & Production Engineer (both have strong positive and negative scores).

Least Complaints → Assistant Manager (neutral Dislikes sentiment).

Keyword Extraction

In [15]:
```
#Keyword Extraction(Using wordcloud and TF-IDF scores)
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Generate Word Cloud for Positive Likes & Negative Dislikes
positive_likes = " ".join(df[df["Likes_sentiment"] == "Positive"]["Likes"].dropna())
negative_dislikes = " ".join(df[df["Dislikes_sentiment"] == "Negative"]["Dislikes"].dropna())

# Plot WordClouds
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Positive Likes WordCloud
axes[0].imshow(WordCloud(background_color="white").generate(positive_likes))
axes[0].set_title("Positive Words in Likes")
axes[0].axis("off")

# Negative Dislikes WordCloud
axes[1].imshow(WordCloud(background_color="black", colormap="Reds").generate(negative_dislikes))
axes[1].set_title("Negative Words in Dislikes")
axes[1].axis("off")

plt.show()
```



Positive Words in Likes (Left Word Cloud)

The most frequently mentioned positive words are: "Good," "Company," "Samsung," "Security," "Culture," "Job," "Salary," "Learning," "Management," "Work-life balance," "Team," "Experience," "Supportive," and "Growth."

Employees appreciate job security, salary, and work culture at the company. Words like "learning" and "growth" indicate that employees value the career development opportunities provided. The presence of "team," "management," and "supportive" suggests that teamwork and leadership are well-regarded. The inclusion of "work-life balance" shows that some employees find it satisfactory.

Negative Words in Dislikes (Right Word Cloud)

The most frequently mentioned negative words are: "Work," "Company," "Employee," "Culture," "Politics," "Pressure," "Poor," "Salary," "Management," "Life balance," "Job security," "Promotion," "Dislike," and "Worst." "Politics" and "pressure" appear prominently, indicating concerns about workplace culture, stress, or internal politics. "Job security" and "promotion" show up as negative words, aligning with the previous sentiment analysis where these were identified as areas needing improvement.

"Salary" appears in both positive and negative clouds, meaning some employees find it good, while others feel it is inadequate. "Life balance" and "work pressure" suggest that while some employees are satisfied with work-life balance, others struggle with workload and stress. The presence of "management" in both positive and negative clouds indicates mixed reviews about leadership.

Key Takeways-

Strengths: Employees appreciate job security, salary, culture, learning opportunities, and management support. Concerns: Issues related to work pressure, internal politics, job security, promotions, and management practices need improvement. Actionable Recommendations:

Address workplace politics and pressure—Encourage open communication and fair policies. Improve job security and promotion opportunities—Provide clearer career growth paths. Work-life balance improvements—Reassess workload distribution to reduce employee stress.

In [16]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

# Fill NaN values with an empty string and combine Likes & Dislikes
combined_text = df["Likes"].fillna("") + " " + df["Dislikes"].fillna("")
```

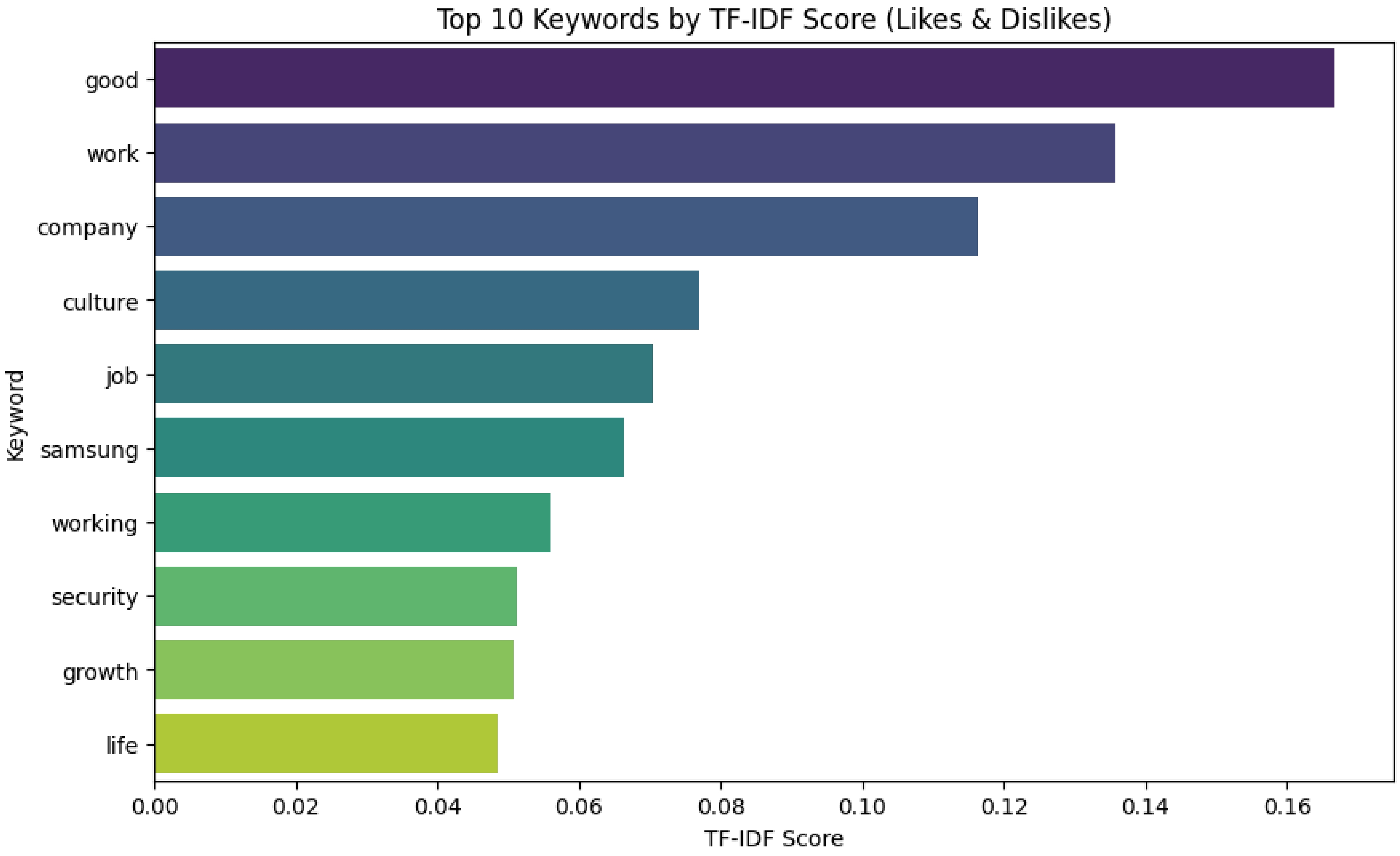
```
# Initialize TfidfVectorizer to remove common words and extract top keywords
vectorizer = TfidfVectorizer(stop_words="english", max_features=20)  # Adjust max_features as needed

# Fit and transform the data (TF-IDF matrix)
tfidf_matrix = vectorizer.fit_transform(combined_text)

# Get the feature names (keywords)
keywords = vectorizer.get_feature_names_out()

# Convert the TF-IDF matrix into a DataFrame for easier viewing
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=keywords)

# Extract top 10 keywords from Likes & Dislikes based on average TF-IDF score
top_keywords = pd.DataFrame(tfidf_df.mean(axis=0).sort_values(ascending=False).head(10), columns=["TF-IDF Score"])

print("🔹 **Top 10 Keywords from Likes & Dislikes based on TF-IDF:**")
print(top_keywords)
```

```
🔹 **Top 10 Keywords from Likes & Dislikes based on TF-IDF:**
          TF-IDF Score
good          0.166881
work          0.135696
company       0.116401
culture       0.076966
job           0.070383
samsung       0.066367
working       0.055934
security      0.051126
growth        0.050870
life          0.048423
```

In [17]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

# Fill NaN values with an empty string and combine Likes & Dislikes
combined_text = df["Likes"].fillna("") + " " + df["Dislikes"].fillna("")

# Initialize TfidfVectorizer
```

```
vectorizer = TfidfVectorizer(stop_words="english", max_features=20)  # Adjust max_features as needed

# Fit and transform the data (TF-IDF matrix)
tfidf_matrix = vectorizer.fit_transform(combined_text)

# Get the feature names (keywords)
keywords = vectorizer.get_feature_names_out()

# Convert the TF-IDF matrix into a DataFrame
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=keywords)

# Extract top 10 keywords based on average TF-IDF score
top_keywords = pd.DataFrame(tfidf_df.mean(axis=0).sort_values(ascending=False).head(10), columns=["TF-IDF Score"])

# Reset index to get keywords as a column
top_keywords = top_keywords.reset_index()
top_keywords.columns = ["Keyword", "TF-IDF Score"]

# Plot the top keywords using a bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x="TF-IDF Score", y="Keyword", data=top_keywords, palette="viridis")

# Add labels and title
plt.xlabel("TF-IDF Score")
plt.ylabel("Keyword")
plt.title("Top 10 Keywords by TF-IDF Score (Likes & Dislikes)")

# Show the plot
plt.show()
```

```
<ipython-input-17-4eab06be3dd8>:30: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `
hue` and set `legend=False` for the same effect.

  sns.barplot(x="TF-IDF Score", y="Keyword", data=top_keywords, palette="viridis")
```



Topic Modelling

In [20]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import pandas as pd

# Load and preprocess data
```

```python
df["Likes"] = df["Likes"].fillna("").astype(str)
df["Dislikes"] = df["Dislikes"].fillna("").astype(str)

# Use a single vectorizer for both Likes and Dislikes to ensure the same vocabulary
vectorizer = TfidfVectorizer(stop_words="english", max_df=0.95, min_df=2)  # Adjust min_df as needed
X_all = vectorizer.fit_transform(df["Likes"] + " " + df["Dislikes"])  # Combine both columns

# Apply same vocabulary for both Likes & Dislikes
X_likes = vectorizer.transform(df["Likes"])
X_dislikes = vectorizer.transform(df["Dislikes"])

# Number of topics
n_topics = 3

# LDA models
lda_likes = LatentDirichletAllocation(n_components=n_topics, random_state=42)
lda_dislikes = LatentDirichletAllocation(n_components=n_topics, random_state=42)

lda_likes.fit(X_likes)
lda_dislikes.fit(X_dislikes)

# Function to print topics with dynamic n_top_words
def print_topics(model, feature_names):
    available_words = len(feature_names)  # Ensure we don’t exceed available words
    for idx, topic in enumerate(model.components_):
        n_top_words = min(10, available_words)  # Prevent out-of-bounds error
        print(f"Topic {idx + 1}:")
        print([feature_names[i] for i in topic.argsort()[:-n_top_words - 1:-1]])
        print("\n")

# Print topics for Likes
print("\n🔹 Topics for 'Likes':")
print_topics(lda_likes, vectorizer.get_feature_names_out())

# Print topics for Dislikes
print("\n🔹 Topics for 'Dislikes':")
print_topics(lda_dislikes, vectorizer.get_feature_names_out())
```

```
🔹 Topics for 'Likes':
Topic 1:
['samsung', 'experience', 'like', 'excellent', 'brand', 'compensation', 'company', 'great', 'organization', 'india']


Topic 2:
['good', 'environment', 'salary', 'growth', 'working', 'company', 'overall', 'career', 'benefits', 'policy']


Topic 3:
['work', 'job', 'culture', 'company', 'security', 'life', 'balance', 'good', 'management', 'policies']



🔹 Topics for 'Dislikes':
Topic 1:
['samsung', 'like', 'politics', 'compensation', 'working', 'company', 'experience', 'brand', 'employee', 'opportunit
y']


Topic 2:
['good', 'growth', 'salary', 'working', 'career', 'appraisal', 'promotion', 'company', 'policy', 'people']


Topic 3:
['work', 'job', 'culture', 'security', 'dislike', 'life', 'balance', 'management', 'company', 'pressure']
```

Likes: Company Reputation & Compensation (Topic 1)

Employees appreciate Samsung as a strong brand and organization. Compensation appears to be a major positive factor. The experience of working at Samsung is generally seen as excellent.

Work Environment & Career Growth (Topic 2)

Positive work environment, salary, and career growth opportunities are valued. Benefits and company policies are seen as good.

Job Security & Work Culture (Topic 3)

Job security, company culture, and work-life balance are viewed positively. Management and policies are also considered good. Dislikes: Politics & Compensation Issues (Topic 1)

Despite compensation being mentioned in "Likes," it is also a concern in "Dislikes," indicating inconsistency in salary satisfaction. Office politics is a major complaint. Limited employee opportunities might be an issue.

Career Growth & Promotion Issues (Topic 2)

Growth and salary concerns exist, especially around career progression and appraisals. Company policies may not be perceived as favorable by some employees.

Work Culture & Pressure (Topic 3)

While culture and job security were mentioned as positives, they are also present in "Dislikes," possibly due to inconsistencies in different teams or roles. Work pressure and management issues may be affecting employee satisfaction.

Overall Takeaway: Strong brand, good compensation, and career growth opportunities attract employees. Concerns over office politics, inconsistent salary satisfaction, and work pressure need attention. Management and company policies have mixed reviews, indicating potential room for improvement in leadership and workplace culture.

Finding Important words by multiplying Lambda weight and TF-IDF scores

In [6]:

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
df=pd.read_csv('New.csv')

# Fit TF-IDF to the text corpus
tfidf_vectorizer = TfidfVectorizer(stop_words="english")
tfidf_matrix = tfidf_vectorizer.fit_transform(df["Likes"].dropna())

# Get vocabulary & word importance
feature_names = tfidf_vectorizer.get_feature_names_out()
tfidf_scores = np.mean(tfidf_matrix.toarray(), axis=0)

# Combine TF-IDF scores with lambda-weighted words
lambda_weight = 0.3  # Adjust weight for lambda filtering
tfidf_dict = {feature_names[i]: tfidf_scores[i] for i in range(len(feature_names))}
```

```
# Re-rank words based on (TF-IDF * Lambda weight)
final_keywords = sorted(tfidf_dict.items(), key=lambda x: x[1] * lambda_weight, reverse=True)[:10]
print(final_keywords)
```

```
[('good', 0.10260055046245263), ('work', 0.06731039869194158), ('company', 0.0550961353339056), ('culture', 0.037646
44610282085), ('job', 0.03261712724540863), ('samsung', 0.029223791642726718), ('security', 0.027642512492651473),
('life', 0.025739323057678332), ('working', 0.023809930124064963), ('environment', 0.023301622079004717)]
```

Using Lambda values to Analyse important words

In [8]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import pandas as pd

# Load and preprocess data
df["Likes"] = df["Likes"].fillna("").astype(str)
df["Dislikes"] = df["Dislikes"].fillna("").astype(str)

# Use a single vectorizer for both Likes and Dislikes
vectorizer = TfidfVectorizer(stop_words="english", max_df=0.95, min_df=2)
X_all = vectorizer.fit_transform(df["Likes"] + " " + df["Dislikes"])

# Apply same vocabulary for both Likes & Dislikes
X_likes = vectorizer.transform(df["Likes"])
X_dislikes = vectorizer.transform(df["Dislikes"])

# Number of topics
n_topics = 3

# Train LDA models
lda_likes = LatentDirichletAllocation(n_components=n_topics, random_state=42)
lda_dislikes = LatentDirichletAllocation(n_components=n_topics, random_state=42)

lda_likes.fit(X_likes)
lda_dislikes.fit(X_dislikes)
```

Out[8]:

```
▾ LatentDirichletAllocation
LatentDirichletAllocation(n_components=3, random_state=42)
```

In [10]:

```python
import numpy as np

# Function to compute lambda scores
def get_lambda_scores(model, feature_names):
    topic_word_distributions = model.components_
    lambda_scores = topic_word_distributions / topic_word_distributions.sum(axis=1, keepdims=True)

    # Store lambda scores in a dictionary
    lambda_dict = {}
    for topic_idx, topic in enumerate(lambda_scores):
        lambda_dict[f"Topic {topic_idx + 1}"] = {
            feature_names[i]: topic[i] for i in topic.argsort()[:-11:-1]
        }
    return lambda_dict

# Compute lambda scores for Likes
lambda_scores_likes = get_lambda_scores(lda_likes, vectorizer.get_feature_names_out())

# Compute lambda scores for Dislikes
lambda_scores_dislikes = get_lambda_scores(lda_dislikes, vectorizer.get_feature_names_out())

# Print results
print("\n Lambda Scores for 'Likes' Topics:")
for topic, words in lambda_scores_likes.items():
    print(f"{topic}: {words}\n")

print("\n Lambda Scores for 'Dislikes' Topics:")
for topic, words in lambda_scores_dislikes.items():
    print(f"{topic}: {words}\n")
```

```
 Lambda Scores for 'Likes' Topics:
Topic 1: {'samsung': 0.04635132936822327, 'experience': 0.02732959903468255, 'like': 0.02421415659800387 6, 'excellen
t': 0.021786955010323104, 'brand': 0.021767157782129426, 'compensation': 0.019590257995698072, 'company': 0.01911569
7332930465, 'great': 0.013850485015676954, 'organization': 0.011855305261451263, 'india': 0.011754321410815564}

Topic 2: {'good': 0.10758035470556034, 'environment': 0.030966885764792788, 'salary': 0.028953040187784572, 'growt
h': 0.024754401346325783, 'working': 0.022194877733088484, 'company': 0.02097354334911493, 'overall': 0.015996666048
17473, 'career': 0.015741372801603478, 'benefits': 0.014474161968618603, 'policy': 0.013831385995240 98}

Topic 3: {'work': 0.06522665020172312, 'job': 0.03519722701205212, 'culture': 0.03273131525812354, 'company': 0.0315
9977452945274, 'security': 0.029542938128901686, 'life': 0.026857095325335034, 'balance': 0.02358937698197495, 'goo
d': 0.023567360078697214, 'management': 0.02189346407569115, 'policies': 0.016660391826459092}


 Lambda Scores for 'Dislikes' Topics:
Topic 1: {'samsung': 0.030882392602946528, 'like': 0.02086747732372316, 'politics': 0.0174480386261473, 'compensatio
n': 0.014518992857351592, 'working': 0.014356849275701977, 'company': 0.014156779765943862, 'experience': 0.01332324
269069077, 'brand': 0.013008154568714821, 'employee': 0.012828921391216537, 'opportunity': 0.010031747409844012}

Topic 2: {'good': 0.07655741096308727, 'growth': 0.034491643704081504, 'salary': 0.032572222633307606, 'working': 0.
020789350452099516, 'career': 0.019864922907127255, 'appraisal': 0.017225721298129566, 'promotion': 0.01514086697861
4784, 'company': 0.012545479108067803, 'policy': 0.012449833726451812, 'people': 0.011962876080297452}

Topic 3: {'work': 0.05592932742932628, 'job': 0.038087600113823825, 'culture': 0.036537921101209084, 'security': 0.0
29943344150783823, 'dislike': 0.026233740618030073, 'life': 0.02593474516654007, 'balance': 0.024186041644775728, 'm
anagement': 0.01751003916945211, 'company': 0.017481436357454916, 'pressure': 0.014814585699660302}
```

Topic 1: Brand Reputation & Experience Words like "Samsung", "experience", "brand", "company", and "organization" indicate that people appreciate the brand reputation and their experience with it. "Excellent", "great", and "compensation" suggest positive sentiments towards salary and workplace experience.

Inference: Employees value Samsung's reputation, compensation, and work experience.

Topic 2: Work Environment & Career Growth Words like "good", "environment", "salary", "growth", "career", "benefits", and "policy" indicate that people appreciate the overall work environment, career growth, and company policies.

Inference: Employees appreciate the positive work culture, salary structure, and career growth opportunities.

Topic 3: Work-Life Balance & Job Security Keywords "work", "job", "culture", "security", "life", "balance", and "management" highlight themes related to job security and work-life balance. "Good" and "policies" suggest appreciation for management policies.

Inference: Employees value job security, work-life balance, and management policies.

Dislikes Topics Analysis The topics derived from negative feedback (Dislikes) reveal key areas of concern:

Topic 1: Politics & Limited Opportunities Words like "politics", "compensation", "experience", "opportunity", and "employee" suggest dissatisfaction with workplace politics and lack of growth opportunities. "Samsung" appearing here indicates that some employees have negative experiences with the company.

Inference: Employees are unhappy with office politics, limited career opportunities, and fairness in compensation.

Topic 2: Salary & Growth Concerns "Growth", "salary", "working", "career", "appraisal", "promotion", and "policy" suggest dissatisfaction with salary, career progression, and promotion policies.

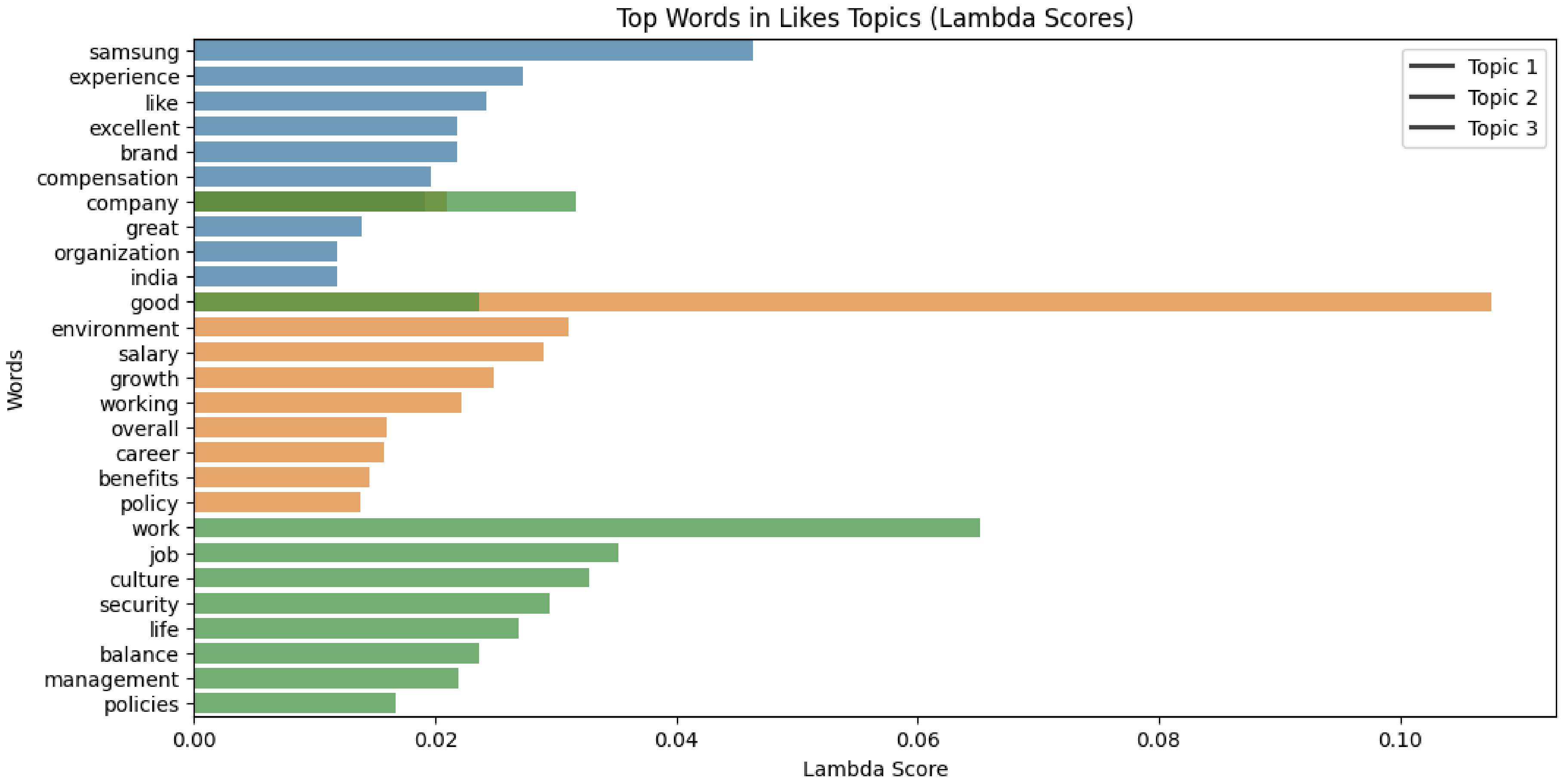
Inference: Employees feel underpaid and lack growth and promotion opportunities.

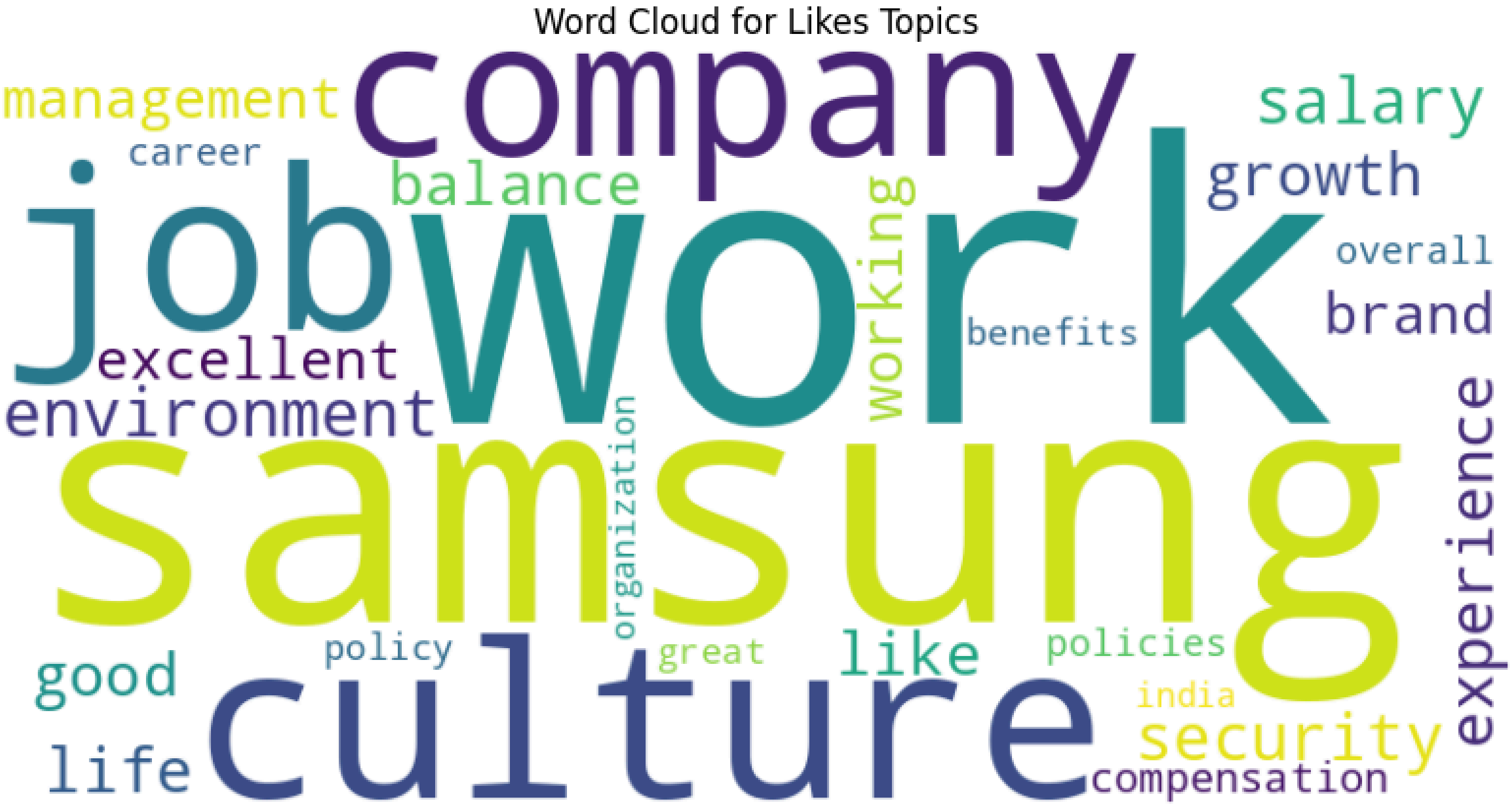
Topic 3: Work-Life Balance & Job Pressure "Work", "job", "culture", "security", "dislike", "life", "balance", "pressure", and "management" suggest stress due to workload, poor management, and job insecurity.

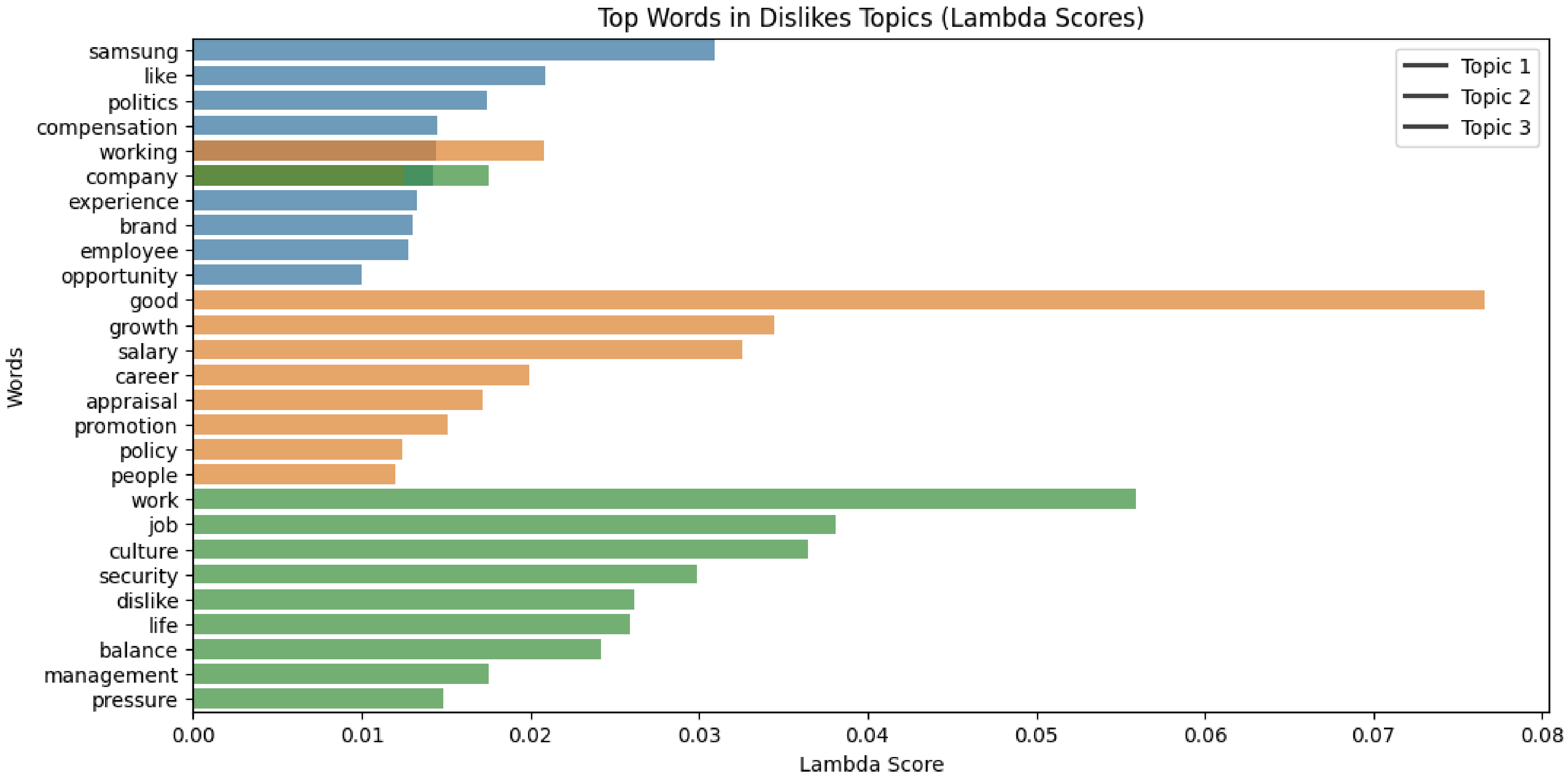
Inference: Employees face high work pressure, poor management, and concerns over job security.

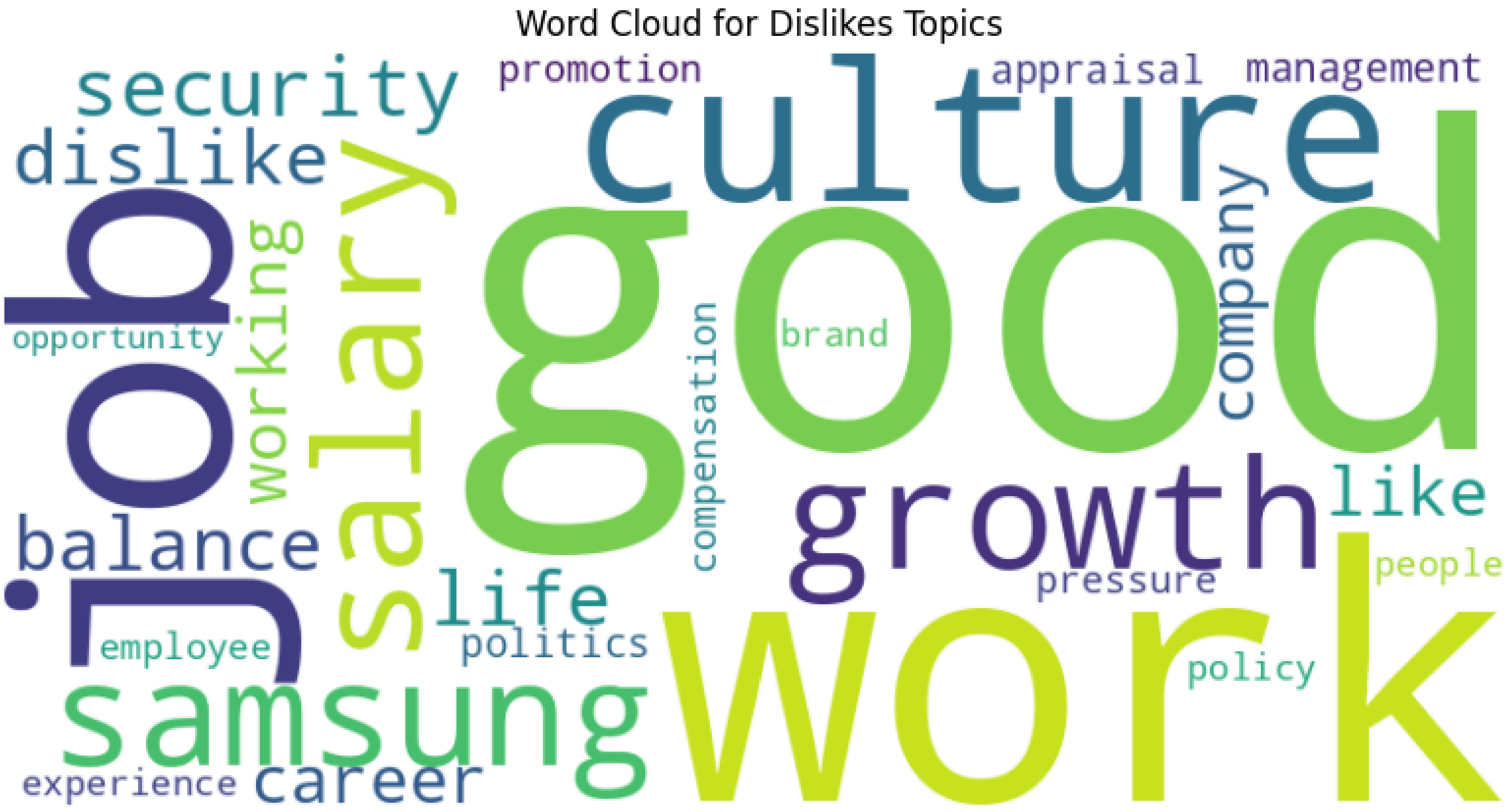
In [11]:

```python
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud

# Lambda scores for Likes Topics
lambda_scores_likes = {
    "Topic 1": {'samsung': 0.0463, 'experience': 0.0273, 'like': 0.0242, 'excellent': 0.0218, 'brand': 0.0218, 'com
    "Topic 2": {'good': 0.1076, 'environment': 0.0310, 'salary': 0.0290, 'growth': 0.0248, 'working': 0.0222, 'comp
    "Topic 3": {'work': 0.0652, 'job': 0.0352, 'culture': 0.0327, 'company': 0.0316, 'security': 0.0295, 'life': 0.
}

# Lambda scores for Dislikes Topics
lambda_scores_dislikes = {
    "Topic 1": {'samsung': 0.0309, 'like': 0.0209, 'politics': 0.0174, 'compensation': 0.0145, 'working': 0.0144, '
    "Topic 2": {'good': 0.0766, 'growth': 0.0345, 'salary': 0.0326, 'working': 0.0208, 'career': 0.0199, 'appraisal
    "Topic 3": {'work': 0.0559, 'job': 0.0381, 'culture': 0.0365, 'security': 0.0299, 'dislike': 0.0262, 'life': 0.
}

# Function to plot bar charts
def plot_bar_chart(lambda_scores, title):
    plt.figure(figsize=(12, 6))
    for topic, words in lambda_scores.items():
        words_sorted = sorted(words.items(), key=lambda x: x[1], reverse=True)
        words_list, scores = zip(*words_sorted)

        sns.barplot(x=scores, y=words_list, label=topic, alpha=0.7)

    plt.xlabel("Lambda Score")
    plt.ylabel("Words")
    plt.title(title)
    plt.legend(lambda_scores.keys())
    plt.show()

# Function to generate word clouds
def plot_wordcloud(lambda_scores, title):
    word_freq = {word: score for topic in lambda_scores.values() for word, score in topic.items()}
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(word_freq)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title(title)
    plt.show()

# Visualizing Likes Topics
plot_bar_chart(lambda_scores_likes, "Top Words in Likes Topics (Lambda Scores)")
plot_wordcloud(lambda_scores_likes, "Word Cloud for Likes Topics")

# Visualizing Dislikes Topics
plot_bar_chart(lambda_scores_dislikes, "Top Words in Dislikes Topics (Lambda Scores)")
plot_wordcloud(lambda_scores_dislikes, "Word Cloud for Dislikes Topics")
```


Top Words in Likes Topics (Lambda Scores)
Words
samsung
experience
like
excellent
brand
compensation
company
great
organization
india
good
environment
salary
growth
working
overall
career
benefits
policy
work
job
culture
security
life
balance
management
policies
Topic 1
Topic 2
Topic 3
0.00
0.02
0.04
0.06
0.08
0.10
Lambda Score


Word Cloud for Likes Topics
work
samsung
company
job
culture
management
salary
career
balance
growth
working
overall
brand
benefits
excellent
environment
experience
organization
good
policy
great
like
policies
india
security
life
compensation


Top Words in Dislikes Topics (Lambda Scores)
Topic 1
Topic 2
Topic 3
samsung
like
politics
compensation
working
company
experience
brand
employee
opportunity
good
growth
salary
career
appraisal
promotion
policy
people
work
job
culture
security
dislike
life
balance
management
pressure
Words
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Lambda Score



Likes Topics: Topic 1: Strong Brand Recognition & Experience

The word "Samsung" has the highest lambda score, indicating that employees strongly associate the brand name with positive experiences. Other words like "experience," "excellent," "brand," and "compensation" suggest that employees appreciate their experience working at Samsung, along with its reputation and salary benefits. "Organization" and "India" hint at regional or company structure-related aspects being discussed positively.

Topic 2: Work Environment & Growth Opportunities

"Good" has the highest score, reflecting general satisfaction. "Environment," "salary," "growth," and "working" indicate that employees value the work culture, financial compensation, and career advancement opportunities. "Career," "benefits," and "policy" suggest that HR policies and job security are appreciated.

Topic 3: Work Culture & Job Security

"Work" dominates, indicating a strong emphasis on job-related aspects. "Job," "culture," "company," and "security" suggest employees perceive their workplace culture and job security as positive. "Life balance" being present shows that employees also recognize work-life balance as a positive factor.

Dislikes Topics: Topic 1: Internal Politics & Compensation Issues

"Samsung" still appears, but in a negative context. "Politics" is a major concern, highlighting workplace politics and favoritism. "Compensation," "working," and "company" indicate dissatisfaction with salary or company practices. "Employee" and "opportunity" being present suggest grievances related to employee treatment and career advancement.

Topic 2: Career Growth & Appraisals

"Good" is present but in a different context (possibly sarcasm or mixed reviews). "Growth," "salary," "appraisal," and "promotion" highlight concerns about promotions and salary hikes. "Policy" and "people" suggest HR policies and interpersonal relations as areas of concern.

Topic 3: Work Pressure & Job Security Issues

"Work," "job," "culture," and "security" again appear, but in a negative sense. "Dislike," "pressure," and "management" suggest employees face high-pressure environments and management issues. "Life balance" being mentioned here indicates dissatisfaction with work-life balance

Key Takeaways: Positive Factors: Samsung’s brand reputation, work culture, salary, career growth, and policies are well-received.

Negative Factors: Internal politics, slow career progression, work pressure, and dissatisfaction with salary appraisals need attention.

Actionable Insights:

Improve Appraisals & Promotions: Address concerns around salary hikes and career advancement.

Reduce Workplace Politics: Implement fair policies and reduce favoritism.

Enhance Work-Life Balance: Address complaints about excessive work pressure.

Basic Operations of NLP

In [1]:

```
import numpy
```

In [2]:

```
print(numpy.__version__)
```

```
1.26.4
```

In [3]:

```
import spacy
```

In [4]:

```
print(spacy.__version__)
```

```
3.5.3
```

In [5]:

```
nlp=spacy.load('en_core_web_sm')
```

In [6]:

```
type(nlp)
```

Out[6]:

```
spacy.lang.en.English
```

In [9]:

```
import pandas as pd
```

In [11]:

```
df=pd.read_csv('Samsung.csv')
df.head(5)
```

Out[11]:

	Title	Place	Job_type	Department	Date	Overall_rating	work_life_balance	skill_development	salary_and_benefits
0	Professional Logistics	Chennai	Full Time	SCM & Logistics Department	5 Sep 2023	Average	Average	Poor	Average
1	Supervisor Instructor	Noida	Full Time	Production & Manufacturing Department	1 Sep 2023	Excellent	Excellent	Excellent	Excellent
2	Quality Inspector	Noida	Full Time	Quality Assurance and Testing Department	3 Aug 2023	Good	Good	Good	Good
3	Lead Engineer	Noida	Full Time	Software Development Department	6 Aug 2023	Good	Average	Good	Excellent
4	Zonal Sales Manager	Chennai	Full Time	Retail & B2C Sales Department	1 Aug 2023	Good	Below Average	Average	Average

In [15]:

```python
df.shape
```

Out[15]:

```
(1211, 15)
```

In [14]:

```python
df["Tokenized_Likes"] = df["Likes"].astype(str).apply(lambda text: [token.text for token in nlp(text)])

print(df[["Likes", "Tokenized_Likes"]].head(10))
```

```
                                               Likes  \
0  Company provide free of cost transportation\nG...   
1  Samsung India electronics company bhut acchi h...   
2  Everything in Samsung India Pvt Ltd is done in...   
3                  There is a lot of scope to learn.   
4  Wonderful data and tracking mechanism. Innovat...   
5                                     Salary is good   
6  Good Infrastructure, Great Global Brand, Capex...   
7  job security, very driven company, management ...   
8  It's a very good organization. We can enjoy th...   
9  Good company to work with.Have spent a decade ...   

                                     Tokenized_Likes  
0  [Company, provide, free, of, cost, transportat...  
1  [Samsung, India, electronics, company, bhut, a...  
2  [Everything, in, Samsung, India, Pvt, Ltd, is,...  
3       [There, is, a, lot, of, scope, to, learn, .]  
4  [Wonderful, data, and, tracking, mechanism, .,...  
5                                 [Salary, is, good]  
6  [Good, Infrastructure, ,, Great, Global, Brand...  
7  [job, security, ,, very, driven, company, ,, m...  
8  [It, 's, a, very, good, organization, ., We, c...  
9  [Good, company, to, work, with, ., Have, spent...  
```

In [16]:

```
df.head()
```

Out[16]:

	Title	Place	Job_type	Department	Date	Overall_rating	work_life_balance	skill_development	salary_and_benefits
0	Professional Logistics	Chennai	Full Time	SCM & Logistics Department	5 Sep 2023	Average	Average	Poor	Average
1	Supervisor Instructor	Noida	Full Time	Production & Manufacturing Department	1 Sep 2023	Excellent	Excellent	Excellent	Excellent
2	Quality Inspector	Noida	Full Time	Quality Assurance and Testing Department	3 Aug 2023	Good	Good	Good	Good
3	Lead Engineer	Noida	Full Time	Software Development Department	6 Aug 2023	Good	Average	Good	Excellent
4	Zonal Sales Manager	Chennai	Full Time	Retail & B2C Sales Department	1 Aug 2023	Good	Below Average	Average	Average

In [17]:

```
df["Tokenized_Dislikes"] = df["Dislikes"].astype(str).apply(lambda text: [token.text for token in nlp(text)])

print(df[["Dislikes", "Tokenized_Dislikes"]].head(10))
```

```
                                            Dislikes  \
0  Montonous work\nManager don't assign responsib...   
1  Koi bhi problem nhi h company ki \nSab kuchh a...   
2  Samsung private limited I eat discipline with ...   
3  The rating system is not transparent. Maintain...   
4        At times we don't accept the market reality   
5  Don't join if you have another opportunity. \n...   
6                                 Work Life Balance.   
7  got vitamin deficiency due to lack of sun, 8 h...   
8  Growth will come slowly.. but u can gain more ...   
9                            Appraisal and promotion   

                                   Tokenized_Dislikes  
0  [Montonous, work, \n, Manager, do, n't, assign...  
1  [Koi, bhi, problem, nhi, h, company, ki, \n, S...  
2  [Samsung, private, limited, I, eat, discipline...  
3  [The, rating, system, is, not, transparent, .,...  
4  [At, times, we, do, n't, accept, the, market, ...  
5  [Do, n't, join, if, you, have, another, opport...  
6                           [Work, Life, Balance, .]  
7  [got, vitamin, deficiency, due, to, lack, of, ...  
8  [Growth, will, come, slowly, .., but, u, can, ...  
9                        [Appraisal, and, promotion]  
```

In [18]:

```
df.head(5)
```

Out[18]:

	Title	Place	Job_type	Department	Date	Overall_rating	work_life_balance	skill_development	salary_and_benefits
0	Professional Logistics	Chennai	Full Time	SCM & Logistics Department	5 Sep 2023	Average	Average	Poor	Average
1	Supervisor Instructor	Noida	Full Time	Production & Manufacturing Department	1 Sep 2023	Excellent	Excellent	Excellent	Excellent
2	Quality Inspector	Noida	Full Time	Quality Assurance and Testing Department	3 Aug 2023	Good	Good	Good	Good
3	Lead Engineer	Noida	Full Time	Software Development Department	6 Aug 2023	Good	Average	Good	Excellent
4	Zonal Sales Manager	Chennai	Full Time	Retail & B2C Sales Department	1 Aug 2023	Good	Below Average	Average	Average

In [23]:

```
df.to_csv("New.csv", index=False)
```

In [19]:

```
from spacy.lang.en.stop_words import STOP_WORDS
```

In [1]:

```
#print(STOP_WORDS)
```

In [21]:

```
len(STOP_WORDS)
```

Out[21]:

```
326
```

In [ ]:

```
#def print_tokens_with_stopwords(text):
#    doc = nlp(str(text))  # Convert to string and process with spaCy
#    for token in doc:
#        print(token.text, "=>", token.is_stop)

# Apply function to the 'Likes' column
#df['Likes'].apply(print_tokens_with_stopwords)
```

In [4]:

```
# Function to extract non-stop words
#def extract_non_stop_words(text):
#    doc = nlp(text)  # Process text using spaCy
#    non_stop_words = [token.text for token in doc if not token.is_stop]  # Filter non-stop words

#    print('The list of non-stop words:', non_stop_words)
#    print('The no of non-stop words is:', len(non_stop_words))

# Apply function to each row in the "Likes" column
#df['Likes'].apply(extract_non_stop_words)
```

In [6]:

```
#def print_tokens_with_punctuations(text):
#    doc = nlp(str(text))  # Convert to string and process with spaCy
#    for token in doc:
#        print(token.text, "=>", token.is_punct)

# Apply function to the 'Likes' column
#df['Likes'].apply(print_tokens_with_punctuations)
```

In [8]:

```
#def print_tokens_with_num(text):
#    doc = nlp(str(text))  # Convert to string and process with spaCy
#    for token in doc:
#        print(token.text, "=>", token.like_num)

# Apply function to the 'Likes' column
#df['Dislikes'].apply(print_tokens_with_num)
```

In [10]:

```
#def print_tokens_with_pos(text):
#    doc = nlp(str(text))  # Convert to string and process with spaCy
#    for token in doc:
 #       print(token.text, "=>", token.pos_)
# Apply function to the 'Likes' column
#df['Dislikes'].apply(print_tokens_with_pos)
```

In [12]:

```
#from collections import Counter
#if 'Tokenized_Likes' in df.columns:
#    # Process the text and extract POS tags
#    pos_counts = Counter()

#    for text in df['Tokenized_Likes'].dropna():
#        doc = nlp(str(text))  # Convert to string to avoid errors
#        pos_counts.update([token.pos_ for token in doc])

    # Convert to DataFrame for better visualization
#    pos_df = pd.DataFrame(pos_counts.items(), columns=['POS', 'Count']).sort_values(by='Count', ascending=False)
#    print(pos_df)
#else:
#    print("Column 'tokenised_likes' not found in the dataset.")
```

In [14]:

```
#import matplotlib.pyplot as plt
#import seaborn as sns

# Plot bar chart
#plt.figure(figsize=(6, 6))
#sns.barplot(x=pos_df['POS'], y=pos_df['Count'], palette="viridis")

# Customize the plot
#plt.xlabel("Part of Speech (POS)")
#plt.ylabel("Count")
#plt.title("POS Tag Counts in tokenised_likes Column")
#plt.xticks(rotation=45)
#plt.grid(axis="y", linestyle="--", alpha=0.7)

# Show the plot
#plt.show()
```

In [42]:

```
text_df=pd.DataFrame(df, columns=['Likes'])
text_df
```

Out[42]:

	Likes
0	Company provide free of cost transportation\nG...
1	Samsung India electronics company bhut acchi h...
2	Everything in Samsung India Pvt Ltd is done in...
3	There is a lot of scope to learn.
4	Wonderful data and tracking mechanism. Innovat...
...	...
1206	work-life balance is good.\nthe work environme...
1207	Learning experience and benefits are good
1208	Best in the Industry
1209	I love the fact that I am a part of a team tha...
1210	Yes

1211 rows × 1 columns

In [46]:

```python
from spacy import displacy
```

In [50]:

```python
if 'Likes' in df.columns:
    text_df = pd.DataFrame(df, columns=['Likes'])

    # Create a list to store entity details
    ent_cols = ['ENTITY', 'ENTITY TYPE', 'ENTITY EXPLANATION']
    ent_rows = []

    # Process and visualize NER for the first few rows
    for text in text_df['Likes'].dropna().head(5):  # Adjust the number of rows as needed
        doc = nlp(str(text))  # Convert to string to avoid errors

        # Render Named Entity Recognition (NER) visualization
        displacy.render(doc, style='ent')
        # Extract named entities and store them in a DataFrame
        for ent in doc.ents:
            row = (ent.text, ent.label_, spacy.explain(ent.label_))
            ent_rows.append(row)

    # Convert extracted entities into a DataFrame
    ent_df = pd.DataFrame(ent_rows, columns=ent_cols)

    # Display entity DataFrame
    print(ent_df)

else:
    print("Column 'Likes' not found in the dataset.")
```

```
/opt/anaconda3/envs/Sayan_NLP/lib/python3.10/site-packages/spacy/displacy/__init__.py:215: UserWarning: [W006] No en
tities to visualize found in Doc object. If this is surprising to you, make sure the Doc was processed using a model
that supports named entity recognition, and check the `doc.ents` property manually if necessary.
  warnings.warn(Warnings.W006)
```

Company provide free of cost transportationGood medical insurance policies including parents

Samsung India electronics company bhut acchi h ORG Maine GPE usme 1 year DATE experience bhi kiya h

Everything in Samsung India Pvt Ltd ORG is done in good quality and at a critical level

There is a lot of scope to learn.

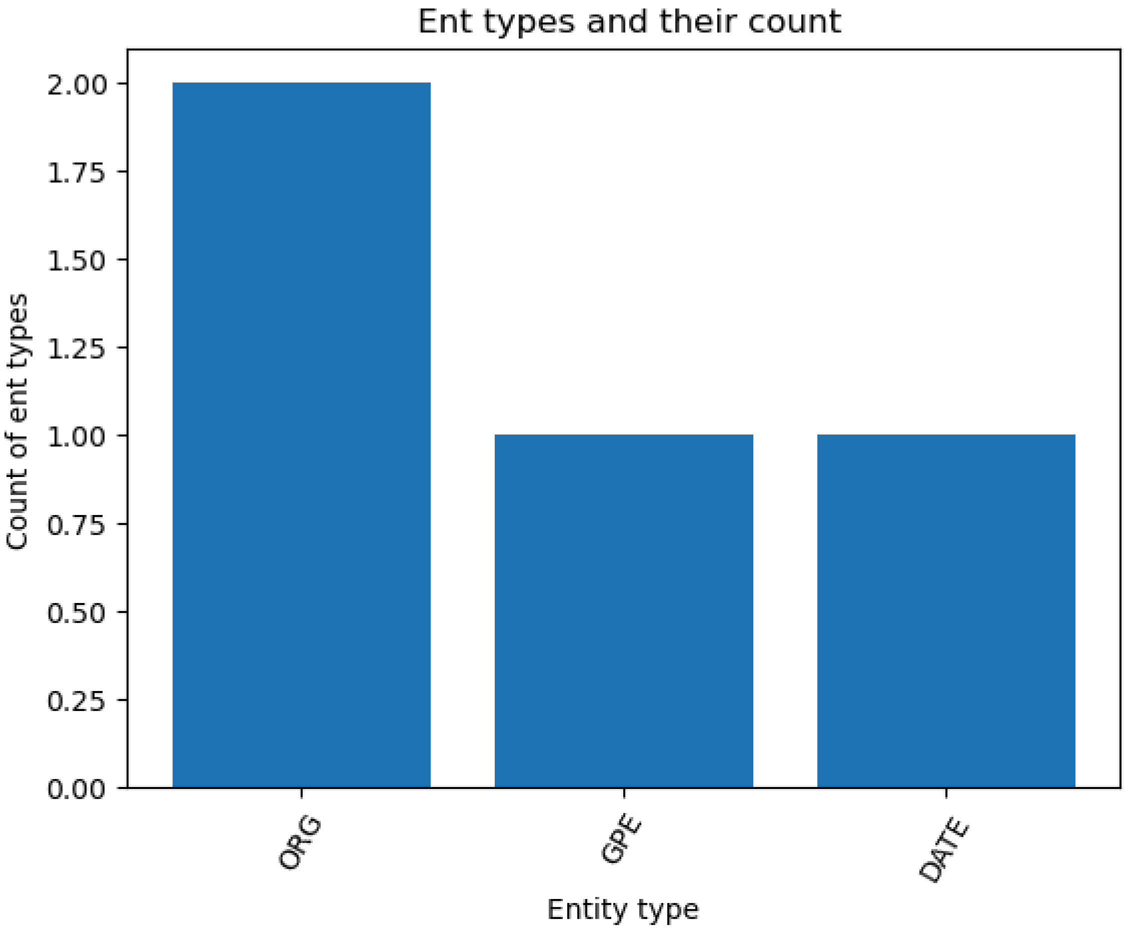
Wonderful data and tracking mechanism. Innovation at its best

```
                                              ENTITY ENTITY TYPE  \
0  Samsung India electronics company bhut acchi h \n         ORG   
1                                              Maine         GPE   
2                                             1 year        DATE   
3                              Samsung India Pvt Ltd         ORG   

                        ENTITY EXPLANATION  
0  Companies, agencies, institutions, etc.  
1                Countries, cities, states  
2    Absolute or relative dates or periods  
3  Companies, agencies, institutions, etc.  
```

```
In [51]: spacy.explain('NORP')
```

```
Out[51]: 'Nationalities or religious or political groups'
```

```
In [52]: plt.bar(ent_df['ENTITY TYPE'].value_counts().index,ent_df['ENTITY TYPE'].value_counts())
         plt.xlabel('Entity type')
         plt.ylabel('Count of ent types')
         plt.title(' Ent types and their count')
         plt.xticks(rotation=60);
```



In [16]:
```
#if 'Dislikes' in df.columns:
    # Concatenate all text in the 'Likes' column into a single string
#    dislikes_text = " ".join(df['Dislikes'].dropna().astype(str))

    # Convert the entire text into a spaCy Doc object
 #   doc_dislikes = nlp(dislikes_text)

    # Now, doc_likes can be used like doc1
 #   print(doc_dislikes)
#else:
#    print("Column 'Likes' not found in the dataset.")
```

In [18]:
```
#if 'Likes' in df.columns:
    # Concatenate all text in the 'Likes' column into a single string
#    Likes_text = " ".join(df['Likes'].dropna().astype(str))

    # Convert the entire text into a spaCy Doc object
#    doc_likes = nlp(Likes_text)

    # Now, doc_likes can be used like doc1
#    print(doc_likes)
#else:
#    print("Column 'Likes' not found in the dataset.")
```

In [56]:
```
docsList=[doc_dislikes,doc_likes]
```

In [60]:
```
fullTokens = []  # A list of all tokens in the full set

for document in docsList:
    doc = nlp(document)
    docTokens = []  # A list of tokens for each doc

    for token in doc:
        # Remove stopwords, punctuation, numbers, and empty/newline tokens
        if not token.is_stop and not token.is_punct and not token.like_num and token.text.strip() and not token.tex
            docTokens.append(token.lemma_)

    fullTokens.append(docTokens)
```

In [20]:
```
#print(fullTokens)
```

In [62]:
```
from gensim.corpora import  Dictionary

tokenDict=Dictionary(fullTokens)
print(tokenDict)
```

```
Dictionary<2079 unique tokens: ['-Abhik', '-favouritism', '-military', '-promotion', '-too']...>
```

In [22]:
```
#print(tokenDict.token2id)
```

In [24]:
```
bows=[]
for token in fullTokens:
    bows.append(tokenDict.doc2bow(token))

#print(bows)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[24], line 2
      1 bows=[]
----> 2 for token in fullTokens:
      3     bows.append(tokenDict.doc2bow(token))

NameError: name 'fullTokens' is not defined
```

In [65]:
```
from gensim.models import TfidfModel
tfidf=TfidfModel(bows)
print(tfidf)
```

```
TfidfModel<num_docs=2, num_nnz=2653>
```

In [26]:
```
tfidf_vec=[]
for word in bows:
    tfidf_vec.append(tfidf[word])
#print(tfidf_vec)
```

In [67]:
```
from gensim.similarities import MatrixSimilarity
```

```
sim=MatrixSimilarity(tfidf_vec,num_features=len(tokenDict))
print(sim)
```

```
MatrixSimilarity<2 docs, 2079 features>
```

In [68]:

```
print(sim[tfidf_vec[0]])
```

```
[1.0000002 0.       ]
```

In [69]:

```
print(sim[tfidf_vec[1]])
```

```
[0.        0.9999998]
```

In [ ]: