Reconstructing the house from the ad: Structured prediction on real estate classifieds

Giannis Bekoulis, Johannes Deleu, Thomas Demeester and Chris Develder
Ghent University - imec, Belgium

GHENT UNIVERSITY   imec

I. Introduction

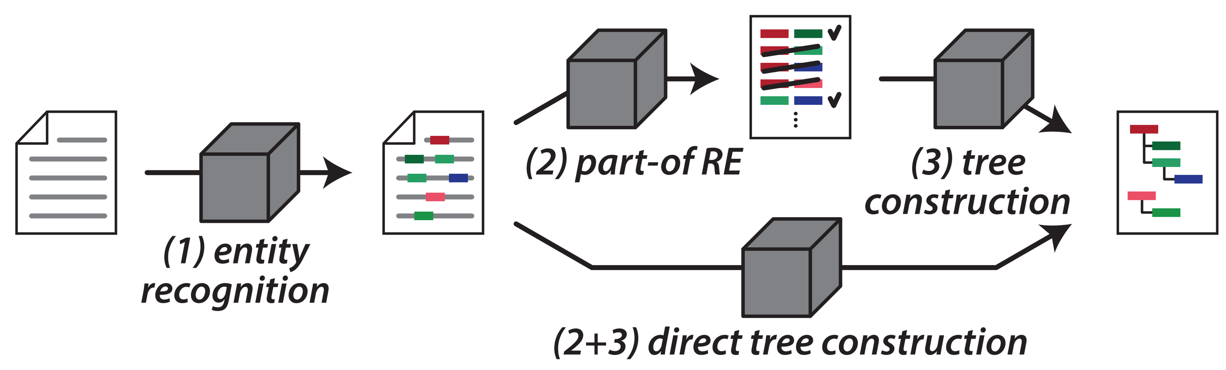
Real estate advertisements:
- Useful, but unstructured, plain text

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

II. Structured prediction model

- **Entity recognition:** Identify important entities of a property from classifieds
- **Part-of tree construction:** Structure them into a tree format the so-called *property tree*



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'
```

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

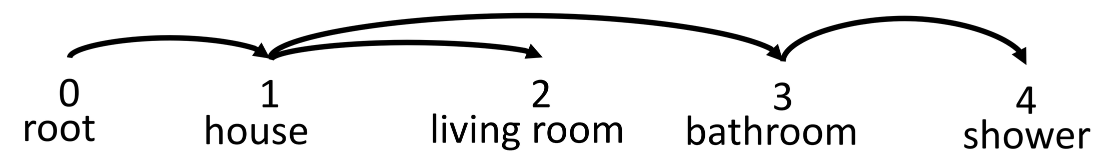


Figure: Projective dependencies

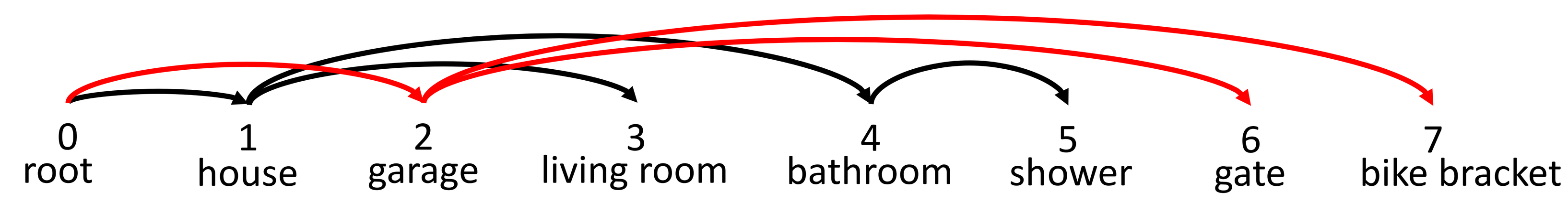


Figure: Non-projective dependencies

V. Step (1): Entity recognition

- Extract the entity boundaries
- Map the type of the entities
- CRF

Entity type	Description	Examples
property	The property.	bungalow, apartment
floor	A floor in a building.	ground floor
space	A room within the building.	bedroom, bathroom
subspace	A part of a room.	shower, toilet
field	An open space inside or outside the building.	bbq, garden
extra building	An additional building which is also part of the property.	garden house

VI. Entity recognition results

Entity type	TP	FP	FN	Precision	Recall	F_1
property	3170	1912	2217	0.62	0.59	0.61
floor	2685	515	529	0.84	0.84	0.84
space	11952	2053	2003	0.85	0.86	0.86
subspace	4338	575	1181	0.88	0.79	0.83
field	2083	700	718	0.75	0.74	0.75
extra building	253	34	143	0.88	0.64	0.74
Overall	24481	5789	6791	0.81	0.78	0.80

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))

Transition based (TB)

- Greedy transition-based parsing system
- Handles non-projective arcs using SWAP operation
- Predict the sequence of transitions to derive the dependency parse tree given a set of permissible actions

Locally trained model

- Binary classifier for part-of relations
- Classifier score reflects the likelihood of the part-of relation (between parent-child)
- **Threshold:** keep all edges with weights > threshold
- **Edmond:** Find maximum spanning tree starting from a fully connected directed graph

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 $y \in T(x)$:

$$P(y|x;\theta) = \frac{1}{Z(x;\theta)} \exp\left(\sum_{h,m \in y} \theta_{h,m}\right)$$

normalized by $Z(x;\theta)$ requires a summation over all $T(x)$

- MTT allows direct computation as $det(L(\theta))$ where L is the Laplacian matrix of the graph

VIII. Part-of tree construction results

	Model	TP	FP	FN	Precision	Recall	F_1
known entities	TB	14816	17368	17368	0.46	0.46	0.46
	Thresh.	15723	6365	16461	0.71	0.49	0.58
	Edm.	22058	10126	10126	0.69	0.69	0.69
	MTT	22361	9823	9823	0.70	0.70	**0.70**
full pipeline	TB	9677	19043	22507	0.34	0.30	0.32
	Thresh.	9309	9846	22965	0.49	0.29	0.36
	Edm.	12859	17417	19415	0.42	0.40	**0.41**
	MTT	12426	17850	19848	0.41	0.39	0.40

Conclusion & Future work

- Comparative study on the newly defined problem of extracting the structured description of real estate properties
- Divided the problem into the sub-problems of sequence labeling and non-projective dependency parsing

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

firstname.lastname@ugent.be