I. Introduction
Real estate advertisements:
- Useful, but unstructured, plain text
Need for structured representation:
- User queries (e.g., “at least 3 bedrooms”) Additional services (e.g., statistics, price prediction)
New problem:
- Extract a structured description of the property based on the ad

III. Example
Original ad:
The property includes an apartment house with a garage. The house has a living room and a bathroom with shower. The garage is equipped with an electric gate and a bike wall bracket.

Structured representation:
- `house` | mention = “apartment house”
- `living room` | mention = “living room”
- `bathroom` | mention = “bathroom”
- `shower` | mention = “shower”
- `garage` | mention = “garage”
- `gate` | mention = “electric gate”
- `bike bracket` | mention = “bike wall bracket”

V. Step (1): Entity recognition
- Extract the entity boundaries
- Map the type of the entities
- CRF

VI. Entity recognition results

<table>
<thead>
<tr>
<th>Entity type</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>property</td>
<td>3170</td>
<td>1912</td>
<td>2117</td>
<td>0.62</td>
<td>0.59</td>
<td>0.61</td>
</tr>
<tr>
<td>floor</td>
<td>2685</td>
<td>515</td>
<td>529</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>space</td>
<td>11952</td>
<td>2053</td>
<td>2003</td>
<td>0.85</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>subspace</td>
<td>4338</td>
<td>575</td>
<td>1181</td>
<td>0.88</td>
<td>0.79</td>
<td>0.83</td>
</tr>
<tr>
<td>field</td>
<td>2083</td>
<td>708</td>
<td>718</td>
<td>0.75</td>
<td>0.74</td>
<td>0.75</td>
</tr>
<tr>
<td>extra building</td>
<td>253</td>
<td>34</td>
<td>143</td>
<td>0.88</td>
<td>0.64</td>
<td>0.74</td>
</tr>
</tbody>
</table>

| Overall     | 24481| 5789 | 6791 | 0.81      | 0.78   | 0.80 |

IV. Dependency types

- Projective dependency structures, i.e., crossing dependencies are not allowed
- Non-projective dependency structures, i.e., dependencies are allowed to cross
- Significant number of non-projective arcs (26%) in real estate classifieds
- Entities in the part-of relation are non-adjacent

VII. Steps (2)+(3): Part-of tree construction
- The aim is to connect each entity to its parent
- Similar to dependency parsing but map only the identified entity set x (e.g., house) to the dependency structure y
- Evaluation:
  - Dependency parsing subtask by itself
  - Pipeline approach combining both sequence labeling and dependency parsing subtasks (steps (1)+(2))

Globally trained model (MTT)
- Train globally normalized models that learn directed spanning trees
- Score parse trees for a given sentence
- The conditional distribution over all dependency structures $y \in T(x)$:
  \[
  P(y|x; \theta) = \frac{1}{Z(x; \theta)} \exp \left( \sum_{h,m,y} \theta_{h,m} \right)
  \]
- Normalized by $Z(x; \theta)$ requires a summation over all $T(x)$
- MTT allows direct computation as $\text{dot}(L(\theta))$ where $L$ is the Laplacian matrix of the graph

VIII. Part-of tree construction results

<table>
<thead>
<tr>
<th>Model</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB</td>
<td>14816</td>
<td>17368</td>
<td>17368</td>
<td>0.46</td>
<td>0.46</td>
<td>0.46</td>
</tr>
<tr>
<td>Thresh.</td>
<td>15723</td>
<td>6365</td>
<td>16461</td>
<td>0.71</td>
<td>0.49</td>
<td>0.58</td>
</tr>
<tr>
<td>Edmond</td>
<td>22058</td>
<td>10126</td>
<td>10126</td>
<td>0.69</td>
<td>0.69</td>
<td>0.69</td>
</tr>
<tr>
<td>MTT</td>
<td>22361</td>
<td>9823</td>
<td>9823</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
</tr>
</tbody>
</table>

| Overall    | 12426| 17850| 19848| 0.41      | 0.39   | 0.40 |

Conclusion & Future work

**CONCLUSION**
- MTT approach better in dependency parsing subtask
- Locally trained approach better in pipeline setting

**FUTURE WORK**
- Joint models (perform all steps at once) for non-projective dependency parsers
- Neural scoring functions