Human-Guided Ontology Learning

Hui Yang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
huiyang@cs.cmu.edu

Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
callan@cs.cmu.edu

ABSTRACT
This paper leverages human knowledge and understanding in machine learning algorithms for constructing ontologies. Ontology construction is a highly subjective task where a human user builds a data model which represents a set of concepts within a domain and the relationships between those concepts. Personal preferences have crucial impact on manually-built ontologies, however are inadequately captured by traditional supervised machine learning approach. This paper proposes a human-guided machine learning approach, which incorporates periodical manual guidance into a supervised clustering algorithm, for the task of ontology construction. A user study demonstrates that guided machine learning is able to generate ontologies with manually-built quality and less costs. It also shows that periodical manual guidance successfully directs machine learning towards personal preferences.

INTRODUCTION
Ontology construction, or ontology learning, is an important task in Artificial Intelligence, Semantic Web and Knowledge Management. It is the process of building an ontology, a data model that represents a set of concepts within a domain and the relationships between those concepts. An ontology is about the given corpus or domain, identifies and often organizes the concepts into a tree-structured hierarchy. In most cases, ontology learning is highly subjective and task-specific. For example, when writing a literature review for human computer interaction (HCI), we may crawl the Internet for the relevant materials, sort through various documents, identify important concepts and milestones in the literature, find the important relationships between them, and organize them based on the relationships. Note that different person will have different ways to define "what is an important concept or milestone" and "what is an important relationship", and hence results in different ontologies for HCI. In general, personal preferences show crucial impact on manually-built ontologies.

In the context of ontology construction, personal preferences are represented as periodical manual guidance in guided machine learning, which combines the strengths of both human expertise and machine learning to build ontologies. In particular, human users teach the system to create a personalized, task-specific ontology by providing appropriate scaffolding, a concept in the Situated Learning Theory referring to the supports provided by a teacher to help a student achieve tasks which are not able to accomplish independently, while the system learns from such manual guidance, adjusts the learning process with appropriate changes and produces learned results by following the guidance. The teaching and learning actions occur alternatively at each learning cycle and the entire process continues until a human-satisfied ontology is built. There are two major questions for research on constructing ontologies by guided machine learning and they are:

(1) Can a guided machine learning approach produce ontologies with the same quality as manually-built ones?

(2) Can a guided machine learning approach learn from individual users and capture the distinctions among their personal preferences?

To answer the above questions, this paper studies the effects of guided machine learning on ontology construction. In particular, it employs a supervised clustering algorithm, which learns distance metrics for concept pairs in an ontology, in a guided bottom-up hierarchical clustering framework. At each human-computer interaction cycle, cluster partitions from human guidance, are taken as the training data, from which a distance metric is learned. The distance metric is then used in a flat clustering algorithm to create clusters at the higher level. A user study demonstrates that guided machine learning is able to generate ontologies with manually-built quality and manual guidance successfully directs machine learning towards personal preferences.

A GUIDED HIERARCHICAL CLUSTERING FRAMEWORK
In this section, we model the process of ontology construction as a guided machine learning framework. Given the fact that most ontologies are hierarchies in nature, we employ hierarchical clustering as the main guided learning framework, in particular, a bottom-up hierarchical clustering framework. Algorithm 1 gives the pseudo-codes for the guided hierarchical clustering algorithm. Starting from the bottom, the process builds up the ontology level by level by learning a new distance metric from the current level and applying it to the higher level. At each iteration, any flat clustering algorithm can be used to construct concept groups. The flat clustering algorithm used in this work is K-medoids [2]. We adopt Gap statistics [3] to estimate the number of clusters.

After concepts are clustered by K-medoids, if the system is in its interactive mode, it displays the learned ontology on the User Interface and waits for manual guidance. Users can interact with the system via a tool called OntoCop (**Onto**logy

Algorithm 1: Guided Hierarchical Clustering

while not satisfied or not all concepts connected in a tree
  construct groups for level i by flat clustering;
  if in interactive mode
    wait for manual guidance;
  learn distance metric function from level i;
  predict distance scores for level $i+1$;
  $i \leftarrow i+1$;
Output the tree

Construction **P**anal). Users are able to add, delete, modify concepts, drag & drop concepts around and group them accordingly. Users can also search and view the documents relevant to a concept for a better understanding of the domain knowledge when they are making decisions. When they are done with modifications to the concepts, they can upload the hierarchy to the server, which learns from the user modifications, predicts new distance scores for unorganized concepts and runs K-medoids to cluster them and returns the new hierarchy to the user.

In an uploaded hierarchy, there are many concept groups, each contains a parent concept and a group of child concepts. We call such concept groups "*ontology fragments*". From an uploaded hierarchy, which usually is a partial ontology, we decompose it into ontology fragments and use them as manual guidance in the learning process. In the proposed bottom-up approach, the grouping information in ontology fragments at the lower levels are used to estimate a distance metric function, which then predicts the distance scores for concepts at the higher levels.

INCORPORATING MANUAL GUIDANCE

In Figure 1, the ontology fragments suggest that (child, maker) is close since they are in the same group, (sport hunter, trophy hunter) is also close, (sea ice habitat, child) may be far away since they are in different groups. The goal is to find a mapping from such grouping information to their semantic distances and then use the mapping function to predict the semantic distances for ungrouped concept pairs such as (habitat, person) and (habitat, territory). The mapping is required to give reasonable scores to concept pairs such that (habitat, territory) is closer than (habitat, person).

We propose a supervised clustering algorithm based on distance metric learning [4]. In particular, the ontology construction problem is modelled such that at each time, a set of concepts $\mathbf{x}^{(i)}$ on the ith level of the ontology hierarchy is under consideration. Another training input is a distance matrix $\mathbf{y}^{(i)}$. An entry of this matrix which corresponding to concept $x_j^{(i)}$ and $x_k^{(i)}$ is $y_{jk}^{(i)} \in \{1, 0\}$, where $y_{jk}^{(i)} = 0$, if $x_j^{(i)}$ and $x_k^{(i)}$ in the same cluster; 1, otherwise. The training data consists n levels of training concepts $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ..., $\mathbf{x}^{(n)}$, each with $|\mathbf{x}^{(1)}|$, $|\mathbf{x}^{(2)}|$, ..., $|\mathbf{x}^{(n)}|$ concepts. Each set $\mathbf{x}^{(i)}$ represents a set of concepts at the level indexed by i. For each set of training data, the correct partition (clustering) are given via distance matrices $\mathbf{y}^{(1)}$, $\mathbf{y}^{(2)}$, ..., $\mathbf{y}^{(n)}$.

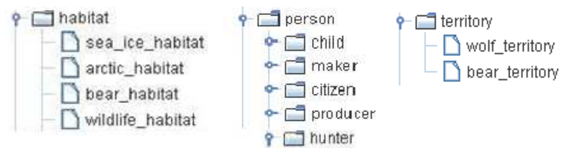

Figure 1. Ontology Fragments

In the distance matrix, within-cluster distance is defined as 0 and between-cluster distance is defined as 1. From the training distance matrix, we would like to learn a good pairwise distance metric function which best preserves the regularity in the training distance matrix. In our work, the estimated pairwise distance metric function is represented as a Mahalanobis distance [4].

$$d_{jk} = \sqrt{||x_j - x_k||^T A ||x_j - x_k||}$$

Theoretically, the parameter estimation problem in our settings is to get A such that the expected loss is minimized. The loss function is minimized through minimizing the squared errors. The optimization function is then defined as :

$$\min_A \sum_{j=1}^{|\mathbf{x}^{(i)}|} \sum_{k=1}^{|\mathbf{x}^{(i)}|} (y_{jk}^{(i)} - \sqrt{\Phi(x_j^{(i)}, x_k^{(i)})^T A \Phi(x_j^{(i)}, x_k^{(i)})})^2$$

$$\text{subject to } A \succeq 0$$

where $\Phi(x_j^{(i)}, x_k^{(i)})$ represents a set of pairwise underlying feature functions, where each feature function is $\phi_d : (x_j^{(i)}, x_k^{(i)}) \mapsto r \in \Re$ with d=1,...,$|\Phi|$. The underlying feature functions evaluates the relationship between $(x_j^{(i)}, x_k^{(i)})$ from various aspects. The next section will give more details about the feature functions. *A* is a parameter matrix, which weighs the underlying distance feature functions.

Given the learned parameter matrix *A*, it is easy to generate distance metric for any pair of unmeasured concepts. By calculating the distance for each concept pairs, we obtain the entries in a new distance matrix $\hat{\mathbf{y}}^{(i+1)}$, which contains the distance scores for concepts at the $(i+1)^{th}$ level. Note that previously they were unmeasured and unorganized. The scores are then used to produce partitions.

In a nutshell, in the guided hierarchical clustering framework, the learner requests for manual guidance at each learning cycle, and adjusts the learning of the distance metric accordingly. In particular, by taking into account a user's modification to the ontology, the system learns from his/her personalized grouping of concepts.

FEATURES

The distance metric learning process models a distance metric as a function of some underlying feature functions, where each feature function is a measurement of how distant two concepts are. Features used in this work are a balanced mixture of *statistical, contextual* and *knowledge-based* distance functions. *Statistical Features* are basically various forms of term (co-)occurrences in corpora, which are statistical evidence of how distant two concepts are. In particular, we use raw and log frequencies of term occurrences for a single concept, which is at the diagonal entries of a distance matrix, and raw and log frequencies of term co-occurrences for a concept pair. *Contextual Features* measure the concept similarity based on the distributional hypothesis. There are two kinds of contextual features used in this work. The first measures the number of word overlaps between the subjects/objects of verb predicates where each of the two concepts is the object/subject. For example, for concepts "polar bear" and "seal", habitat(polar bear, arctic ice) and habitat(seal, sea ice) are two corresponding verb predicates, where the two concepts are the subjects. The word overlap between the objects is 1 ("ice" in this case). The second measures the number of word overlaps between noun or adjective modifiers in front of two concepts. For example, the overlap between modifiers in "high blood pressure" and "peer pressure" is 0. *Knowledge-based Feature* used in this work is the number of word overlaps between the Web definitions of two concepts, for instance, for a concept pair (habitat, arctic sea) we issue query "define:habitat" and "define:arctic sea" to Google search engine. The Web definitions are then compared and the feature function outputs the number of word overlap after removing the stopwords. Note that Web definitions for concepts are mainly from Wordnet. All values from the above feature functions are normalized into $[0, 1]$ by dividing by the maximum possible values.

A USER STUDY AND EXPERIMENTAL RESULTS

To evaluate the system performance and answer the two questions posed at the beginning of the paper, a user study has been conducted for the task of ontology construction. The task is defined in the domain of public comments, where administrative agencies of the U.S. government seek comments from stakeholders and the public to issue draft versions of proposed regulations and respond in the final rule to substantive issues. The situation given in the evaluation is that the agencies need to organize the relevant materials into rule-specific ontologies based on the their actual needs.

We collaborated with an independent coding lab to conduct the user evaluation. Twelve professional coders familiar with the problem domain participated in the experiments. They were divided into two groups, four for the manual group and eight for the interactive group. Users in the manual group were asked to construct ontology with the concept candidates produced by the system in a bottom-up fashion until they felt satisfied with their work or reaching a 90-minute limit (which is carefully evaluated by the experiment designers). The interactive group were asked to work interactively with the system until they felt satisfied with the work or reaching a 90-minute limit. Each user in the interactive group worked on organizing the concept candidates for a few minutes, then uploaded the modified hierarchy to the system; then the system learned from user feedback, produced a new hierarchy and returned it to the user. It is a user's decision to continue modifying the ontology and teaching the system to learn or stop. Both groups used the same editing tool provided in OntoCop, such as deleting, adding a node, dragging and dropping a node, promoting a node to the higher level, undoing previous actions, etc. The set of concept candidates given to both groups were the same.

Table 1. Intercoder Agreements on Parent-Child Pairs

	manual-manual	manual-interactive	t	p
wolf	0.55	0.55	0	0.5
polar bear	0.44	0.46	0.21	0.42
mercury	0.61	0.51	1.89	0.03

There are four public comment data sets used in the experiments, namely "toxic release inventory (tri)" (Docket id: USEPA-TRI-2005-0073),"wolf" (USEPA-RIN-1018-AU53), "polar bear" (USDOI-FWS-2007-0008),"mercury"(USEPA-OAR-2002-0056). The vocabulary sizes of each dataset are 12,838, 51,938, 67,110 and 102,503, which result in 248, 795, 351, and 1084 concept candidates for each dataset respectively. Among these four datasets, "tri" is the one with the smallest vocabulary and used for tool training for both manual and interactive users. The experimental results generated on "wolf", "polar bear" and "mercury" datasets are reported in the following sections.

For a given ontology, a list of all parent-child pairs in the hierarchy are generated. Performance metrics for parent-child pairs measure whether a concept is assigned to the correct parent. In section we use the intercoder agreement as the performance metric while in section we use the F3-measure.

Quality of Constructed Ontologies

This experiment investigates whether the proposed guided machine learning approach is able to produce ontologies with the same quality as manually built ones. We compare the intercoder agreement between two manual runs and that between one manual and one interactive run in this experiment. The intercoder agreement measured by Cohen's Kappa between two manual runs is averaged over 4x3=12 pairs of manual-manual runs. The intercoder agreement between manual and interactive runs is averaged over 4x8=32 pairs of manual-interactive runs. Table 1 shows the averaged intercoder agreements and the significance test results for parent-child pairs and sibling pairs respectively. We can see that both the intercoder agreement between manually built ontologies and that between manual-interactive runs are within the range of 0.44 to 0.61, which indicates moderate agreement. We also observe that manual-interactive intercoder agreement is comparable with manual-manual intercoder agreement, which indicates that the guided machine learning approach is able to produce the same quality ontologies as humans do. A series of one-tailed t-tests also confirm it. Almost all significant test results are not significant, $t < 2$ and $p > 0.01$, which show no statistical significant differences from manually-built ontologies and interactively-built ontologies. The results demonstrate that guided machine learning is able to produce the same quality ontologies as humans do.

Costs of Constructing Ontologies

Table 2. Average Manual Editing Costs

	add	delete	move	name change	undo	total
manual	56.25	200	2806.75	70.25	19	3152.25
interactive	20.17	129	1693.17	39.5	7.83	1889.67

Table 3. Ontology Construction Duration

	wolf	polar bear	mercury	**average**
manual	1:24	1:22	1:33	**1:27**
interactive	1:06(0:33)	0:34(0:29)	1:05(0:30)	**0:55(0:31)**

This experiment investigates the construction costs of taking manual or interactive approach. We compare the construction logs for users from both manual and interactive groups. Table 2 shows the number of manual editings of building ontologies for three datasets. The editings include adding a (child or sibling) concept, moving a concept by drag & drop, deleting a concept, changing name for a concept and undoing previous actions. In total, interactive users use 40% less editing actions to produce the same quality ontologies as manual users do. A one-tailed t-test shows a significant reduction, t=10 and $p < 0.001$, of interactive runs in editing costs as compared to manual runs. It demonstrates that guided machine learning is significantly more cost effective than manual work.

We also compare the ontology construction duration. Table 3 shows the actual time needed to construct an ontology for both manual and interactive runs. It also shows the time in part spent by human users in the interactive runs in the brackets. In general, interactive runs save 30 to 60 minutes for building one ontology. Within an interactive run, a human user only needs to spend 31 minutes in average to construct an ontology, which is 64% less than 1 hour and 27 minutes in a manual run. It shows that guided machine learning greatly saves a human user's time to construct an ontology.

Learning from Personal Preferences

This experiment investigates the system's ability to learn from personal preferences from different users and eventually fulfil their personal needs. Figure 2 shows the changes of average F3-measure for parent-child pairs over six learning cycles. The x-axis are the learning cycles for each dataset. The y-axis indicates the averaged F3-measures.

Results for both interactive and manual users before and after each learning cycle are shown. For manual users, we use their partially constructed ontologies with 20%, 40%, 60%, and 80% modifications in the editing log and plot the F3-measures. Each individual's partial ontologies are compared with his/her own finalized ontology. The F3-measure is averaged over the 4 manual users. For interactive users, we take the ontologies that uploaded by them each time to the server and plot the F3-measures of each uploaded version and the learned ontology afterwards against his/her own finalized ontology. The F3-measure for the interactive group is averaged over the 8 members.

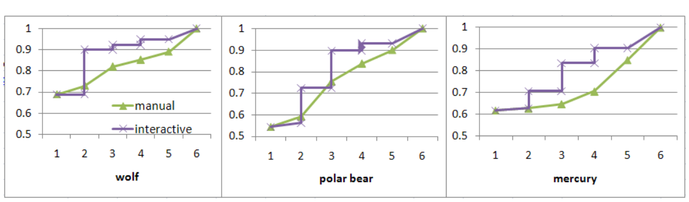

Figure 2. F3 for Parent-Child Pairs over Cycles

In Figure 2, F3-measures for both manual and interactive groups converge to 1 at the end of the learning process since it is a personalized task and each individual's finalized ontology is used as the gold standard. For interactive users, we notice an obvious performance gain between an uploaded ontology and the ontology learned automatically from it. Moreover, comparing the performances of interactive and manual users, we notice that the learning curve of the interactive users are steeper than that of the manual users. It indicates that the guided machine learning approach not only learns from personal preferences but also helps interactive users move faster towards their personal satisfaction levels.

CONCLUSIONS

This paper has shown a guided machine learning approach for the task of ontology construction. By incorporating periodical manual guidance into a distance learning algorithm in a hierarchical supervised clustering framework, it takes into account human expertise in a real-time interactive ontology construction process. A user study and experimental results demonstrate positive answers to the two questions posed on the effects of guided machine learning for ontology construction: guided machine learning is able to generate ontologies with manually-built quality and manual guidance has positive effects on directing machine learning towards personal preferences. Moreover, an analysis of the construction costs and duration shows that guided machine learning is significantly more cost effective and efficient than the manual work. Given that both guided machine learning and manual work produce ontologies with the same quality, the former becomes more attractive. Further, the results show that guided machine learning not only learns from personal preferences but also accelerates the process of ontology construction towards the personal satisfaction levels. This is very encouraging for the proposed framework.

REFERENCES

1. Z. Harris. Distributional structure. In *Word, 10(23): 146-162s*, 1954.
2. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.
3. R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. In *Tech. Rep. 208, Dept. of Statistics, Stanford University*, 2000.
4. E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 2002.