Craigslist Bargains Dataset

Overview

The Craigslist Bargains dataset is a collection of over 6,600 negotiation dialogues scraped from Craigslist, spanning multiple product categories. Each dialogue represents a buyer-seller negotiation over an item's price, complete with item details, negotiation strategies, and outcomes.

Dataset Statistics

```
Total Records: 6,608 (98.89% of raw data)
Splits:
- Train: 5,202 (78.72%)
- Test: 809 (12.24%)
- Validation: 597 (9.03%)

Categories:
- Vehicles: 2,052 (31.05%)
- Furniture: 1,677 (25.38%)
- Electronics: 1,524 (23.06%)
- Housing: 1,355 (20.51%)
```

Raw Data Structure

The original dataset is provided as three JSON files (train.json, test.json, validation.json). Each JSON entry contains:

Item Information

- Category
- Title
- Description
- Price
- Images (references)

Negotiation Data

- Agent roles (buyer/seller)
- Target prices
- Dialogue turns
- Price offers
- Negotiation outcomes

Sample Raw JSON Entry

```
{
  "agent_info": {
    "Bottomline": ["None", "None"],
    "Role": ["buyer", "seller"],
```

```
    "Target": [120.0, 200.0]
  },
  "agent_turn": [0, 1, 0, ...],
  "dialogue_acts": {
    "intent": ["intro", "unknown", ...],
    "price": [-1.0, -1.0, ...]
  },
  "items": {
    "Category": ["electronics", ...],
    "Description": ["Product description..."],
    "Images": ["image_ref.jpg", ...],
    "Price": [200.0, ...],
    "Title": ["Product title"]
  },
  "utterance": ["Hi I'm interested", ...]
}
```

Processed Data Format

The dataset is preprocessed into CSV format with standardized features for inference tasks.

CSV Columns

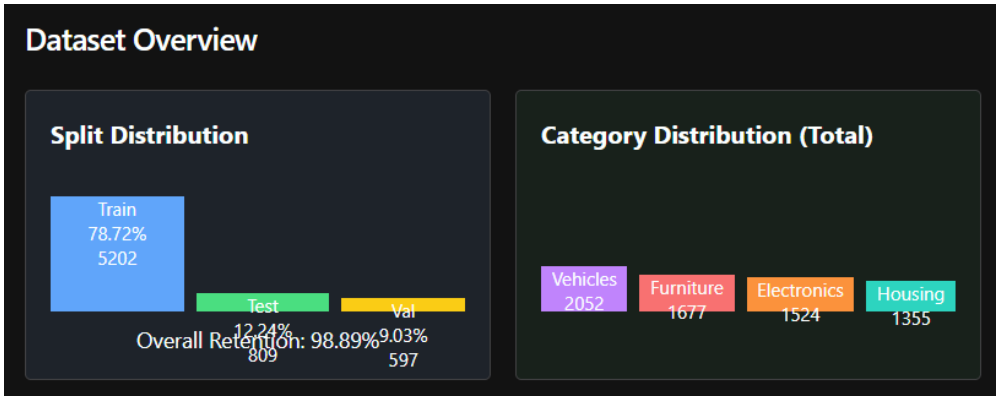
1. **Core Identifiers**
   - `scenario_id`: Unique identifier for each negotiation
   - `split_type`: train/test/validation indicator
   - `category`: Mapped category (electronics, furniture, housing, vehicles)
2. **Price Information**
   - `list_price`: Original listing price
   - `buyer_target`: Buyer's target price
   - `seller_target`: Seller's target price
   - `price_delta_pct`: Percentage difference between targets
   - `relative_price`: Price relative to category median
3. **Text Features**
   - `title`: Item title
   - `description`: Item description
   - `title_token_count`: Number of words in title
   - `description_length`: Character count of description
4. **Quality Metrics**
   - `data_completeness`: Record completeness score (0-1)
   - `price_confidence`: Price relationship validation
   - `has_images`: Binary indicator for image presence

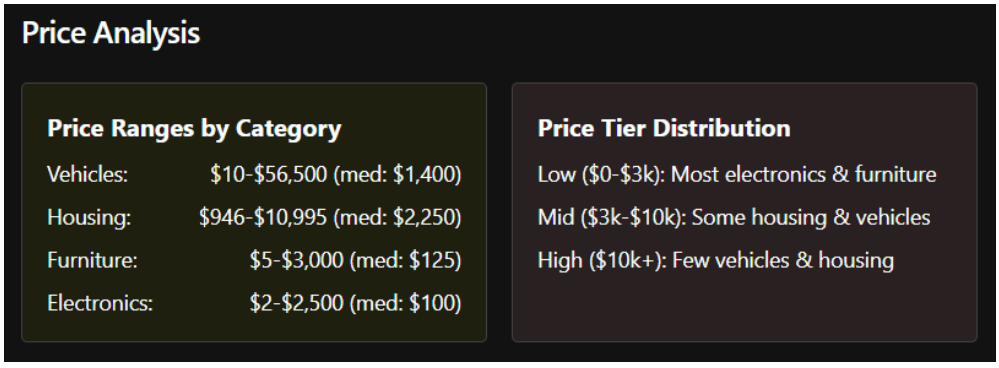
Price Categories

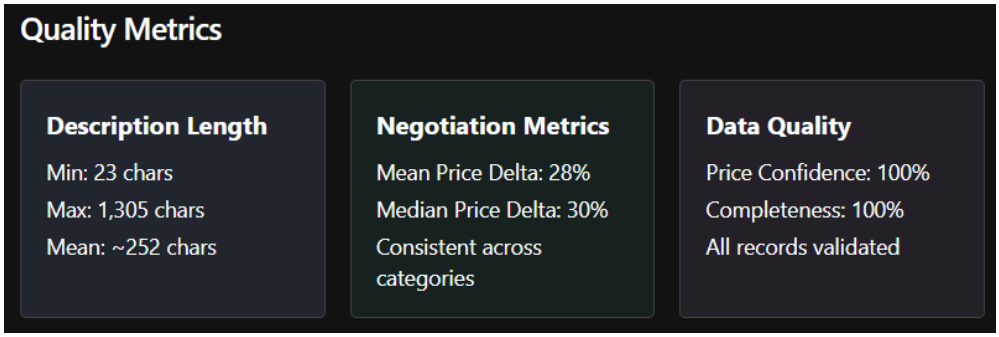
```
Low Tier: $0-3,000
- Most electronics items
- Most furniture items
- Some vehicles

Mid Tier: $3,000-10,000
- Some housing listings
- Many vehicles

High Tier: $10,000+
- Some vehicles
- Few housing listings
```

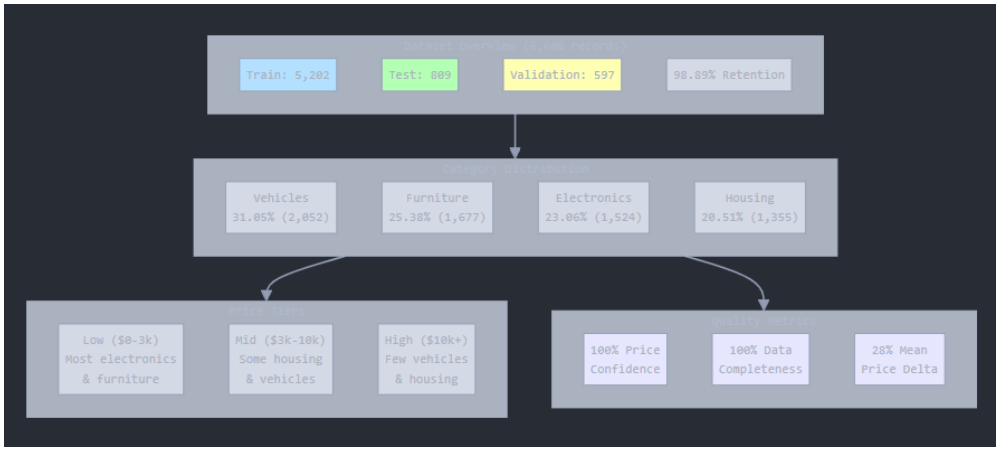
Data Quality

- Overall retention rate: 98.89%
- Complete price validation across all splits
- 100% data completeness in processed records
- Description lengths: 23-1,305 characters (mean ~252)
- Consistent negotiation patterns (~28% price delta)









Processing Pipeline

1. **Data Loading**
   - Parse raw JSONs
   - Extract core fields
   - Validate structure
2. **Cleaning & Normalization**
   - Price validation and cleaning
   - Text normalization
   - Category mapping
   - Missing value handling
3. **Feature Engineering**
   - Calculate price relationships
   - Generate text metrics
   - Compute quality scores
   - Create category-specific features
4. **Quality Filtering**
   - Minimum description length (>20 chars)
   - Price relationship validation
   - Data completeness checks (>80%)

Dataset Goals

1. **Negotiation Analysis**
   - Study price negotiation strategies
   - Analyze buyer-seller dynamics
   - Understand category-specific patterns
2. **Model Training**
   - Develop negotiation agents
   - Price prediction models
   - Category-specific strategy optimization
3. **Market Research**
   - Price range analysis
   - Category-specific patterns
   - Negotiation success factors

Usage

```
# Load processed data
train_df = pd.read_csv('train.csv')
test_df = pd.read_csv('test.csv')
val_df = pd.read_csv('validation.csv')

# Access statistics
with open('dataset_info.json', 'r') as f:
    stats = json.load(f)
```

Directory Structure

```
agreemate/
└── data/
    └── craigslist_bargains/
        ├── raw/
        │   ├── dataset_info.json
        │   ├── test.json
        │   ├── train.json
        │   └── validation.json
        ├── downloader.py
        ├── reformatter.py
        ├── train.csv
        ├── test.csv
        ├── validation.csv
        └── dataset_info.json
```

References

- Original dataset: stanfordnlp/craigslist_bargains

- Paper: Decoupling Strategy and Generation in Negotiation Dialogues (He et al., 2018)

Citation

```
@article{he2018decoupling,
  title={Decoupling Strategy and Generation in Negotiation Dialogues},
  author={He, He and Chen, Derek and Balakrishnan, Anusha and Liang, Percy},
  journal={arXiv preprint arXiv:1808.09637},
  year={2018}
}
```