OpenAI Patent Policy Archive Manipulation - Reproduction Guide

Prerequisites

- Unix/Linux terminal or macOS Terminal
- `curl` command available
- `jq` for JSON parsing (install via `brew install jq` or `apt install jq`)
- Basic command line knowledge

Step 1: Get Archive Timeline

Extract all archived timestamps for the OpenAI patent policy:

```
# Get all archived timestamps
curl -s "https://web.archive.org/cdx/search/cdx?url=openai.com/approach-to-patents/&output=json&fl=ti

# View the timestamps
cat patent_timestamps.txt
```

Step 2: Systematic Archive Extraction

Create directory and extract all versions systematically:

```
# Create directory for versions
mkdir patent_policy_versions

# Extract all versions using systematic method
counter=1
for timestamp in $(cat patent_timestamps.txt); do
    echo "Extracting version $counter: $timestamp"
    curl -s "https://web.archive.org/web/${timestamp}id_/https://openai.com/approach-to-patents/" > "

    # Also try alternative method
    curl -s "https://web.archive.org/web/${timestamp}/https://openai.com/approach-to-patents/" > "pat

    counter=$((counter + 1))
    sleep 1  # Be respectful to the archive
done
```

Step 3: Analyze Systematic Results

Check file sizes and content fingerprints:

```
# Analyze all versions
echo "=== PATENT POLICY VERSION ANALYSIS ===" > analysis_summary.txt

for file in patent_policy_versions/*.html; do
    echo "=== $(basename $file) ===" >> analysis_summary.txt
    echo "Size: $(wc -c < "$file") bytes" >> analysis_summary.txt
    echo "Date references:" >> analysis_summary.txt
    grep -i "effective\|date\|published\|updated\|october\|2024" "$file" >> analysis_summary.txt
    echo "JavaScript embedded dates:" >> analysis_summary.txt
    grep -o '"publicationDateText":"[^"]*"' "$file" >> analysis_summary.txt

    if grep -q "403\|404\|Forbidden\|Not Found" "$file"; then
        echo "STATUS: Error page detected" >> analysis_summary.txt
    else
        echo "STATUS: Content page" >> analysis_summary.txt
    fi
    echo "" >> analysis_summary.txt
done

# Display results
cat analysis_summary.txt
```

Step 4: Generate Content Fingerprints

Create MD5 hashes to detect identical content:

```
# Generate content fingerprints
for file in patent_policy_versions/version_*.html; do
    if [ -s "$file" ]; then
        echo "$(basename "$file"): $(md5sum "$file" 2>/dev/null | cut -d' ' -f1 || md5 "$file" | cut -
    fi
done > content_fingerprints.txt

echo "Content fingerprint analysis:"
cat content_fingerprints.txt
```

Step 5: Direct Archive Access

Test direct access to specific archived versions:

```
# Direct access to July 2025 version (known to contain publication date)
curl --compressed -s 'https://web.archive.org/web/20250714184707id_/https://openai.com/approach-to-pa

# Check size difference
```

```
echo "July direct access: $(wc -c < july_direct.html) bytes"
echo "Systematic extraction: 208 bytes (typical systematic result)"
echo "Size difference: $(($(wc -c < july_direct.html) - 208)) bytes"
```

Step 6: Publication Date Analysis

Search for publication dates in different access methods:

```
# Search systematic extractions for publication dates
echo "=== SYSTEMATIC EXTRACTION PUBLICATION DATES ==="
for file in patent_policy_versions/*.html; do
    echo "$(basename "$file"):"
    grep -o '"publicationDateText":"[^"]*"' "$file" 2>/dev/null || echo "No publication date found"
done

# Search direct access for publication dates
echo "=== DIRECT ACCESS PUBLICATION DATES ==="
grep -o '"publicationDateText":"[^"]*"' july_direct.html || echo "No publication date found in direct

# Extract any date-related JSON from direct access
echo "=== DATE-RELATED JSON EXTRACTION ==="
grep -C5 "publicationDateText\|October.*2024" july_direct.html
```

Step 7: Metadata Cross-Verification

Verify archive metadata consistency:

```
# Check archive metadata for July timestamp
curl -s "https://web.archive.org/cdx/search/cdx?url=openai.com/approach-to-patents/&from=20250714&to=

# Compare with actual retrieved content
echo "Metadata vs Retrieved Content Analysis:"
echo "Metadata size: [from above query]"
echo "Direct access size: $(wc -c < july_direct.html) bytes"
echo "Systematic access size: $(wc -c < patent_policy_versions/version_*20250714*.html 2>/dev/null | l
```

Expected Results (Based on Original Investigation)

Systematic Extraction Results

- **All versions**: ~208 bytes
- **Identical MD5**: `37cddf077c5c845db581950f95c9034b` (or similar)
- **Publication dates**: None found
- **Content**: Minimal HTML structure

Direct Access Results

- **July 2025**: ~200,102 bytes
- **Publication date**: `"publicationDateText":"October 11, 2024"`
- **Full content**: Complete website structure

Key Indicators of Manipulation

1. **Size differential**: 199,000+ byte difference between methods
2. **Content fingerprints**: Identical across all systematic extractions
3. **Publication dates**: Present in direct access, absent in systematic
4. **Missing timestamps**: July 2025 timestamp missing from systematic extraction

Verification Commands

Quick verification of key findings:

```
# Count unique file sizes in systematic extraction
ls -la patent_policy_versions/*.html | awk '{print $5}' | sort | uniq -c

# Count unique MD5 hashes
cat content_fingerprints.txt | cut -d':' -f2 | sort | uniq -c

# Search for any publication dates in systematic extractions
find patent_policy_versions -name "*.html" -exec grep -l "publicationDateText" {} \; | wc -l

# Verify direct access contains publication date
grep -q "publicationDateText.*October.*2024" july_direct.html && echo "Publication date found in dire
```

Documentation

Save all results for analysis:

```
# Create comprehensive report
{
    echo "=== ARCHIVE MANIPULATION INVESTIGATION REPORT ==="
    echo "Date: $(date)"
    echo "Target: openai.com/approach-to-patents/"
    echo ""
    echo "=== SYSTEMATIC EXTRACTION SUMMARY ==="
    cat analysis_summary.txt
    echo ""
    echo "=== CONTENT FINGERPRINTS ==="
    cat content_fingerprints.txt
    echo ""
    echo "=== SIZE ANALYSIS ==="
    echo "Direct access (July 2025): $(wc -c < july_direct.html) bytes"
    echo "Systematic extraction: 208 bytes (typical)"
    echo "Differential: $(($(wc -c < july_direct.html) - 208)) bytes removed"
```

```
    echo ""
    echo "=== PUBLICATION DATE SEARCH ==="
    echo "Direct access contains: $(grep -o '"publicationDateText":"[^"]*"' july_direct.html || echo '
    echo "Systematic extractions contain: $(find patent_policy_versions -name "*.html" -exec grep -l '
} > investigation_report.txt

echo "Investigation complete. Results saved to investigation_report.txt"
```

Notes for Reproduction

1. **Rate Limiting**: Add delays between requests to respect archive.org
2. **Timestamps**: Some timestamps may not be accessible depending on timing
3. **Network Issues**: Some requests may fail due to network conditions
4. **Platform Differences**: MD5 command syntax varies (Linux: `md5sum`, macOS: `md5`)

This methodology provides independent verification of the archive manipulation findings through reproducible technical analysis.