```
import os
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Download NLTK resources
nltk.download('vader_lexicon')

# Path to the directory containing news articles
news_directory = 'E:\\SIH\\news_articles'

# Define keywords associated with different departments
department_keywords = {
    'Health': ['health', 'medical', 'hospital', 'disease'],
    'Education': ['education', 'school', 'student', 'learning'],
    'Finance': ['finance', 'economy', 'budget', 'tax'],
    # Add more departments and associated keywords
}

# Function to load data from a directory
def load_data_from_directory(directory):
    data = []
    for filename in os.listdir(directory):
        if filename.endswith('.txt'):
            with open(os.path.join(directory, filename), 'r', encoding='utf-8') as file:
                text = file.read()
                data.append((filename, text))  # Store file name along with content
    return data

# Load news articles from the directory
news_articles = load_data_from_directory(news_directory)

# Initialize VADER sentiment analyzer
sia = SentimentIntensityAnalyzer()

# Initialize counters and department counters
positive_count = 0
negative_count = 0
neutral_count = 0
department_counts = {department: 0 for department in department_keywords}

# Classify each news article and count sentiments and departments
for idx, (filename, article) in enumerate(news_articles, start=1):
    sentiment_scores = sia.polarity_scores(article)

    # Classify sentiment
    if sentiment_scores['compound'] >= 0.05:
        sentiment = 'positive'
        positive_count += 1
    elif sentiment_scores['compound'] <= -0.05:
        sentiment = 'negative'
        negative_count += 1
    else:
        sentiment = 'neutral'
        neutral_count += 1
```

```
    # Categorize into departments
    article_lower = article.lower()
    department_type = None  # Initialize department type
    for department, keywords in department_keywords.items():
        if any(keyword in article_lower for keyword in keywords):
            department_counts[department] += 1
            department_type = department  # Assign department type

    print(f"Article {idx}: {sentiment} - Department: {department_type or 'Unknown'} - Text: {article[:50]}...")
indian goverment betrays farmers

total_articles = len(news_articles)
positive_percentage = (positive_count / total_articles) * 100
negative_percentage = (negative_count / total_articles) * 100
neutral_percentage = (neutral_count / total_articles) * 100

print("\nSentiment Distribution:")
print(f"Positive: {positive_percentage:.2f}%")
print(f"Negative: {negative_percentage:.2f}%")
print(f"Neutral: {neutral_percentage:.2f}%")
india loses in world cup
print("\nDepartment Distribution:")
for department, count in department_counts.items():
    print(f"{department}: {count}")india government betrays farmers
```