Handout 1: Introduction to Sentiment Analysis (Basic Level)

Machine Learning for Smarter Innovation

1 Handout 1: Introduction to Sentiment Analysis (Basic Level)

1.1 Week 3 - NLP for Emotional Context

1.1.1 What is Sentiment Analysis?

Sentiment analysis is the computational study of people's opinions, sentiments, emotions, and attitudes expressed in text. It helps us understand how users feel about products, services, or experiences at scale.

1.1.2 Key Concepts

1. Sentiment Polarity

- **Positive**: Expressing satisfaction, happiness, or approval
- **Negative**: Expressing dissatisfaction, frustration, or disapproval

- **Neutral**: Factual statements without clear emotion

2. Why Sentiment Analysis Matters for Design

- Understand user pain points automatically
- Identify delightful moments in the user journey
- Prioritize features based on emotional impact
- Monitor brand perception in real-time

1.1.3 Traditional vs Machine Learning Approaches

Rule-Based Methods

```
# Simple rule-based sentiment using word lists
positive_words = ['good', 'great', 'excellent', 'amazing', 'wonderful']
negative_words = ['bad', 'terrible', 'awful', 'horrible', 'poor']

def simple_sentiment(text):
    text_lower = text.lower()
    pos_count = sum(1 for word in positive_words if word in text_lower)
    neg_count = sum(1 for word in negative_words if word in text_lower)

    if pos_count > neg_count:
        return 'positive'
    elif neg_count > pos_count:
        return 'negative'
    else:
        return 'neutral'
```

Limitations: - Misses context ("not good" would be positive) - Can't detect sarcasm - Limited vocabulary - No understanding of word relationships

Machine Learning Approach

```python
from textblob import TextBlob

def ml_sentiment(text):
    analysis = TextBlob(text)
    # Polarity ranges from -1 (negative) to 1 (positive)
    if analysis.sentiment.polarity > 0.1:
        return 'positive'
    elif analysis.sentiment.polarity < -0.1:
        return 'negative'
    else:
        return 'neutral'
```

1.1.4 Getting Started with TextBlob

Installation

```
pip install textblob
python -m textblob.download_corpora
```

Basic Usage

```python
from textblob import TextBlob

# Analyze a product review
review = "This product is amazing! The quality exceeded my expectations."
blob = TextBlob(review)

# Get sentiment
print(f"Polarity: {blob.sentiment.polarity}")  # 0.625 (positive)
print(f"Subjectivity: {blob.sentiment.subjectivity}")  # 0.6

# Polarity: -1 to 1 (negative to positive)
# Subjectivity: 0 to 1 (objective to subjective)
```

1.1.5 Practice Exercise: Movie Review Sentiment

Task 1: Basic Analysis

```python
reviews = [
    "This movie was fantastic! Best film of the year.",
    "Boring and predictable. Waste of time.",
    "The acting was okay but the plot was confusing.",
    "Absolutely loved it! Can't wait to watch again.",
    "Terrible movie. Do not recommend."
]

# Your task: Classify each review as positive, negative, or neutral
# Use both rule-based and TextBlob methods
# Compare the results
```

Task 2: Batch Processing

```python
def analyze_reviews(reviews):
    results = []
    for review in reviews:
        blob = TextBlob(review)
        sentiment = 'positive' if blob.sentiment.polarity > 0.1 else \
                   'negative' if blob.sentiment.polarity < -0.1 else 'neutral'
        results.append({
            'text': review,
            'sentiment': sentiment,
            'confidence': abs(blob.sentiment.polarity)
        })
    return results

# Analyze all reviews and find:
# 1. What percentage are positive?
# 2. Which review has the strongest sentiment?
# 3. Which reviews might need human review (low confidence)?
```

1.1.6 Understanding Context Matters

Consider these examples: 1. “This product is sick!” → Context: Teen slang (Positive) vs Medical (Negative) 2. “It’s fine…” → Context: After complaint (Negative) vs First impression (Neutral) 3. “Interesting choice” → Context: Design critique (Negative) vs Genuine interest (Positive)

1.1.7 Common Pitfalls to Avoid

1. **Ignoring Negation**: “not bad” “bad”
2. **Missing Sarcasm**: “Oh great, another bug!” (Actually negative)
3. **Domain Specificity**: “This app is the bomb!” (Positive in casual, concerning in security)
4. **Mixed Sentiment**: “Love the design but hate the price”

1.1.8 Real-World Applications

Customer Reviews

```
# Analyzing product feedback
def analyze_product_reviews(reviews):
    sentiments = {'positive': 0, 'negative': 0, 'neutral': 0}
    issues = []

    for review in reviews:
        blob = TextBlob(review)

        # Classify sentiment
        if blob.sentiment.polarity > 0.1:
            sentiments['positive'] += 1
        elif blob.sentiment.polarity < -0.1:
            sentiments['negative'] += 1
            # Extract potential issues from negative reviews
            if 'bug' in review.lower() or 'crash' in review.lower():
                issues.append('Technical issues')
            elif 'expensive' in review.lower() or 'price' in review.lower():
                issues.append('Pricing concerns')
        else:
            sentiments['neutral'] += 1

    return sentiments, issues
```

1.1.9 Key Takeaways

1. **Start Simple**: Begin with TextBlob for quick prototypes
2. **Context is King**: Always consider the domain and audience

3. **Validate Results**: Manual review of a sample is essential
4. **Iterate**: Improve based on false positives/negatives
5. **Combine Methods**: Use rules for obvious cases, ML for complex ones

1.1.10 Further Learning

1. Try analyzing your favorite product's reviews
2. Experiment with different sentiment thresholds
3. Build a simple sentiment dashboard
4. Explore emotion detection (beyond positive/negative)

1.1.11 Useful Resources

- TextBlob Documentation: https://textblob.readthedocs.io/
- NLTK Sentiment Analysis: https://www.nltk.org/
- Hugging Face Sentiment Models: https://huggingface.co/models?pipeline_tag=sentiment-analysis
- Real-time Twitter Sentiment: https://www.sentiment140.com/

1.1.12 Assignment

Create a simple sentiment analyzer for a product of your choice: 1. Collect 20-50 reviews (from Amazon, App Store, etc.) 2. Analyze sentiment using TextBlob 3. Identify the top 3 positive and negative themes 4. Create a simple visualization of results 5. Suggest one design improvement based on negative sentiment

Deliverable: Python script + 1-page insights report

Remember: Sentiment analysis is a tool to understand users at scale, but always validate insights with real user research!