Comprehensive Data Analysis of Donald Trump's Speech Transcripts

A Complete NLP Pipeline with Predictive Analytics

Course: Database Analytics (DBA)
Semester: 7th Semester

Submitted by:
Team Claude

December 2025

Abstract

This report presents the predictive analytics and behavioral modeling phase of the Trump Speech Analysis project. Building upon the previously submitted data warehousing documentation, this report focuses on the NLP transformation pipeline, feature engineering methodology, descriptive analytics findings, and six predictive models for behavioral analysis. The pipeline processes 43 speech transcripts containing over 320,000 words, with 2,872 unique entities extracted and 80+ engineered features computed. The key contribution is the development of the "Mega Trump Model" — a unified predictive system combining entity reaction profiling, personality compatibility prediction, negotiation success modeling, ML-based response classification, psychological influence strategies, and trigger word detection.

Contents

1 Introduction

1.1 Project Overview

This project implements a complete end-to-end data analytics pipeline for analyzing Donald Trump's speech transcripts. The pipeline encompasses:

1. **Data Collection:** Web scraping from Rev.com transcription service
2. **Data Cleaning:** Text preprocessing and normalization
3. **Data Transformation:** Advanced NLP processing
4. **Feature Engineering:** Deriving analytical features
5. **Data Warehousing:** Star schema design with ETL pipeline
6. **Descriptive Analytics:** Statistical analysis and visualization
7. **Predictive Analytics:** Machine learning and rule-based models

1.2 Objectives

The primary objectives of this analysis are:

- Extract and clean speech transcripts from online sources
- Apply Natural Language Processing (NLP) techniques to extract linguistic features
- Perform sentiment analysis and emotion classification
- Extract named entities (persons, organizations, locations)
- Design and implement a data warehouse using star schema
- Conduct comprehensive descriptive analytics
- Build predictive models for behavioral analysis
- Create an interactive dashboard for model exploration

1.3 Scope of This Report

This report focuses primarily on the **new contributions** beyond the previously submitted data warehousing documentation:

- Detailed NLP transformation methodology
- Feature engineering approaches (80+ derived attributes)
- Descriptive analytics results and findings
- **Six predictive analytics models** (new)
- **Mega Trump Model** – unified behavioral predictor (new)
- Interactive Streamlit dashboard (new)

1.4 Dataset Summary

Table 1: Dataset Statistics

Metric	Value
Total Speeches	43
Total Words	320,000+
Average Words per Speech	7,400+
Unique Entities Extracted	2,872
Engineered Features	80+
Time Period	2024-2025

Note: The data warehousing aspects (ETL pipeline, star schema design, entity extraction methodology) were documented in the previously submitted report. This report focuses on the NLP transformation, descriptive analytics, and predictive modeling components.

2 Data Collection: Web Scraping

2.1 Data Source

The speech transcripts were collected from **Rev.com**, a professional transcription service that provides accurate transcriptions of political speeches, press conferences, and public addresses.

2.2 Scraping Implementation

The scraping module was implemented using:

- **Selenium WebDriver:** For dynamic content rendering
- **BeautifulSoup:** For HTML parsing
- **Chrome (Headless):** Browser automation

2.2.1 Key Features of the Scraper

```
        class TrumpTranscriptScraper:
        def __init__(self, driver_path=None):
        self.base_url = "https://www.rev.com"
        self.driver = None
        self.speeches = []
        self.known_urls = [...]  # 44 pre-identified URLs

        def setup_driver(self):
        # Headless Chrome configuration
        chrome_options = Options()
        chrome_options.add_argument('--headless')
        chrome_options.add_argument('--no-sandbox')
```

```
    def scrape_transcript(self, url):
    # Extract: title, date, transcript text
    # Multiple parsing strategies for robustness

    def save_results(self):
    # Save to JSON and CSV formats
```

Listing 1: TrumpTranscriptScraper Class Structure

2.3 Data Extraction Strategy

The scraper employs multiple parsing strategies to ensure robust extraction:

1. **Primary Strategy:** Extract from `main-content` div
2. **Secondary Strategy:** Extract from `fs-toc-element='contents'` attribute
3. **Tertiary Strategy:** Extract from article body containers
4. **Fallback Strategy:** Filter all paragraphs by length and content

2.4 Output Format

Each scraped transcript contains:

Table 2: Raw Data Schema

Field	Type	Description
url	string	Source URL
title	string	Speech title
date	string	Date of speech
transcript	string	Full transcript text
scraped_at	datetime	Scraping timestamp

3 Data Cleaning Pipeline

3.1 Cleaning Operations

The data cleaning pipeline performs comprehensive text preprocessing:

3.1.1 HTML Tag Removal

All residual HTML markup is stripped using regex patterns and BeautifulSoup parsing.

3.1.2 Timestamp Removal

Transcript timestamps in various formats are removed:

```
        # Patterns removed:
        [HH:MM:SS], [MM:SS]
        (HH:MM:SS), (MM:SS)
```

3.1.3 Reaction Tag Removal

Audience reactions and editorial annotations are removed:

- (Applause), [Applause]
- (Cheers), [Cheers]
- (Laughter), [Laughter]
- (Inaudible), [Crosstalk]

3.1.4 Speaker Standardization

Speaker tags are normalized to consistent format:

```
        # Before: "Donald Trump:", "President Trump:", "Trump:"
        # After: "TRUMP:"
```

3.1.5 Noise Token Removal

Common noise tokens and metadata are filtered:

- “Transcript”
- “Rev.com”
- Copyright notices
- Fair use disclaimers

3.1.6 Punctuation Normalization

- Smart quotes converted to standard quotes
- Multiple consecutive punctuation marks reduced
- Spacing after sentence terminators ensured

3.1.7 Duplicate Removal

Consecutive duplicate paragraphs are identified and removed to eliminate scraping artifacts.

3.2 Cleaning Statistics

Table 3: Data Cleaning Results

Metric	Value
Speeches Processed	44
Average Character Reduction	15-20%
UTF-8 Validation	100% Pass
Duplicate Paragraphs Removed	200

4 Data Transformation: NLP Pipeline

4.1 NLP Tools and Libraries

The transformation pipeline utilizes:

- **spaCy** (en_core_web_sm): Tokenization, POS tagging, NER
- **NLTK:** N-gram extraction, stopwords
- **VADER:** Sentiment analysis
- **NRCLex:** Emotion classification
- **textstat:** Readability metrics
- **Sentence-Transformers:** Semantic embeddings

4.2 Transformation Operations

4.2.1 Sentence Segmentation

Text is segmented into sentences using spaCy's sentence boundary detection:

```
        def segment_sentences(self, text):
        doc = self.nlp(text)
        return [sent.text.strip() for sent in doc.sents]
```

4.2.2 Tokenization and POS Tagging

Each text is tokenized and annotated with:

- Tokens (words)
- Part-of-speech (POS) tags
- Lemmas (base forms)

4.2.3 Named Entity Recognition (NER)

Eight entity types are extracted:

Table 4: Named Entity Types

Type	Description
PERSON	People, including fictional characters
ORG	Companies, agencies, institutions
GPE	Countries, cities, states
DATE	Absolute or relative dates
MONEY	Monetary values
NORP	Nationalities, religious/political groups
FAC	Buildings, airports, highways
LOC	Non-GPE locations (mountains, rivers)

4.2.4 Sentiment Analysis (VADER)

VADER (Valence Aware Dictionary and sEntiment Reasoner) provides:

- **Positive Score:** Proportion of positive sentiment (0-1)
- **Negative Score:** Proportion of negative sentiment (0-1)
- **Neutral Score:** Proportion of neutral sentiment (0-1)
- **Compound Score:** Normalized weighted composite (-1 to +1)

The compound score formula:

$$\text{compound} = \frac{\sum_i v_i}{\sqrt{(\sum_i v_i)^2 + \alpha}} \tag{1}$$

where v_i are valence scores and α is a normalization constant.

4.2.5 Emotion Classification

Using NRCLex lexicon, eight emotions are scored:

- Anger, Fear, Joy, Sadness
- Surprise, Disgust, Trust, Anticipation

4.2.6 Readability Metrics

Multiple readability formulas are computed:

Flesch-Kincaid Grade Level:

$$\text{FK} = 0.39 \cdot \frac{\text{words}}{\text{sentences}} + 11.8 \cdot \frac{\text{syllables}}{\text{words}} - 15.59 \tag{2}$$

Flesch Reading Ease:

$$\text{FRE} = 206.835 - 1.015 \cdot \frac{\text{words}}{\text{sentences}} - 84.6 \cdot \frac{\text{syllables}}{\text{words}} \tag{3}$$

Gunning Fog Index:

$$\text{GF} = 0.4 \cdot \left(\frac{\text{words}}{\text{sentences}} + 100 \cdot \frac{\text{complex words}}{\text{words}} \right) \tag{4}$$

4.2.7 N-gram Extraction

Unigrams, bigrams, and trigrams are extracted with frequency counts, excluding stopwords.

4.2.8 Semantic Embeddings

384-dimensional sentence embeddings are generated using `all-MiniLM-L6-v2` model for semantic similarity analysis.

5 Feature Engineering

5.1 Feature Categories

Over 80 features are engineered across six categories:

5.1.1 Linguistic Features

- Average sentence length
- Standard deviation of sentence length
- Type-token ratio (lexical diversity)
- Average word length
- Unique word count
- All readability metrics

Type-Token Ratio (TTR):

$$\text{TTR} = \frac{|\text{unique tokens}|}{|\text{total tokens}|} \tag{5}$$

5.1.2 Rhetorical Features

- Anaphora patterns ("We will...", "I am...", "They are...")
- Contrast markers ("but", "however", "although")
- Repetition density
- Alliteration count
- Superlative usage ("best", "greatest", "most")

5.1.3 Political/Thematic Features

Keyword clusters for major themes:

- Economy: “jobs”, “trade”, “economy”, “taxes”, “business”
- Security: “military”, “defense”, “border”, “terrorism”
- Immigration: “immigration”, “border”, “wall”, “visa”
- Foreign Policy: “China”, “Russia”, “NATO”, “allies”

5.1.4 Emotional Features

- Overall sentiment (pos, neg, neu, compound)
- Sentiment statistics (mean, variance, range)
- Eight emotion scores
- Dominant emotion identification
- Emotional volatility (sentiment standard deviation)

5.1.5 Psychological Features

- **Pronoun Analysis:**
  - First person singular (“I”, “me”, “my”)
  - First person plural (“we”, “us”, “our”)
  - Second person (“you”, “your”)
  - I/We ratio (ego vs. collective focus)
- **Modal Verb Usage:** “will”, “should”, “must”, “can”
- **Certainty Markers:** “absolutely”, “definitely”, “clearly”
- **Power vs. Affiliation Words:**
  - Power: “strong”, “control”, “dominate”
  - Affiliation: “together”, “team”, “support”

5.1.6 Stylistic Features

- Adjective/Adverb ratio
- Question count
- Exclamation count
- All-caps word count (emphasis)

6 Data Warehousing Summary

Note: The complete data warehousing documentation, including detailed star schema design, ETL pipeline architecture, entity extraction methodology, and database schema definitions, was submitted in the previous report titled "Trump Speech Analysis Data Warehouse Documentation."

This section provides a brief summary of the key components for context:

6.1 Schema Overview

The data warehouse implements a **star schema** with:

- **1 Fact Table:** `fact_speech_metrics` containing 40+ computed metrics per speech
- **6 Dimension Tables:** Speech, Person, Organization, Location, Date, Topic
- **4 Bridge Tables:** For many-to-many relationships (speech-entity mappings)

6.2 Key Metrics Stored

The fact table stores metrics computed during feature engineering:

- Sentiment scores (VADER compound, positive, negative, neutral)
- Eight emotion scores (NRCLex)
- Readability metrics (Flesch-Kincaid, Gunning Fog, SMOG)
- Linguistic features (lexical diversity, sentence length statistics)
- Psychological indicators (pronoun ratios, power/affiliation words)
- Entity counts by type

6.3 ETL Pipeline

The three-stage ETL pipeline extracts data from the NLP outputs, preprocesses with normalization and surrogate key generation, and loads into the warehouse schema. See previous report for implementation details.

7 Descriptive Analytics

7.1 Linguistic Analysis

7.1.1 Readability Statistics

Table 5: Readability Metrics Summary (Approximate)

Metric	Mean	Min	Max
Flesch Reading Ease	75-77	65	85
Flesch-Kincaid Grade	5.5-6.0	4.0	8.0
Gunning Fog Index	7.5-8.5	6.0	10.0

The Flesch-Kincaid grade level of approximately 5-6 indicates speeches are highly accessible, targeting a 5th-6th grade reading level. This is consistent with political communication strategies designed to reach broad audiences.

7.1.2 Lexical Diversity

- Mean Type-Token Ratio: 0.13-0.15
- Lower TTR reflects repetitive rhetorical style common in persuasive speeches
- Longer speeches naturally have lower TTR due to word reuse

7.2 Rhetorical Analysis

7.2.1 Anaphora Patterns

Common anaphoric patterns detected:

- "We will..." (most frequent)
- "I am..."
- "They are..."
- "We are..."

7.2.2 Superlative Usage

High frequency of superlatives ("best", "greatest", "most", "biggest") characteristic of Trump's rhetorical style.

7.3 Emotional Analysis

7.3.1 Sentiment Distribution

Table 6: Sentiment Classification (VADER Compound Score)

Category	Count	Percentage
Positive (compound $>$ 0.05)	42	97.7%
Neutral (-0.05 to 0.05)	1	2.3%
Negative (compound $<$ -0.05)	0	0.0%

The overwhelmingly positive sentiment (compound scores approaching 1.0) reflects the promotional and optimistic framing typical of political rally speeches. Individual sentence-level sentiment shows more variance (ranging from -0.95 to +0.95), but aggregated speech-level sentiment is consistently positive.

7.3.2 Emotion Distribution

Average emotion scores reveal:

- **Trust** and **Anticipation:** Highest scores
- **Joy:** Moderate presence
- **Fear** and **Anger:** Present but controlled
- **Sadness** and **Disgust:** Lowest scores

7.4 Psychological Profiling

7.4.1 Pronoun Analysis

- Mean I/We ratio: $>$ 1.0 (ego-focused)
- High first-person singular usage indicates personal attribution style
- "We" usage increases in policy discussions

7.4.2 Power vs. Affiliation

- Power/Affiliation ratio: $>$ 1.0 (power-oriented)
- Consistent with authoritative communication style

7.5 Named Entity Analysis

7.5.1 Entity Distribution by Type

Table 7: Entity Type Distribution (Total: 2,872 unique entities)

Entity Type	Unique Count
DATE	915
PERSON	696
ORG	491
GPE	287
MONEY	242
NORP	108
LOC	87
FAC	46

7.5.2 Most Mentioned Entities

Top entities by mention frequency include political figures, countries (China, Russia, Ukraine), and organizations (NATO, FBI, DOJ).

8 Predictive Analytics

8.1 Overview

Six predictive models were developed to analyze Trump's behavioral patterns:

1. Entity Reaction Profiler
2. Personality Compatibility Predictor
3. Negotiation Success Predictor
4. Response Classifier (ML)
5. Psychological Influence Model
6. Trigger Word Detector

8.2 Model 1: Entity Reaction Profiler

8.2.1 Objective

Predict Trump's emotional reaction to specific entities (people, countries, organizations).

8.2.2 Methodology

1. Aggregate sentiment scores for each entity across all speeches
2. Compute baseline sentiment (global average)
3. Calculate centered sentiment: $s_{centered} = s_{raw} - s_{baseline}$
4. Classify reactions based on sentiment and emotion ratios

8.2.3 Reaction Classification

- **CELEBRATION_MODE:** High positive sentiment, low negative emotions
- **ATTACK_MODE:** High negative sentiment, high anger/fear
- **CRITICISM_MODE:** Moderate negative sentiment
- **NEUTRAL_MODE:** Near-baseline sentiment
- **UNPREDICTABLE:** High sentiment volatility

8.2.4 Output

```
{
    'entity': 'China',
    'reaction_type': 'CRITICISM_MODE',
    'sentiment_label': 'NEGATIVE',
    'volatility': 'MEDIUM',
    'confidence': 0.75
}
```

8.3 Model 2: Personality Compatibility Predictor

8.3.1 Objective

Predict how Trump will respond to individuals with different personality types.

8.3.2 Theoretical Framework

Based on the **Big Five Personality Model (OCEAN)**:

- **O**penness to Experience
- **C**onscientiousness
- **E**xtraversion
- **A**greeableness
- **N**euroticism

8.3.3 Trump's Derived Profile

Extracted from speech patterns:

Table 8: Trump's Big Five Profile (Estimated)

Trait	Score (0-100)
Openness	45
Conscientiousness	55
Extraversion	85
Agreeableness	25
Neuroticism	60

8.3.4 Compatibility Scoring

Using Interpersonal Circumplex Theory:

$$\text{Compatibility} = f(\text{Dominance Match}, \text{Warmth Match}, \text{Trait Differences}) \tag{6}$$

8.3.5 Response Categories

- COOPERATIVE (score > 70)
- TRANSACTIONAL (score 50-70)
- COMPETITIVE (score 30-50)
- HOSTILE (score < 30)

8.4 Model 3: Negotiation Success Predictor

8.4.1 Objective

Predict likelihood of successful negotiation based on topic, communication style, and approach strategies.

8.4.2 Input Parameters

- **Topic:** trade, economy, immigration, security, etc.
- **Communication Style:** flattering, transactional, assertive, diplomatic
- **Strategies:** show_win, media_angle, business_frame, loyalty_appeal

8.4.3 Prediction Formula

$$P_{success} = P_{base}(topic) \times E_{style} \times \prod_i M_{strategy_i} \tag{7}$$

Where:

- P_{base}: Base probability by topic

- E_{style}: Style effectiveness multiplier
- $M_{strategy}$: Strategy multipliers (boost or penalty)

8.4.4 Topic Favorability

Table 9: Topic Favorability Scores

Topic	Favorability (%)
Military	90
Trade	85
Economy	85
Immigration	80
Media	30
Environment	25

8.5 Model 4: Response Classifier (ML)

8.5.1 Objective

Classify Trump's likely response type using machine learning.

8.5.2 Response Categories

- ATTACK
- PRAISE
- NEGOTIATE
- DEFLECT
- NEUTRAL

8.5.3 Features Used

- sentiment_compound
- sentiment_neg, sentiment_pos
- power_affiliation_ratio
- certainty markers
- topic indicators

8.5.4 Model Architecture

- **Algorithm:** Random Forest Classifier
- **Alternative:** Gradient Boosting Classifier
- **Feature Scaling:** StandardScaler
- **Train/Test Split:** 75/25 with stratification

8.5.5 Training Process

```
        # Label generation (rule-based)
        def classify_response_type(row):
        if row['sentiment_neg'] > 0.15:
        return 'ATTACK'
        elif row['sentiment_pos'] > 0.2:
        return 'PRAISE'
        # ... additional rules

        # Model training
        rf_model = RandomForestClassifier(
        n_estimators=100,
        max_depth=10,
        random_state=42
        )
        rf_model.fit(X_train, y_train)
```

8.6 Model 5: Psychological Influence Model

8.6.1 Objective

Recommend effective persuasion tactics based on Trump's psychological profile.

8.6.2 Theoretical Framework

Based on **Cialdini's Six Principles of Persuasion**:

1. **Reciprocity:** Tendency to return favors
2. **Commitment/Consistency:** Desire to be consistent
3. **Social Proof:** Following others' actions
4. **Authority:** Deference to experts
5. **Liking:** Persuaded by people we like
6. **Scarcity:** Value rare opportunities

8.6.3 Trump's Psychological Profile

Derived from speech analysis:

- **Ego Score:** High (based on pronoun usage)
- **Approval Seeking:** High
- **Control Need:** Very High
- **Competitiveness:** Very High
- **Status Consciousness:** Very High
- **Emotional Reactivity:** High

8.6.4 Influence Effectiveness Ranking

Table 10: Influence Principle Effectiveness

Principle	Effectiveness (%)
Liking (Flattery)	90
Authority	75
Scarcity	70
Reciprocity	65
Social Proof	60
Commitment	55

8.7 Model 6: Trigger Word Detector

8.7.1 Objective

Identify words and phrases that trigger strong emotional responses.

8.7.2 Methodology

1. Extract all unique words from corpus
2. Compute co-occurrence with high-emotion segments
3. Calculate trigger score based on emotion intensity
4. Classify by valence (positive/negative)

8.7.3 Trigger Score Computation

$$\text{Trigger Score} = \alpha \cdot \text{Emotion Intensity} + \beta \cdot \text{Frequency} + \gamma \cdot \text{Sentiment Deviation} \quad (8)$$

8.7.4 Example Trigger Words

Table 11: High Trigger Words

Word/Phrase	Score	Valence
"fake news"	95	Negative
"winning"	90	Positive
"disaster"	88	Negative
"beautiful"	85	Positive
"corrupt"	82	Negative

9 Mega Trump Model: Unified Predictor

9.1 Architecture

The Mega Trump Model combines all six sub-models into a unified prediction system.

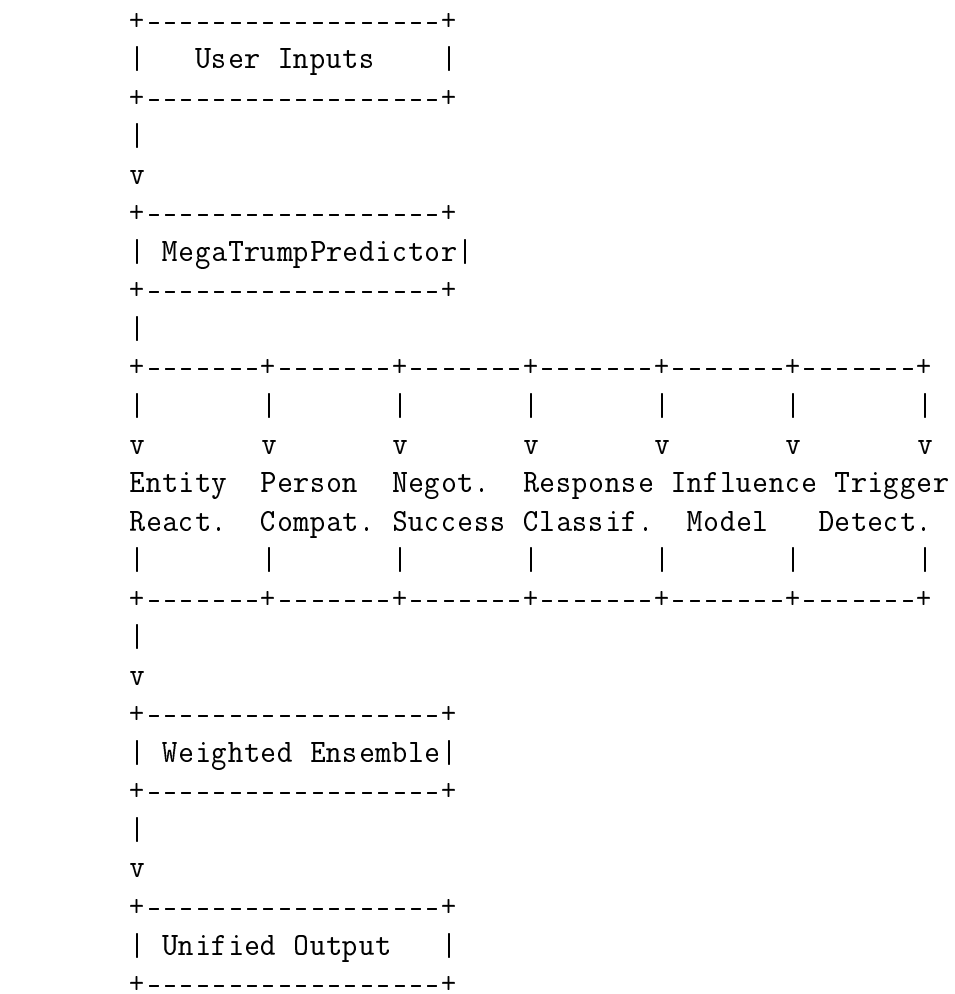

Figure 1: Mega Trump Model Architecture

9.2 Model Weights

Table 12: Sub-Model Weights

Sub-Model	Weight
Entity Reaction	0.25
Personality Compatibility	0.20
Negotiation Success	0.20
Response Classifier	0.15
Influence Strategy	0.10
Trigger Word Detector	0.10

9.3 Behavioral Categories

Predictions are mapped to five behavioral categories:

1. **COOPERATIVE:** Positive engagement likely
2. **TRANSACTIONAL:** Business-focused interaction
3. **COMPETITIVE:** Power dynamics expected
4. **HOSTILE:** High conflict risk
5. **NEUTRAL:** Uncertain/mixed signals

9.4 Confidence Calculation

$$\text{Confidence} = \text{Winning Category Score} \times \min(95, 50 + n_{models} \times 7.5) \tag{9}$$

Where n_{models} is the number of active sub-models (1-6).

9.5 Output Structure

```
{
    'overall_prediction': 'COOPERATIVE',
    'confidence': 75.5,
    'category_scores': {
        'COOPERATIVE': 75.5,
        'TRANSACTIONAL': 15.2,
        'COMPETITIVE': 5.0,
        'HOSTILE': 2.3,
        'NEUTRAL': 2.0
    },
    'sub_model_contributions': {...},
    'recommendations': [
    "Trump is likely to be receptive",
    "Maintain positive framing",
    "Best influence tactic: Liking"
    ]
}
```

10 Interactive Dashboard

10.1 Technology Stack

- **Framework:** Streamlit
- **Language:** Python
- **Visualization:** Native Streamlit components

10.2 Features

1. Model selection sidebar (7 models)
2. Dynamic input forms for each model
3. Real-time predictions
4. Visual confidence indicators
5. Recommendation display
6. Sub-model contribution breakdown

10.3 Usage

```
# Run the dashboard
streamlit run app.py
```

11 Conclusion

11.1 Summary of Achievements

This project successfully implemented:

1. A complete web scraping pipeline for transcript collection
2. Comprehensive data cleaning and preprocessing
3. Advanced NLP transformation with sentiment and emotion analysis
4. Feature engineering with 80+ derived attributes
5. Star schema data warehouse design
6. Six specialized predictive models
7. Unified Mega Trump Model for comprehensive predictions
8. Interactive Streamlit dashboard

11.2 Key Findings

11.2.1 Linguistic Patterns

- Average reading level: 5th-6th grade (highly accessible)
- High repetition and anaphora usage (persuasive style)
- Frequent superlatives (hyperbolic language)

11.2.2 Emotional Profile

- Overwhelmingly positive sentiment at speech level (97%+)
- High trust and anticipation emotions
- Controlled anger and fear for rhetorical effect

11.2.3 Psychological Indicators

- Ego-focused communication (high I/We ratio)
- Power-oriented language
- High certainty markers indicating confidence projection

11.3 Limitations

- Data limited to Rev.com transcripts
- Emotion classification based on lexicon (not ML)
- Predictive models based on observed patterns, not causal relationships
- Response classifier trained on rule-generated labels

11.4 Future Work

1. Expand dataset with historical speeches
2. Implement deep learning for emotion classification
3. Add real-time transcript analysis capability
4. Develop comparative analysis with other political figures
5. Enhance ML models with larger training data

References

1. Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *ICWSM*.

2. Mohammad, S.M. & Turney, P.D. (2013). Crowdsourcing a Word-Emotion Association Lexicon. *Computational Intelligence*.

3. Cialdini, R.B. (2009). *Influence: Science and Practice*. Pearson.

4. Pennebaker, J.W. et al. (2015). The Development and Psychometric Properties of LIWC2015. University of Texas at Austin.

5. spaCy: Industrial-Strength Natural Language Processing. https://spacy.io/

6. Streamlit: The fastest way to build data apps. https://streamlit.io/

A Project Structure

```
donald_trump/
+-- scripts/
|   +-- scrape.py              # Web scraping
|   +-- 01_data_cleaning.py    # Data cleaning
|   +-- 02_data_transformation.py  # NLP pipeline
|   +-- 03_feature_engineering.py  # Feature extraction
|   +-- 04_analysis_suite.py   # Descriptive analytics
|   +-- 05_entity_extraction.py    # NER cataloging
|   +-- 06_entity_relationships.py # Entity mapping
|   +-- 07_etl_extraction.py   # ETL extraction
|   +-- 08_etl_preprocessing.py    # ETL preprocessing
|   +-- 09_etl_load_warehouse.py   # ETL loading
|   +-- 10_data_quality.py     # Quality validation
+-- notebooks/
|   +-- 01-06: Exploratory analysis
|   +-- 07_predictive_entity_reaction.ipynb
|   +-- 08_trigger_word_detector.ipynb
|   +-- 09_personality_compatibility.ipynb
|   +-- 10_negotiation_predictor.ipynb
|   +-- 11_response_classifier.ipynb
|   +-- 12_influence_model.ipynb
+-- data/
|   +-- raw/           # Scraped data
|   +-- cleaned/       # Cleaned data
|   +-- transformed/   # NLP features
|   +-- entities/      # Entity catalogs
|   +-- staging/       # ETL staging
|   +-- warehouse/     # Data warehouse
|   +-- results/       # Analysis results
+-- sql/
```

```
|   +-- 01_create_schema.sql   # Star schema DDL
|   +-- 02_create_indexes.sql  # Index definitions
|   +-- 04_load_data.sql       # Generated inserts
+-- app.py             # Streamlit dashboard
+-- mega_trump_model.py    # Unified predictor
+-- config.yaml        # Configuration
+-- requirements.txt   # Dependencies
```

B Configuration File

Key configuration parameters in `config.yaml`:

- File paths for all data directories
- Cleaning parameters (reaction tags, noise tokens)
- NLP settings (spaCy model, embedding model)
- Feature engineering parameters (anaphora patterns, keyword clusters)
- Analysis settings (LDA topics, correlation thresholds)