Clinical Valence Testing: Measuring the Impact of Subjective Language on Clinical AI Diagnosis Predictions

Project Information

Project Name: Clinical Valence Testing with Attention Analysis

Lead/Mentor: *To be specified*

Contributors: *To be specified*

Current Funding: *To be specified*

Future Funding: *To be specified*

IRB #: *To be specified*

RMID: *To be specified*

SPARCRequest: *To be specified*

Executive Summary

This research project investigates how subjective language in clinical notes affects automated diagnosis prediction systems. Specifically, the project measures the impact of valence-laden words (pejorative, laudatory, and neutral descriptors) on transformer-based clinical NLP models. The framework systematically modifies clinical text with different types of subjective language and measures resulting shifts in diagnosis probability predictions, providing evidence for potential algorithmic bias in healthcare AI systems.

Key Research Questions

1. **How do pejorative patient descriptors affect diagnosis predictions?** Do negative characterizations (e.g., "non-compliant," "difficult," "drug-seeking") systematically alter predicted diagnosis probabilities?

2. **How do laudatory patient descriptors affect diagnosis predictions?** Do positive characterizations (e.g., "compliant," "cooperative," "pleasant") influence model outputs?

3. **Are certain diagnoses more susceptible to valence-induced shifts?** Can we identify diagnosis codes that are disproportionately affected by subjective language?

4. **What attention patterns correlate with valence-induced prediction shifts?** How do transformer attention weights change when valence terms are introduced?

Introduction and Context

The Problem

In clinical practice, medical documentation often contains subjective language describing patient behavior and characteristics. Terms like "non-compliant," "difficult," or "drug-seeking" may reflect provider biases rather than objective clinical findings. As healthcare increasingly relies on AI systems trained on such documentation, there is growing concern that these systems may perpetuate or amplify existing biases.

Our Approach

This project develops a systematic framework for testing how clinical NLP models respond to valence-laden language through controlled text perturbation experiments. By introducing or removing specific types of subjective descriptors and measuring resulting changes in model predictions, we can quantify the sensitivity of clinical AI systems to potentially biased language.

Key Findings (Expected)

- Quantified impact of pejorative vs. laudatory language on diagnosis predictions
- Identification of diagnosis codes most susceptible to valence-induced shifts
- Statistical significance testing using paired permutation tests (Yeh 2000)
- Attention pattern analysis revealing model focus on valence terms

Methods

1. Data Description

Dataset: Clinical notes with associated ICD diagnosis codes

Required Data Format:

- CSV file with clinical text and diagnosis codes
- Text column: Full clinical note text
- Code column: Comma-separated ICD diagnosis codes (short form)

• Minimum code frequency threshold: Configurable (default: 100 occurrences)

Data Preprocessing:

• Code frequency filtering to ensure statistical power
• Text validation for minimum medical content
• Duplicate detection and handling

2. Model Description

Primary Model: DATEXIS/CORe-clinical-diagnosis-prediction

• Architecture: BERT-based transformer for multi-label classification
• Input: Clinical note text (max 512 tokens)
• Output: Probability distribution over ICD diagnosis codes

Model Configuration:

• Batch size: 128 (optimized for NVIDIA H100 GPU)
• Attention extraction: Layer 11, Head 11 (configurable)
• Attention aggregation: Average pooling across heads

Hardware Requirements:

• CUDA 12.1 compatible GPU (optimized for H100)
• Mixed precision support (FP16/FP8)

3. Valence Shift Framework

The core methodology involves systematic text transformations using four shift types:

3.1 Pejorative Shift

Introduces negative patient descriptors at four severity levels:

Level	Terms
NON_COMPLIANT	non-compliant, negligent, careless, irresponsible, unreliable
UNCOOPERATIVE	uncooperative, difficult, problematic, demanding, argumentative, defiant
RESISTANT	resistant, hostile, aggressive, disruptive, troublesome
DIFFICULT	difficult, manipulative, malingering, attention-seeking, drug-seeking, troublesome

3.2 Laudatory Shift

Introduces positive patient descriptors at four levels:

Level	Terms
COMPLIANT	compliant, adherent, responsible, reliable

COOPERATIVE	cooperative, agreeable, courteous, considerate
PLEASANT	pleasant, agreeable
RESPECTFUL	respectful, courteous, considerate

3.3 Neutral Valence Shift

Introduces objective descriptors:

- Terms: typical, average, regular, standard, usual, presenting, referred, evaluated, assessed, monitored

3.4 Neutralize Shift

Removes all valence terms (baseline condition):

- Strips all pejorative, laudatory, and neutral descriptors
- Creates maximally objective text for comparison

4. Text Transformation Algorithm

The shift implementation uses a sophisticated pattern-matching system:

1. **Position Detection:** Identifies optimal insertion point using regex patterns for:

- Age patterns (e.g., "45 yo", "67-year-old")
- Gender patterns (e.g., "male", "female", "woman")
- Patient type patterns (e.g., "patient", "inpatient")
- Ethnicity patterns
- Medical condition patterns

2. **Term Insertion:** Places valence descriptor after patient identifier:

```
Original: "45 yo male with chest pain"
Pejorative: "45 yo non-compliant male with chest pain"
Laudatory: "45 yo compliant male with chest pain"
```

3. **Conflict Resolution:** Removes existing valence terms before insertion to ensure controlled conditions

5. Statistical Analysis Framework

5.1 Primary Statistical Tests

Parametric Tests:

- Paired t-test for normally distributed differences
- 95% confidence intervals for mean shifts

Non-Parametric Tests:

- Wilcoxon signed-rank test for non-normal distributions
- Mann-Whitney U test for unpaired comparisons

Permutation Tests (Yeh 2000 Approximate Randomization):

- Paired permutation test with 10,000 iterations
- Stratified permutation test for binary outcomes
- Bootstrap confidence intervals (5,000 iterations)

5.2 Multiple Comparison Correction

- False Discovery Rate (FDR) correction using Benjamini-Hochberg method
- Bonferroni correction (optional)
- Significance level: 0.05 (configurable)

5.3 Effect Size Measures

- Cohen's d (standardized mean difference)
- Hedges' g (bias-corrected effect size)
- Interpretation thresholds:
- Small: |d| < 0.2
- Medium: 0.2 <= |d| < 0.8
- Large: |d| >= 0.8

6. Evaluation Approach

Primary Metrics:

- Mean probability shift per diagnosis code
- Number of significantly affected diagnoses (after FDR correction)
- Proportion of diagnoses with large effect sizes

Secondary Metrics:

- Attention weight changes for valence terms
- Cross-condition consistency analysis
- Diagnosis category-level aggregations

Technical Implementation

Core Architecture

Module	Purpose	Key Functions
<font name="Courier" size="9">main.py</font>	CLI entry point	Fire-based command interface
<font name="Courier" size="9">valence_testing.py</font>	Core testing framework	ValenceTester class orchestration
<font name="Courier" size="9">prediction.py</font>	Model inference	PredictionModule with attention extraction
<font name="Courier" size="9">statistical_analysis.py</font>	Statistical tests	StatisticalAnalyzer with permutation tests

<font name="Courier" size="9">utils.py</font>	Utilities	Text processing, pattern matching
<font name="Courier" size="9">config_loader.py</font>	Configuration	YAML-based configuration management

Shift Implementation Classes

ass	File	Description
nt name="Courier" size="9">BaseShift</font>	<font name="Courier" size="9">test_shifts/base_shift.py</font>	Abstract base class for all sh
nt name="Courier" size="9">PejorativeShift</font>	<font name="Courier" size="9">test_shifts/pejorative_shift.py</font>	Negative descriptor injection
nt name="Courier" size="9">LaudatoryShift</font>	<font name="Courier" size="9">test_shifts/laudatory_shift.py</font>	Positive descriptor injection
nt name="Courier" size="9">NeutralValShift</font>	<font name="Courier" size="9">test_shifts/neutralVal_shift.py</font>	Neutral descriptor injection
nt name="Courier" size="9">NeutralizeShift</font>	<font name="Courier" size="9">test_shifts/neutralize_shift.py</font>	All valence term removal

Configuration System

Configuration Files:

- `config.yaml`: Primary configuration file
- Dataclass-based configuration with type validation
- Environment-aware defaults (GPU detection, paths)

Key Configuration Sections:

```
model:
name: DATEXIS/CORe-clinical-diagnosis-prediction
max_length: 512
batch_size: 128
attention:
layer_num: 11
head_num: 11

data:
code_label: short_codes
text_label: text
min_code_frequency: 100

analysis:
baseline: neutralize
significance_level: 0.05
multiple_testing_correction: fdr_bh
```

Visualization Capabilities

Static Visualizations:

- Diagnosis probability shift heatmaps
- Effect size distribution plots
- Attention weight comparisons

Interactive Visualizations (Plotly):

• 3D surface plots of diagnosis shifts
• Interactive attention explorers
• Comprehensive HTML dashboards

Experimental Results

Output Structure

Each experiment generates:

1. **Diagnosis Predictions:** `{shift_type}_shift_diagnosis.csv`
• Per-sample probability predictions for each diagnosis code
• Columns: NoteID, Valence, Val_class, [diagnosis codes]

2. **Attention Weights:** `{shift_type}_shift_attention.csv`
• Token-level attention weights
• Columns: NoteID, Word, AttentionWeight, Val_class

3. **Clinical Notes:** `{shift_type}_clinical_notes.csv`
• Original and transformed text pairs
• Columns: NoteID, OriginalText, TransformedText, Val_class

4. **Statistical Report:** `statistical_analysis.txt`
• Comprehensive statistical analysis results
• Significant diagnoses with effect sizes
• Multiple comparison corrected p-values

Interpretation Guidelines

Effect Size	Clinical Interpretation		
	d	< 0.2	Negligible impact
0.2 <=	d	< 0.5	Small but detectable effect
0.5 <=	d	< 0.8	Moderate clinical significance
	d	>= 0.8	Large, clinically meaningful shift

Key Takeaways and Limitations

Limitations

1. **Single Model Evaluation:** Results are specific to the tested model; generalization to other clinical NLP systems requires validation
2. **English Language Only:** Current implementation focuses on English clinical text
3. **Controlled Perturbations:** Real-world bias may manifest differently than experimental manipulations
4. **Diagnosis Code Coverage:** Limited to codes meeting frequency thresholds

Key Takeaways

1. **Quantifiable Bias:** Framework enables systematic measurement of language-induced prediction shifts
2. **Reproducibility:** Deterministic random seeds and configuration management ensure reproducible experiments
3. **Statistical Rigor:** Multiple comparison correction and permutation testing provide robust inference
4. **Interpretability:** Attention analysis offers insights into model decision-making

Future Directions

Next Steps

1. **Multi-model Comparison:** Extend evaluation to additional clinical NLP models (e.g., ClinicalBERT, BioBERT, PubMedBERT)
2. **Demographic Intersectionality:** Investigate interactions between valence language and patient demographic characteristics
3. **Longitudinal Analysis:** Track prediction shifts across sequential clinical notes
4. **Mitigation Strategies:** Develop and evaluate debiasing techniques

Next Questions to Answer

1. Do valence-induced shifts differ by medical specialty or note type?
2. Can we identify model architectures that are more robust to subjective language?
3. What training data characteristics correlate with valence sensitivity?

Next Experiments to Run

1. Cross-validation across multiple dataset splits
2. Sensitivity analysis for valence term placement positions
3. Dose-response analysis for number of inserted valence terms

Resources

Code Repository

Location: Current project directory (`clinical-valence-testing/`)

Key Entry Points:

```
# Run complete valence testing pipeline
python main.py --test_set_path ./data/test.csv \
--shift_keys neutralize pejorative laud neutralval \
--task valence_testing \
--save_dir ./results

# Run statistical analysis only
python main.py --task statistical_analysis \
--results_dir ./results

# Generate interactive visualizations
python interactive_viz.py --results_dir ./results \
--output dashboard.html
```

Datasets Used

- Clinical notes with ICD diagnosis codes
- Test set path: `./data/test.csv`
- Required columns: `text`, `short_codes`

Dependencies

Core Frameworks:

- PyTorch >= 2.1.0 (CUDA 12.1)
- Transformers >= 4.35.0
- NumPy, Pandas, SciPy

Statistical Analysis:

- statsmodels >= 0.14.0
- scikit-learn >= 1.3.0

Visualization:

- Matplotlib, Seaborn
- Plotly >= 5.17.0

Acknowledgements

Lead/Mentor: *To be specified*

Contributors: *To be specified*

Past Contributors: *To be specified*

Current Funding: *To be specified*

Future Funding: *To be specified*

Technical Appendix

A. Statistical Test Details

Paired Permutation Test (Yeh 2000)

The paired permutation test is implemented as follows:

1. Calculate observed mean difference: `d_obs = mean(treatment - baseline)`
2. For each of n_permutations iterations:
- Randomly flip sign of each paired difference
- Calculate permuted mean difference
3. P-value = (count of |d_perm| >= |d_obs| + 1) / (n_permutations + 1)

Bootstrap Confidence Interval

1. For each of n_bootstrap iterations:
- Sample with replacement from difference scores
- Calculate statistic of interest
2. CI bounds = percentiles of bootstrap distribution

B. Pattern Matching Regular Expressions

Age Patterns:

```
r'(\d{2}[ ]?y/?o)' # "45 yo", "45y/o"
r'(\d{2}-year-old)' # "45-year-old"
r'(\d{2} year old)' # "45 year old"
r'(\[\*\*Age over 90 \*\*\])' # MIMIC-style anonymization
```

Gender Patterns:

```
r'(female|woman|lady)'
r'(male|man|gentleman)'
```

C. ICD Code Categories

The framework supports mapping between short codes and full ICD-10-CM descriptions via the `icd_translations` configuration section.

References

1. Yeh, A. (2000). More accurate tests for the statistical significance of result differences. COLING 2000.
2. Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences.
3. Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing.

Document generated: January 2026

Clinical Valence Testing Framework v1.0