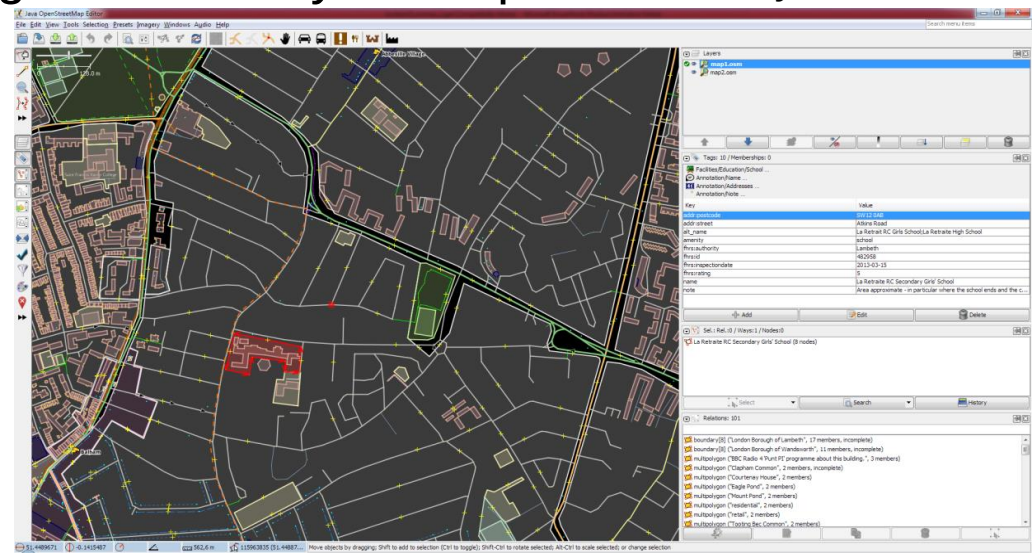
Learning to Classify Spatiotextual Entities in Maps

Giorgos Giannopoulos
Nikos Karagiannakis
Dimitrios Skoutas
Spiros Athanasiou

Institute for the
Management of Information
Systems
Athena Research Center

Motivation

- How to annotate large amounts of geospatial entities?
- OpenStreetMap:
  - Initiative for crowdsourcing map information from users
  - >2M users
  - >4B coordinates → Hundreds of Millions of entities
  - OSM Map Features
    - Hierarchy of categories for characterizing geospatial entities → more than 1400 categories
    - Organized in key-value pairs: Amenity => restaurant, Shop => toys



Problem

- How can we assist the annotation process?
  - Users discouraged by the **large amount of available categories**
  - Users may define an arbitrary category that:
    - might already exist in a slightly different form or
    - might have no actual semantic meaning in the context of OSM
  - Forbidding definition of new categories is not an option
    - Crowdsourcing “cancelled”
    - Some concepts might still be missing

- Solution?
  - Recommendation of categories
  - Based on similarities with already annotated spatiotextual entities

Method

- Train a classifier on available OSM data
  - Exploit already annotated spatiotextual entities as training items
  - Define problem specific training features that quantify the characteristics of the spatial entities
    - Spatial, textual, semantic.
  - Use OSM categories as training classes (labels)

- Recommendations
  - Classify each new entity using the trained model
  - Apply multi-label classification
  - Present the ranked list of classes as recommendations for annotation

Training features

- Spatial
  - Feature types
    - Geometry type (Point, LineString, Polygon, LinearRing, Circle, Rectangle)
    - Number of geometry points
    - Area of geometry
    - Mean edge length
    - Variance of edge lengths
  - Feature construction
    - Identify ranges of values for numerical features
    - Define equivalent boolean features corresponding to these ranges

Training features

- Textual
  - Extract textual description of entity's name
  - Filter out stopwords and infrequent words
  - What remains is mostly special meaning identifiers, e.g.:
    - avenue, church, school, park
  - Construct the TF vector for each entity

Training features

- Textual
  - Extract textual description of entity's name
  - Filter out stopwords and infrequent words
  - What remains is mostly special meaning identifiers, e.g.:
    - avenue, church, school, park
  - Construct the TF vector for each entity

- Semantic
  - Boolean features corresponding to OSM categories
  - Utilized only when an entity is already annotated

Classification algorithms I

- SVM
  - Maps the training entities into a multidimensional feature space
  - Aims at finding the optimal hyperplanes that discriminate the entities belonging to different categories
  - Output:
    - A model (weight vector) that maps the feature vector of a new, unannotated entity to a set of categories
    - Ranked list of recommended categories

Classification algorithms I

- SVM
  - Maps the training entities into a multidimensional feature space
  - Aims at finding the optimal hyperplanes that discriminate the entities belonging to different categories
  - Output:
    - A model (weight vector) that maps the feature vector of a new, unannotated entity to a set of categories
    - Ranked list of recommended categories

- k-Nearest Neighbor (kNN)
  - Compares new entity with all existing training entities
  - Recommends the categories that characterize the most similar training entities

Classification algorithms II

- clustering+SVM
  - Clusters training entities
  - Produces an SVM model for each cluster
  - Compares new entity with all existing training entities
    - Selects and weights the SVM models corresponding to clusters where the most similar training entities belong
    - Combines ranked lists from the selected SVM models to recommend categories

Classification algorithms II

- clustering+SVM
  - Clusters training entities
  - Produces an SVM model for each cluster
  - Compares new entity with all existing training entities
    - Selects and weights the SVM models corresponding to clusters where the most similar training entities belong
    - Combines ranked lists from the selected SVM models to recommend categories

- clustering+kNN
  - Clusters training entities
  - Creates cluster representative entities (centroids)
  - Compares new entity with centroids
  - Recommends the categories that characterize the most similar clusters

Evaluation dataset

- Two OSM datasets from Athens and London
- 5-fold cross validation
  - 5 equal-sized subsets 3-1-1 arrangement for train/validation/test

Statistics	Athens	London
Entities	20K	20K
Distinct classes	186	306
Classes per entity	1	1.1
Majority class #1 (Building/Building) frequency	7133	8036
Majority class #2 (ResidentialHighway/Footway) frequency	3850	1942
Majority class #3 (Footway/UnclassifiedHighway) frequency	1122	1083
Categories with frequency=1	0.2%	0.25%
Categories with frequency=2	0.33%	0.38%

Evaluation measures

- Precision of category recommendations
  - $P = \frac{\#correct\ category\ recommendations}{\#total\ category\ recommendations}$
  - P^1: A recommendation is considered correct if the recommended category with the **highest rank** from the recommendation model is indeed a category that characterizes the test geospatial entity
  - P^5: A recommendation is considered correct if **one of the five** recommended categories with the **highest rank** from the recommendation model is indeed a category that characterizes the test geospatial entity
  - P^{10}: Similarly

Evaluation results - Algorithms

- Comparison of 4 classification algorithms
  - Selected best configurations
  - Comparison with Majority class recommendation
- Results
  - SVM → Better for a few accurate recommendations
  - Clustering+kNN → Better for recommendation of more classes

Algorithms	Athens Test Set			London Test Set		
	P1	P5	P10	P1	P5	P10
SVM	**59.99**	81.61	89.56	**59.59**	**73.2**	80.02
kNN	59.98	71.57	73.45	58.81	70.15	72.39
Clustering+SVM	43.45	63.48	64.75	43.37	50.18	60.71
Clustering+kNN	59.15	**82.53**	**91.08**	51.67	71.44	**82.85**
Majority class recommendation	*35.7*	*68.3*	*75.8*	*40.2*	*63.1*	*73*

Evaluation results – Training features

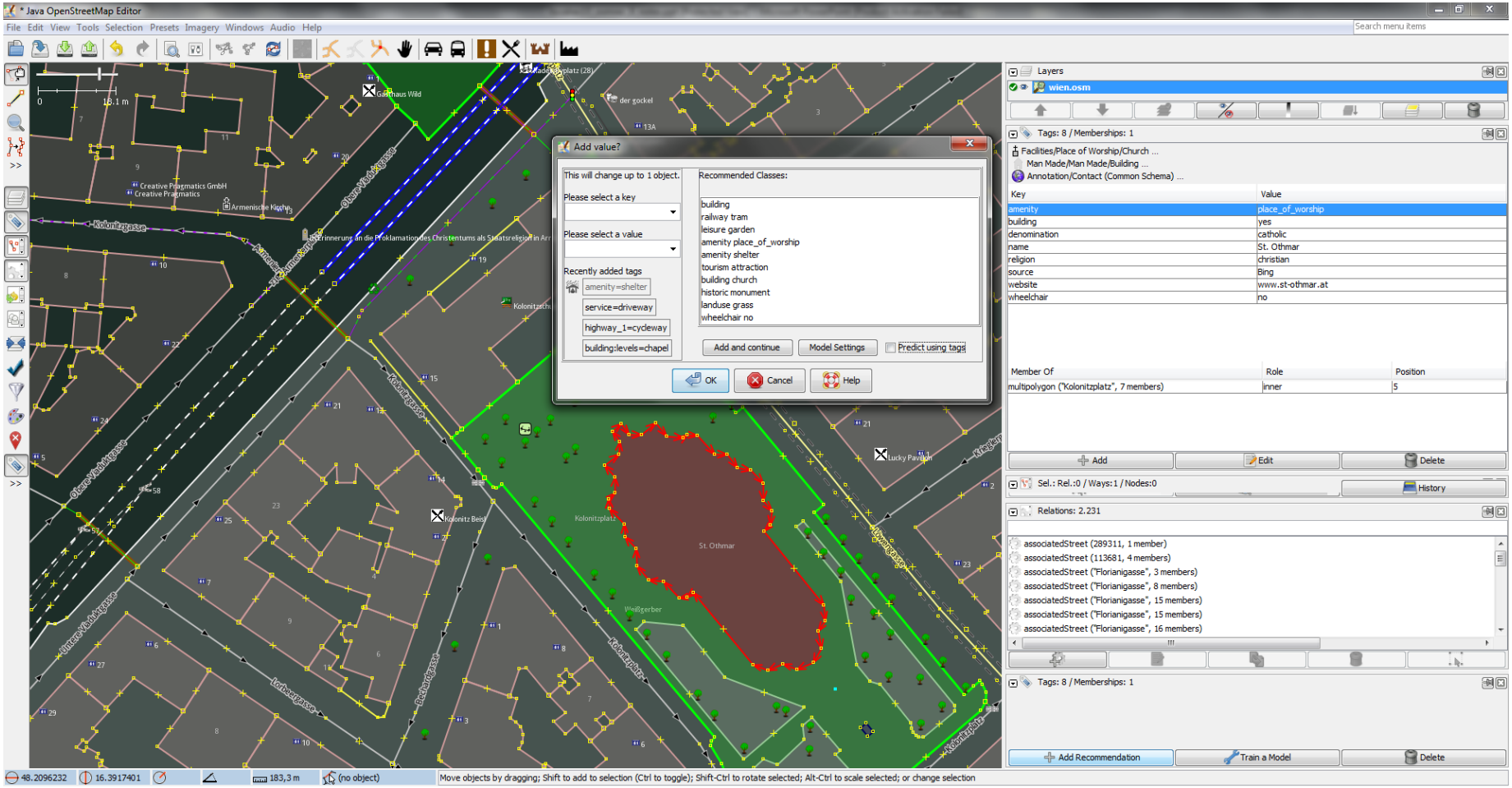
- General
  - Spatial properties >> Text
  - Plain spatial Type features often perform well on their own
- SVM/kNN
  - Spatial properties + Text
- clustering+kNN
  - Plain Type features
  - Spatial properties

Evaluation results – Datasets

- Athens better than London! Why?
  - 186 vs. 306 distinct classes → lower heterogeneity
  - "Outlier" classes:
    - 1 appearance: 20% vs. 25%
    - 2 appearences: 33% vs. 38%

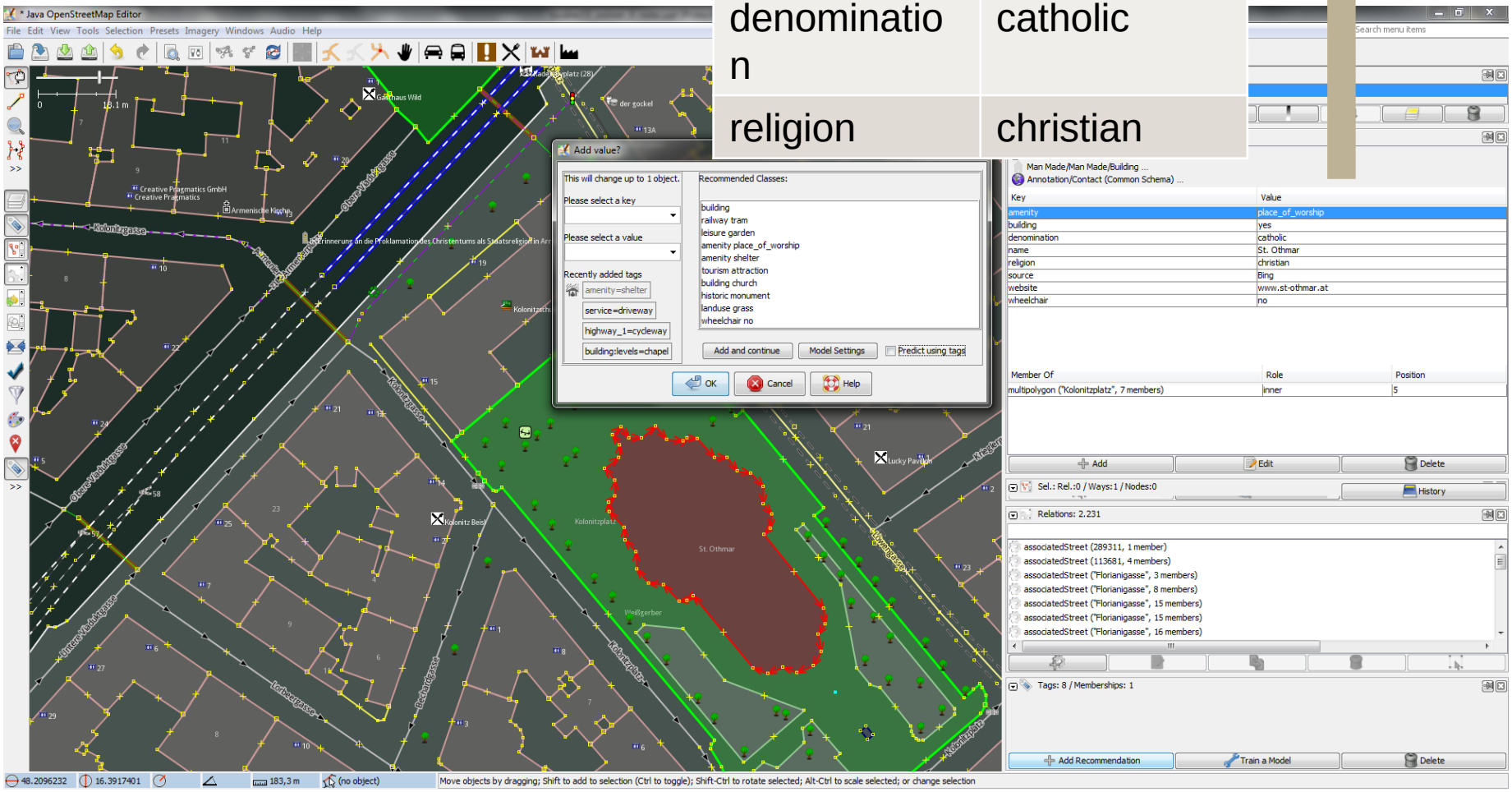
Algorithms	Athens Test Set			London Test Set		
	P1	P5	P10	P1	P5	P10
SVM	**59.99**	81.61	89.56	**59.59**	**73.2**	80.02
kNN	59.98	71.57	73.45	58.81	70.15	72.39
Clustering+SVM	43.45	63.48	64.75	43.37	50.18	60.71
Clustering+kNN	59.15	**82.53**	**91.08**	51.67	71.44	**82.85**
Majority class recommendation	*35.7*	*68.3*	*75.8*	*40.2*	*63.1*	*73*

OSMRec plugin



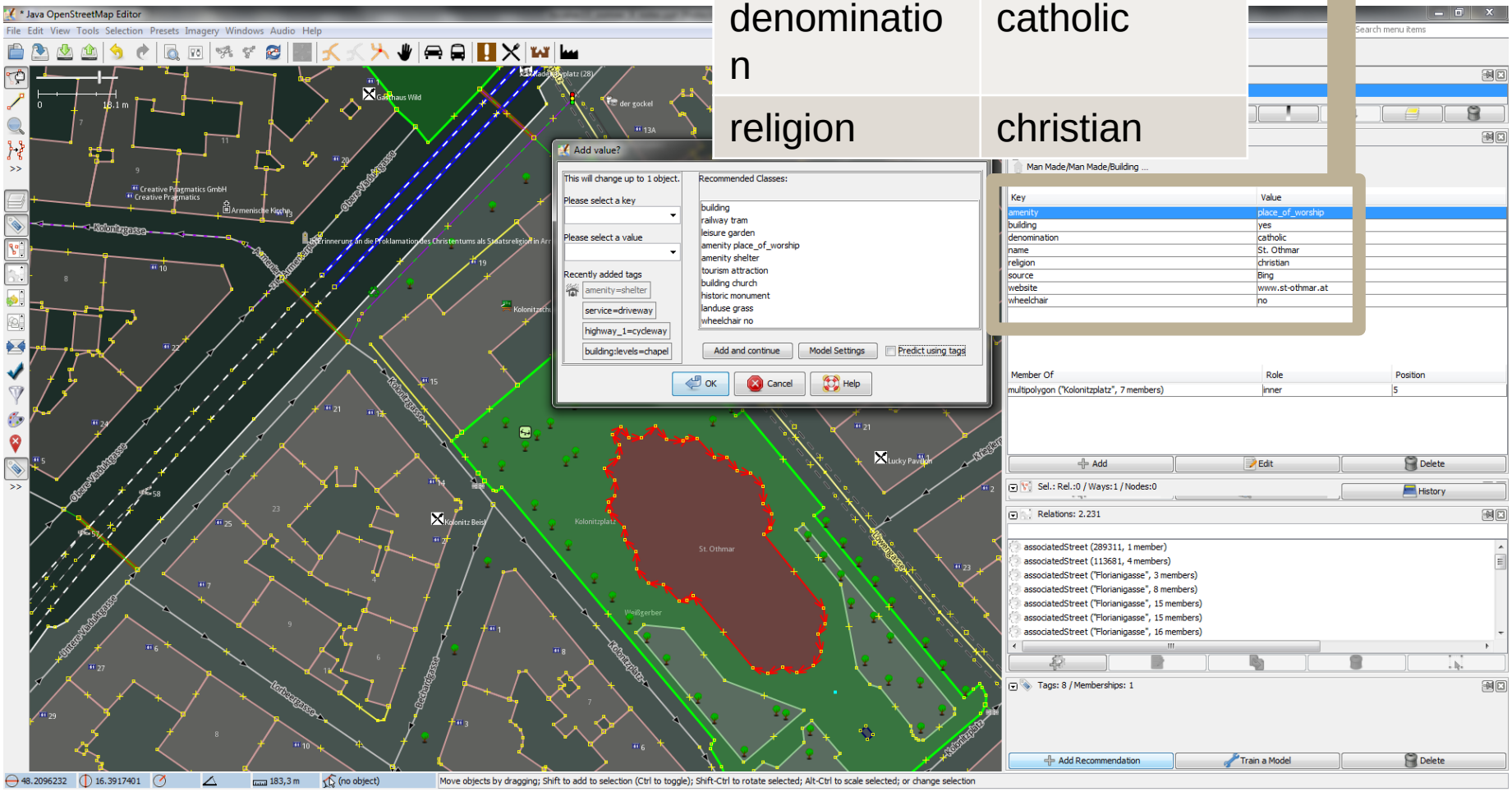
OSMRec plugin

Existing Annotations	
amenity	place of worship
denomination	catholic
religion	christian



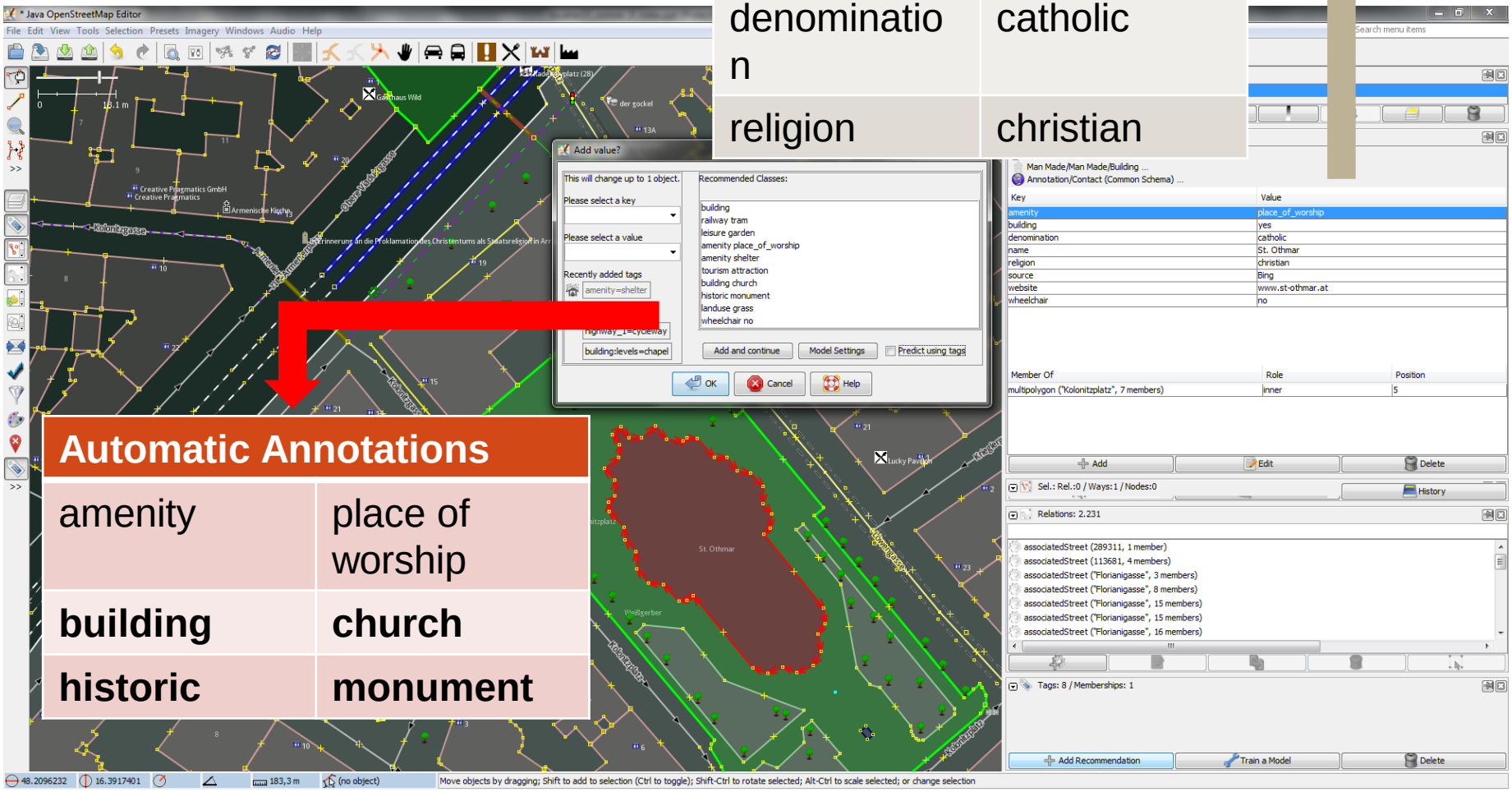
OSMRec plugin

Existing Annotations	
amenity	place of worship
denomination	catholic
religion	christian



OSMRec plugin

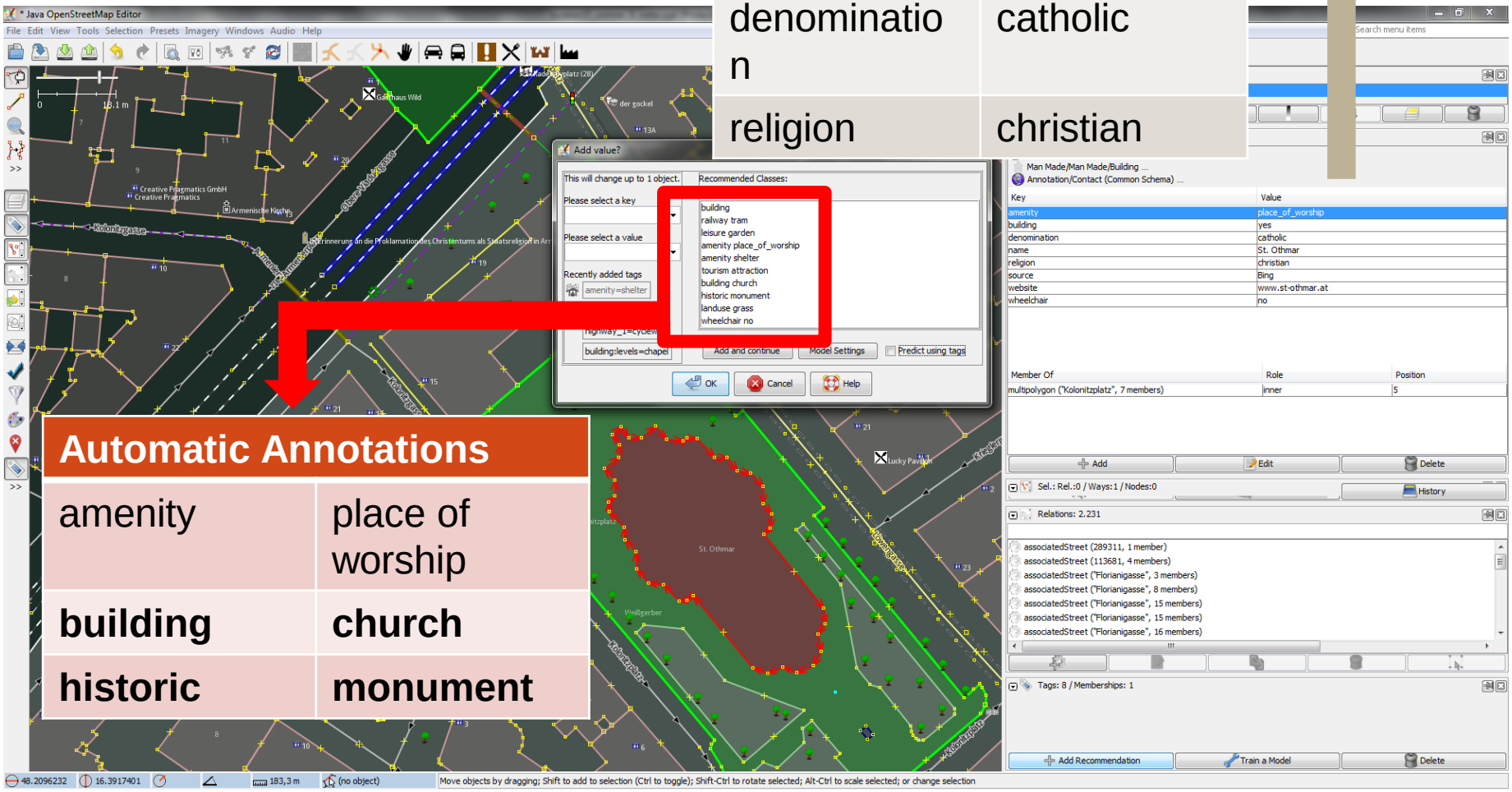
Existing Annotations	
amenity	place of worship
denomination	catholic
religion	christian



Automatic Annotations	
amenity	place of worship
building	**church**
historic	**monument**

OSMRec plugin

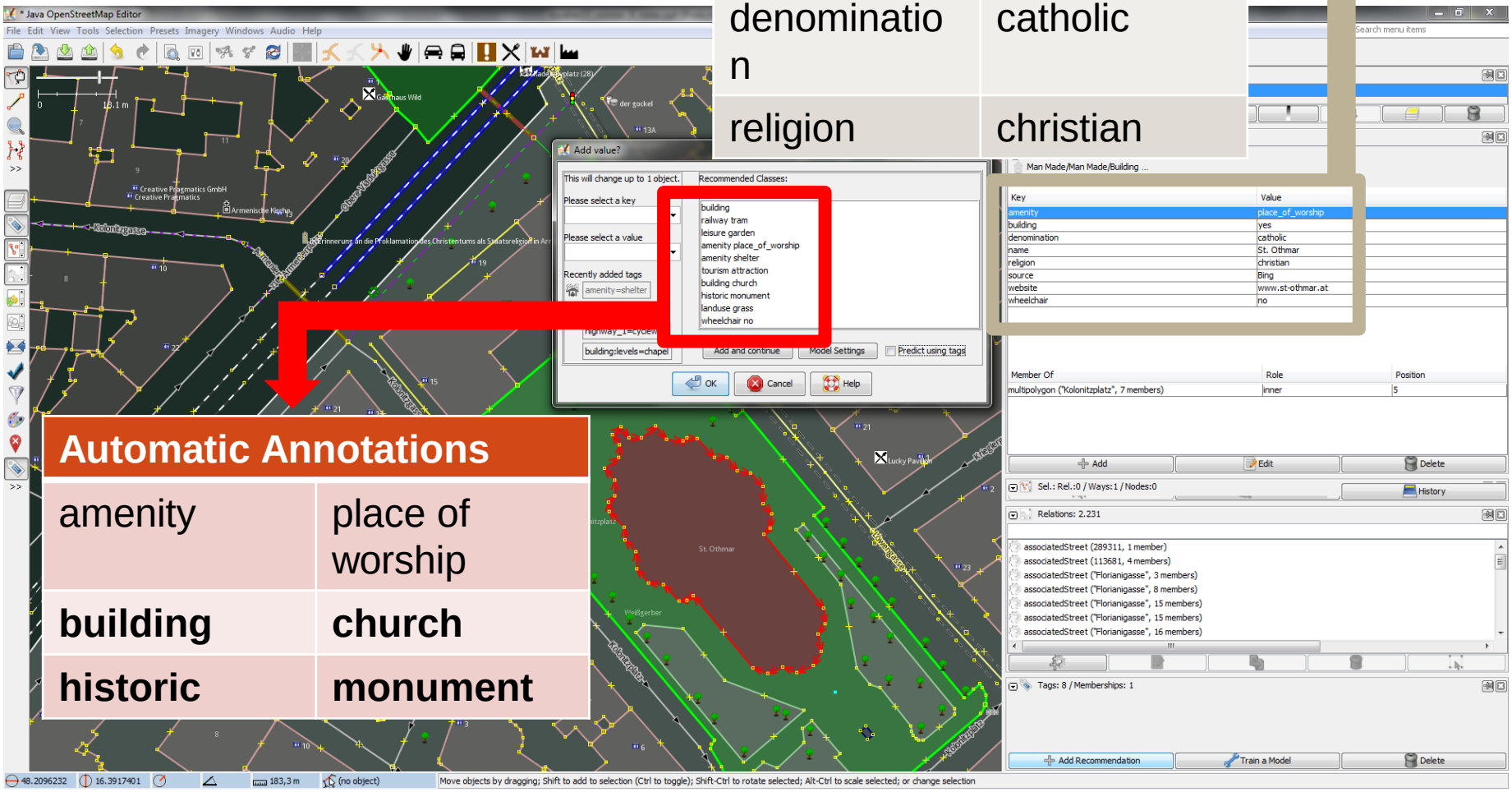
Existing Annotations	
amenity	place of worship
denomination	catholic
religion	christian



Automatic Annotations	
amenity	place of worship
building	**church**
historic	**monument**

OSMRec plugin

Existing Annotations	
amenity	place of worship
denomination	catholic
religion	christian



Automatic Annotations	
amenity	place of worship
building	**church**
historic	**monument**

Conclusion

- Contribution
  - Recommendation of annotation classes for spatiotextual entities
  - Proposed general enough training features
  - High precision → applicable in real-world use cases
  - Exploitation of millions of annotated entities from OSM
  - JOSM plugin

- Future work
  - “Neighborhood” features → seem to further increase precision
  - Diversification of recommendations

References

- OSMRec Tool for Automatic Recommendation of Categories on Spatial Entities in OpenStreetMap
  - Recsys'15 Demo
- Automatic Recommendations of Categories for Geospatial Entities
  - LocalRec'15 Short Paper
- GeoKnow Deliverable 3.3.1 Prototype for spatial knowledge aggregation
  - http://svn.aksw.org/projects/GeoKnow/Public/D3.3.1_Prototype_for_spatial_knowledge_aggregation.pdf
- GeoKnow Deliverable 3.2.2 Context sensitive spatial knowledge aggregation
  - http://svn.aksw.org/projects/GeoKnow/Public/D3.3.2_Context_sensitive_spatial_knowledge_aggregation.pdf
- GeoKnow/OSMRec on github
  - https://github.com/GeoKnow/OSMRec
- OpenstreetMap wiki page
  - http://wiki.openstreetmap.org/wiki/JOSM/Plugins/OSMRec
- OpenStreetMap svn
  - http://svn.openstreetmap.org/applications/editors/josm/plugins/OSMRecPlugin

?