Yale

Hierarchical Clustering for Unsupervised Topical Concept Taxonomy Generation

Wei Tai Ting

Yale College

LILY Lab

Introduction

Topic extraction and taxonomy generation are important tasks for understanding and organizing text corpora, and their results can also be useful inputs for information retrieval and semantic analysis. A topic taxonomy is a tree where each node is a topic with multiple terms, and where child nodes are hypernyms of the parent. However, many knowledge-base specific taxonomies are laboriously hand-crafted. Recent research has looked at automatically generating these taxonomies. We focus on improving an existing unsupervised system, TaxoGen, which uses top-down recursive hierarchical clustering to generate a taxonomy.

Approach

Dynamic k-selection for spherical k-means clustering

First, TaxoGen requires the user to specify a static *k* constant that determines the branching factor of the taxonomy. This is maintained as a constant throughout the taxonomy generation process. This leads to two major concerns. How can the user decide *k* before running the program? What if the "best" taxonomy structure requires different values of *k* at different levels of the taxonomy? Motivated by these questions, we experiment with three traditional scoring metrics to select *k* dynamically at each topic node: the Calinski-Harabasz (CH) Index, the Davies-Bouldin (DB) Index, and the cluster silhouette scores. Instead of specifying a static *k* for all levels of the taxonomy, the user now specifies a ceiling (*max_k*, inclusive) and floor (*min_k*, inclusive, at least 2) for *k* and specifies the metric to score the choice of *k*. We perform spherical *k*-means clustering for *k* from *min_k* to *max_k* and pick the *k* with the highest score from the chosen metric.

All three metrics represent different mathematical formalizations of what a "good" clustering is. Metric scores improve as inter-cluster distance and intra-cluster density increase, corresponding to how a "good" clustering has dense, well-separated clusters.

Reducing document and keyword loss

Moreover, we also encountered some problems with TaxoGen's code when using sparse datasets in practice (the AAN papers dataset). The taxonomy terminated relatively early without recurring to the maximum depth specified for certain branches, as the number of keywords dropped drastically between each level of the taxonomy. The pre-existing TaxoGen implementation uses hard partitioning to allocate documents to each topic cluster, that is, each document is mapped to only one cluster. (Figure 2) TaxoGen then trains local embeddings when recurring downwards to that specific cluster. Since TaxoGen discards keywords in a cluster if they do not exist in the locally trained embeddings, this means that many keywords are discarded because of hard partitioning. Our implementation replaces hard partitioning with soft partitioning: as long as a document contains a keyword in the cluster, it is included in that cluster.

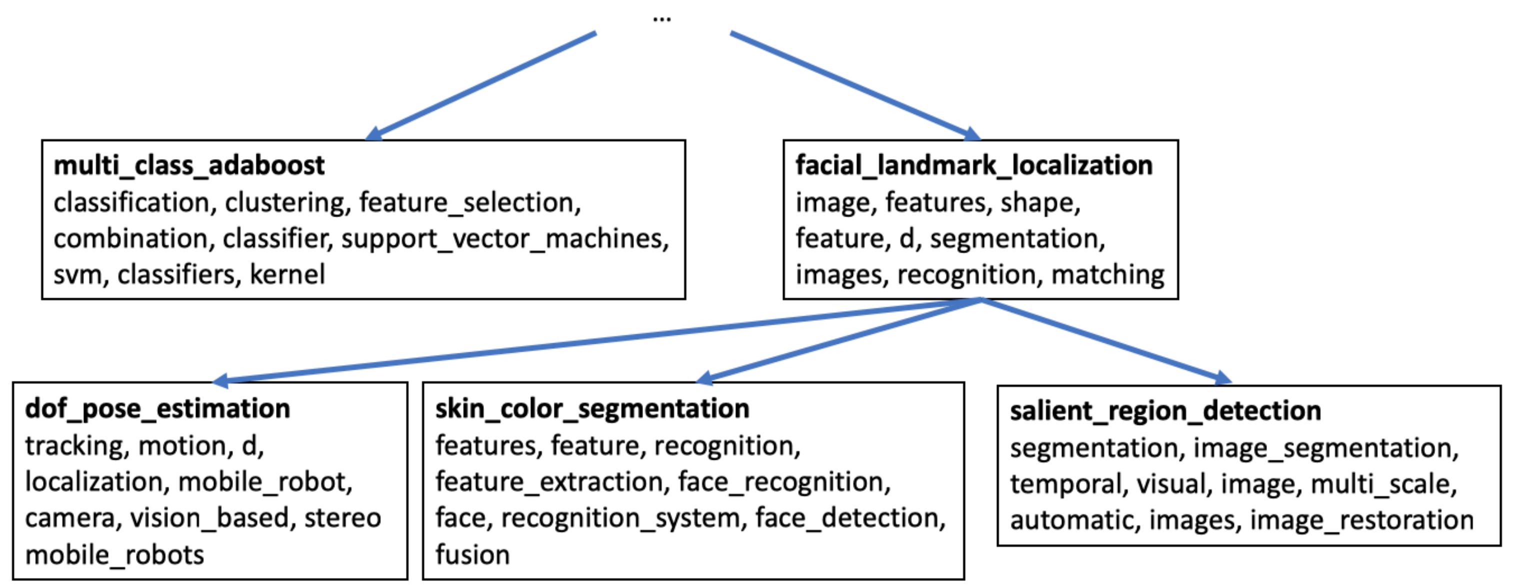
...
multi_class_adaboost
classification, clustering, feature_selection, combination, classifier, support_vector_machines, svm, classifiers, kernel
facial_landmark_localization
image, features, shape, feature, d, segmentation, images, recognition, matching
dof_pose_estimation
tracking, motion, d, localization, mobile_robot, camera, vision_based, stereo mobile_robots
skin_color_segmentation
features, feature, recognition, feature_extraction, face_recognition, face, recognition_system, face_detection, fusion
salient_region_detection
segmentation, image_segmentation, temporal, visual, image, multi_scale, automatic, images, image_restoration

Figure 1: Section of taxonomy from our implementation using the DB Index and the DBLP dataset

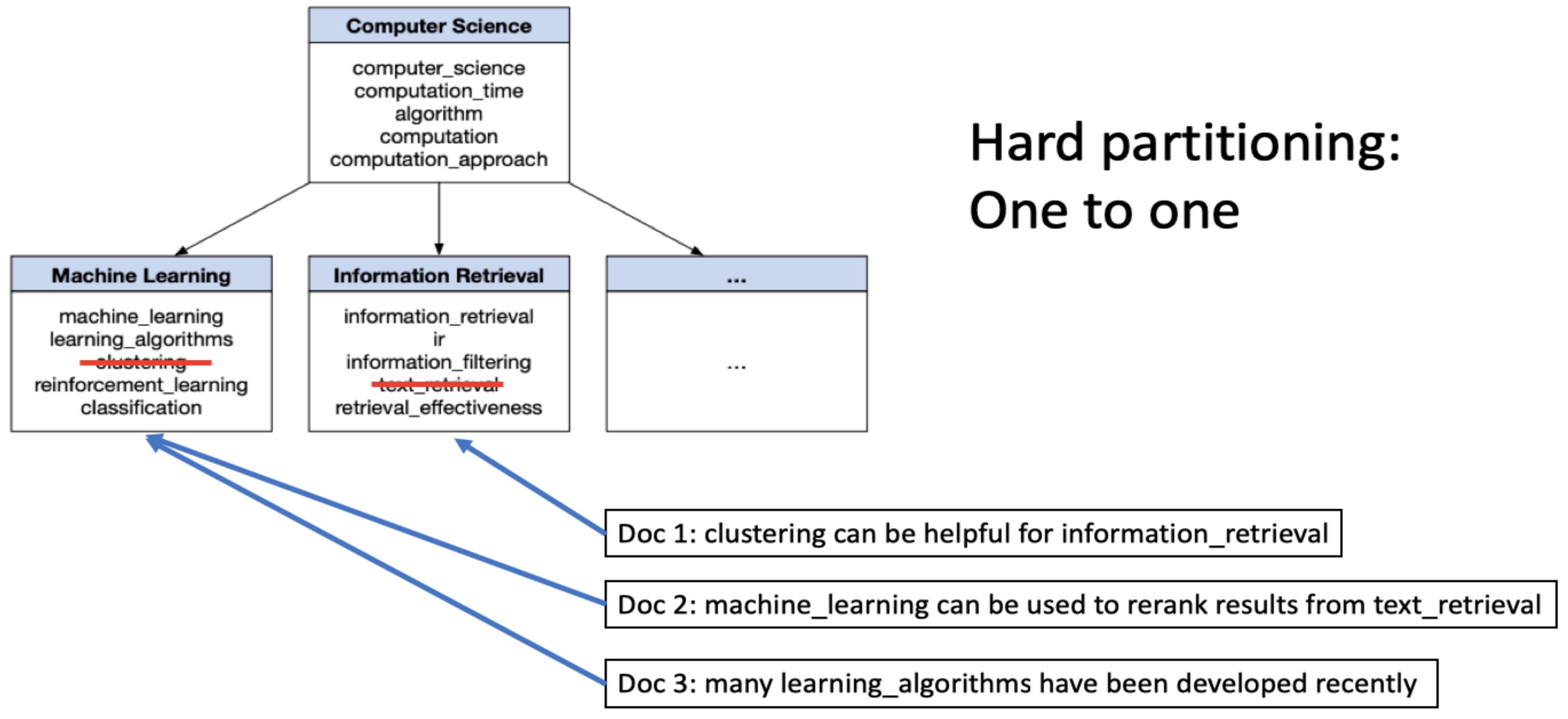

Figure 2: Hard partitioning used in pre-existing TaxoGen implementation

	Static k = 5	CH Index	DB Index	Silhouette Coefficient
% of valid parent-child pairs	**48.4**	43.8	39.7	44.0
# of parent-child pairs	31	53	**145**	66

Table 1. Results from AANP dataset

	Static k (k = 5)	CH Index	DB Index	Silhouette Coefficient
% of valid parent-child pairs	74.0	67.4	67.4	**75.5**
# of parent-child pairs	**117**	28	62	52

Table 2. Results from DBLP dataset

The pre-existing TaxoGen implementation also does not use all keywords in a cluster to select documents. It first uses adaptive clustering to eliminate keywords that are not "representative" of the cluster below a certain threshold (this is specified in the *filter_thre* parameter). Then, from the remaining keywords, it picks the top *n_expand* keywords (a user-specified parameter) which are the most similar to the cluster topic and then uses these keywords to retrieve documents to consider to be included in the cluster. We remove both constraints to increase document recall in our implementation because of the problem of corpus sparsity. All cluster keywords are used to decide which documents belong in a given cluster.

Results

We randomly sampled up to 50 parent-child pairs from each taxonomy and evaluated if each pair is valid. Unfortunately, due to time constraints, we used only one annotator. The results are produced below. While a dynamic *k*-selection policy decreased the validity for the sparser AAN Papers dataset slightly, it increased the size of the generated taxonomy significantly. In the case of the DBLP dataset, the silhouette coefficient *k*-selection policy actually had a slightly higher percentage of valid parent-child pairs compared to static *k*. The difference in the number of parent-child pairs in this case might not be that significant because the results from the experiment came from a max taxonomy depth that was set to 4, and the results indicated that further recursion was possible.

Conclusion

Our results show that dynamically selecting *k* results in a slight decrease in accuracy. However, it is possible that this is the tradeoff of automating hyperparameter search in an unsupervised context, where the correctness of the hyperparameter chosen is not immediately apparent until the human evaluation stage. Future work can be done on making the evaluation process more comprehensive. This could be done by evaluating all taxonomy pairs and to evaluate topic cohesiveness—whether all the terms in the topic cluster belong in the same cluster.

Since our changes to TaxoGen's implementation have been to tackle the problem of sparse datasets, it might be possible to use the implementation to generate concept graphs for a given book. Another possible improvement to our current system would be possible to parallelize the hierarchical clustering operation by using a breadth first approach where each thread processes a node. Thus, with a sufficient number of threads, the runtime would be bounded by the maximum depth of the taxonomy rather than the number of nodes.

Acknowledgements

I thank Professor Dragomir Radev, Alex Fabbri and other members of the LILY Lab for helping me with this project and tackling relevant research.