Assignment: Twitter Text Style Classification

Formal vs Informal English Analysis using NLP

📋 Assignment Overview

Background

Social media platforms like Twitter have revolutionized how people communicate, creating a unique linguistic environment where **informal language** thrives alongside **formal expressions**. This linguistic variation, known as **register variation** in sociolinguistics, provides rich data for studying how language adapts to different communicative contexts.

In this assignment, you will build a text classification system that distinguishes between **Formal** and **Informal** English tweets using Natural Language Processing techniques, specifically **Bag-of-Words (BoW)** and **TF-IDF**.

Linguistic Motivation

Register refers to the variety of language used for a particular purpose or in a particular social setting. Key differences include:

Feature	Formal Register	Informal Register
Vocabulary	Standard, precise words	Slang, abbreviations, contractions
Grammar	Complete sentences, proper structure	Fragments, relaxed grammar
Spelling	Standard orthography	Creative spelling, phonetic representations
Punctuation	Conventional usage	Expressive (!!!, ???), minimal, or absent
Tone	Professional, objective	Casual, emotional, personal

Learning Objectives

By completing this assignment, you will:

1. Perform **text preprocessing** on social media data (handling @mentions, #hashtags, URLs, emojis, etc.)
2. Understand and identify **linguistic markers** of formal vs informal language
3. Apply **Bag-of-Words** and **TF-IDF** for feature extraction
4. Build and evaluate **machine learning classifiers** for style detection
5. Analyze which **linguistic features** are most predictive of writing style

---

📦 Dataset

Source

Sentiment140 Dataset - A collection of 1.6 million tweets originally labeled for sentiment analysis.

- **Download**: Kaggle - Sentiment140
- **Original Columns**: `target`, `id`, `date`, `flag`, `user`, `text`

Your Task: Creating Style Labels

The original dataset has **sentiment labels**, but you will create **style labels** based on linguistic features. You will:

1. Sample a subset of tweets (recommended: 5,000-10,000)
2. Create a **rule-based labeling system** to classify tweets as Formal or Informal
3. Use these labels for supervised classification

Labeling Criteria

Informal Indicators (presence suggests Informal):

```
- Abbreviations: u, ur, r, 2, 4, b4, w/, bc, gonna, wanna, gotta
- Repeated characters: loooove, sooo, yesss, nooo
- Emoticons: :), :(, :D, ;), :P, <3, xD
- All lowercase text (no capitalization)
- Multiple punctuation: !!!, ???, ...
- Slang/internet terms: lol, lmao, omg, btw, idk, tbh, ngl
- Hashtag heavy (3+ hashtags)
```

Formal Indicators (presence suggests Formal):

```
- Proper capitalization (sentence case)
- Complete sentences with subject-verb structure
- Standard spelling throughout
- Professional vocabulary
- Proper punctuation (single ., !, ?)
- No emoticons or emoji
- No abbreviations
```

📝 Assignment Structure

Part 1: Data Loading & Exploration (10%)

- Load the Sentiment140 dataset
- Explore basic statistics (tweet length, word count distribution)
- Sample a working subset

Part 2: Text Preprocessing (25%)

Implement preprocessing functions to handle:

- @mentions (e.g., @username)
- #hashtags
- URLs (http, https, www)
- Emoticons and emojis
- Repeated characters (e.g., loooove → love)
- Contractions (optional: expand or keep)
- Case normalization

Part 3: Style Labeling (15%)

- Implement rule-based functions to detect informal markers
- Create binary labels: `Formal` (0) vs `Informal` (1)
- Analyze class distribution
- Validate labeling quality on sample tweets

Part 4: Feature Engineering (20%)

- Implement **Bag-of-Words** representation
- Implement **TF-IDF** representation
- Compare top features from each method
- Visualize feature distributions for each class

Part 5: Model Training & Evaluation (20%)

- Split data into train/test sets
- Train at least 2 classifiers (e.g., Logistic Regression, Naive Bayes, SVM)
- Evaluate using accuracy, precision, recall, F1-score
- Create confusion matrix visualization

Part 6: Linguistic Analysis (10%)

- Identify top predictive features for Formal vs Informal
- Discuss which linguistic markers are most discriminative
- Reflect on the relationship between features and sociolinguistic theory

---

🔧 Technical Requirements

Environment Setup

```
# Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from collections import Counter

# NLP & ML
```

```
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

File Structure

```
assignment/
├── data/
│   └── sentiment140.csv (or your sampled subset)
├── twitter_style_classification.ipynb (main notebook)
├── requirements.txt
└── README.md
```

📊 Evaluation Rubric

Component	Points	Criteria
Data Loading & Exploration	10	Proper loading, meaningful EDA, clear visualizations
Text Preprocessing	25	Complete handling of Twitter-specific elements, clean implementation
Style Labeling	15	Logical rule-based system, balanced classes, validation
Feature Engineering	20	Correct BoW/TF-IDF implementation, insightful comparisons
Model Training & Evaluation	20	Multiple models, proper evaluation metrics, visualization
Linguistic Analysis	10	Thoughtful interpretation, connection to theory

Total: 100 points

💡 Hints & Tips

Preprocessing Order

```
1. Lowercase conversion (optional - may lose information)
2. URL removal
3. @mention handling (remove or replace with token)
4. Hashtag handling (keep word, remove #)
```

```
5. Emoticon/emoji handling
6. Repeated character normalization
7. Punctuation handling
8. Tokenization
```

Labeling Strategy

```python
def calculate_informality_score(text):
    score = 0
    # Check for informal markers
    if re.search(r'\b(u|ur|r|2|4|bc)\b', text.lower()):
        score += 2
    if re.search(r'(.)\1{2,}', text):  # Repeated chars
        score += 2
    if re.search(r'[:;]-?[)D(P]', text):  # Emoticons
        score += 1
    # ... more rules
    return score
```

Useful Regular Expressions

```python
URL_PATTERN = r'https?://\S+|www\.\S+'
MENTION_PATTERN = r'@\w+'
HASHTAG_PATTERN = r'#\w+'
EMOTICON_PATTERN = r'[:;]-?[)D(P]|<3|xD'
REPEATED_CHAR_PATTERN = r'(.)\1{2,}'
```

📚 References

Sociolinguistics & Register

- Biber, D. (1995). *Dimensions of Register Variation*
- Crystal, D. (2011). *Internet Linguistics*
- Herring, S. C. (2012). "Grammar and Electronic Communication"

NLP & Text Classification

- Jurafsky, D. & Martin, J. H. *Speech and Language Processing*
- Manning, C. D. & Schütze, H. *Foundations of Statistical NLP*

Twitter & Social Media Language

- Eisenstein, J. (2013). "What to do about bad language on the internet"
- Gonçalves, B. & Sánchez, D. (2014). "Crowdsourcing dialect characterization through Twitter"

📤 Submission

Submit the following:

1. **Jupyter Notebook** (.ipynb) with all code and outputs
2. **Brief Report** (1-2 pages) discussing:
    - Your preprocessing decisions
    - Labeling methodology and validation
    - Key findings from linguistic analysis
    - Model performance comparison

Deadline: ~12/23(TUE), 11:59

Good luck! Remember, the goal is not just to build a classifier, but to understand the **linguistic patterns** that distinguish formal from informal digital communication.