OntoCop: Constructing Ontologies for Public Comments

Hui Yang and Jamie Callan, Carnegie Mellon University

U.S. law defines a process known as *Notice and Comment Rulemaking* that requires regulatory agencies to seek comment from the public before establishing new regulations. Regulatory agencies are also expected to demonstrate that the final regulation addresses all substantive issues raised by the public. Most proposed regulations attract few comments from the public, but each year a few regulations attract tens of thousands or hundreds of thousands of comments. When comment volume is high, most comments are form letters and modified form letters, which are not difficult to process [1], but there are still tens of thousands of unique comments that address a variety of issues. There may also be political or public pressure for the agency to issue regulations quickly. In these cases, regulatory agencies and other interested parties need tools that help them to quickly make sense of public comments.

Browsing hierarchies such as the Yahoo! Directory are a popular method of quickly discovering the "lay of the land" in a large text corpus. By associating documents with topics and concepts in a hierarchy, the information space is structured and partitioned into smaller spaces that are easy to understand and navigate. Regulation-specific browsing hierarchies makes it easier to understand the range of issues that were raised by the public, and enables "drilling down" into comments that discuss particular issues, which enables agencies and policymakers to be more responsive to the public's concerns. Information analysis tools that enable comments to be analyzed quickly and efficiently also increase the likelihood that agency staff will process public comments *within the agency*, rather than subcontracting the job to outside companies that may have less topic expertise.

There is a large literature on automatic formation of concept hierarchies (also called "taxonomies" and "ontologies"), most of it focused on creation of general purpose concept hierarchies. Most prior research exploited simple lexical-syntactic patterns [2] and/or contextual information [3] in combination with a lexical resource such as WordNet [4]. However, when the corpus is 'small' and focused on a narrow set of issues, important concepts and relations may be rare or missing in general-purpose lexical resources and/or corpora. For example, in public comments submitted about a proposed EPA regulation *(USEPA-TRI-2005-0073)*, *RHOM* is a *chemical company*; however, this relation is neither available in WordNet, nor is it part of a valid hypernym pattern in the full set of public comments. In this case, an approach that combines multiple technologies is required.

OntoCop is new software that works interactively with a person to organize a set of concepts into an ontology. Ontology construction consists of two subtasks: *Concept extraction* and *relation formation*. OntoCop takes a conventional approach to concept extraction, and a more novel approach to relation formation.

Concept Extraction

Concept extraction identifies the concepts mentioned in a corpus. Concepts are nouns or noun phrases. Concept extraction first extracts all possible concept candidates, such as nominal unigrams, bigrams, trigrams and named entities (NE), using Part-of-Speech (POS) tagging results. Since POS taggers make mistakes, these initial concept candidates contain false positive noun phrases. For example, "protect polar bear" is incorrectly tagged by the POS tagger as three nouns, and hence incorrectly considered a nominal trigram. A web-based frequency test filters out false positive concept candidates. Queries formed by each concept candidate are sent to the Google search engine. Among the first 10 returned snippets, if a concept candidate appears more than a threshold number of times (set to 4), it is considered a commonly-used phrase, and hence a good concept candidate; otherwise, it is discarded.

Relation Formation

Relation formation discovers relations among concepts and builds ontologies based on these relations. Given a set of concepts, OntoCop clusters the concepts and presents an initial ontology. The human then teaches OntoCop by providing manual guidance. OntoCop learns from this manual guidance and adjusts the clustering process accordingly to modify the ontology. Teaching and learning alternate until the human is satisfied.

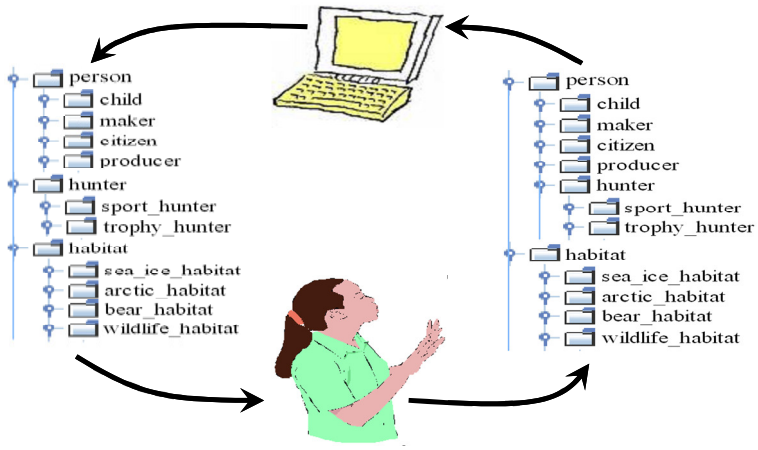

Figure 1: The human-computer interaction cycle (*Polar bear* comment set).

Figure 1 illustrates the human-computer interaction cycle for a fragment of the ontology created for the *polar bear* comment set. The cycle starts when OntoCop presents an initial ontology that consists of three concept groups: *person*, *hunter* and *habitat*. The human makes a modification by dragging and dropping the *hunter* group to be under the *person* group, which makes *hunter* a child concept of *person*. OntoCop recognizes the change, adjusts the clustering algorithm, and shows an improved ontology to the human. The human-computer interaction cycle continues until the human is satisfied with the ontology.

The initial ontology is based on *head noun matching*, *WordNet hypernyms* [4], and *lexico-syntactic patterns* [5].

If two bigrams share the same head noun, then the two bigrams are assigned to the same group; their head noun is elected as the parent of the group. For example, *pollution* becomes the parent of *water pollution* and *air pollution*. The parent concept and the child concepts form an *ontology fragment*. Head noun matching is also applied to unigrams, trigrams and named entities. The head noun matching technique effectively creates the initial ontology fragments.

WordNet hypernyms are used to identify parent/child relations among sibling concepts within an ontology fragment. Given two sibling concepts x and y, if x is y's hypernym in WordNet, x is promoted as the parent of y in their ontology fragment. Refinement by WordNet hypernyms can also be applied among ontology fragments. All roots of the ontology fragments are examined in WordNet, and are connected as one fragment if they are in the same WordNet hypernym chain.

Lexico-syntactic patterns are also used to identify parent/child relations for the roots of the ontology fragments. Each pair of roots is put into the patterns and forms a text segment. For example, “heavy metals” and “toxins” are put into the pattern “NP_B and other NP_A” to form “heavy metals and other toxins”. The text segment is searched in the corpus that supplied the concepts. If the text segment is found in the corpus, then the pair of roots has a parent/child relation and is connected.

Head noun matching, WordNet hypernyms, and lexico-syntactic patterns create an initial forest of ontology fragments, which is the initial version of the ontology. OntoCop then waits for human guidance.

During each human-computer interaction, the human modifies the ontology. The grouping of concepts before this set of modifications is represented by a *before matrix*; likewise, the new grouping of the concepts is represented by an *after matrix*. Formally, an ontology with n concepts can be represented by a nxn *ontology matrix*. The $(i, j)^{th}$ entry of an *ontology matrix* indicates whether c_i, the concept at the i^{th} row, is a sibling of c_j, the concept at the j^{th} column. The value of the $(i, j)^{th}$ entry v_{ij}=1 if i=j or c_i is a sibling of c_j; 0, otherwise. The manual guidance M is a submatrix

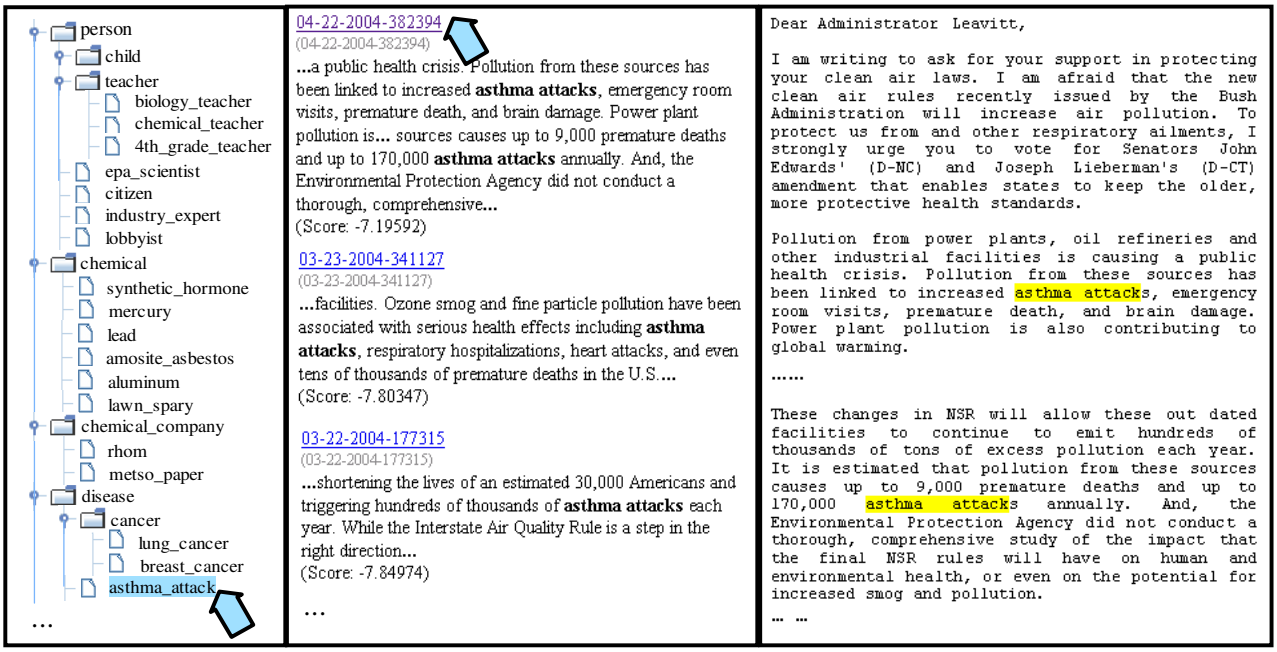

Figure 2: Using an ontology for drilldown (*Mercury* comment set).

that consists of some entries of the *after matrix*; at these entries, there exists difference between the *before matrix* and the *after matrix*. Formally, if A is the before matrix and B is the after matrix, then M = B[r; c], where $r = \{i : b_{ij} - a_{ij} \neq 0\}$, $c = \{j : b_{ij} - a_{ij} \neq 0\}$, a_{ij} is the $(i, j)^{th}$ entry in A, and b_{ij} is the $(i, j)^{th}$ entry in B. For example, the following manual guidance is obtained by dragging and dropping concept *hunter* to be under concept *person*:

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} =$$

	Person	*Hunter*	*Producer*
Person	1	0	0
Hunter	0	1	1
Producer	0	1	1

OntoCop modifies the ontology once at each human-computer interaction by using the manual guidance M as training data. A supervised distance learning algorithm learns a distance function between concepts in M. The distance function used here is a Mahalanobis distance [6]. Mahalanobis distance measures the correlation between two concepts by assigning adaptive weights to different underlying feature functions. The feature functions include *contextual features*, *co-occurrence*, *syntactic dependency*, *lexical-syntactic patterns*, *word length difference* and *definition overlap*. These heterogonous features evaluate the semantic relation between two concepts from different aspects and aim to capture a wide range of characteristics of semantic relations.

The training data consists a set of concepts $\mathbf{x}^{(i)}$, the set of concepts in $M^{(i)}$, the manual guidance obtained at the i^{th} iteration of human-computer interaction, and its corresponding distance matrix $\mathbf{y}^{(i)}$. The entry of $\mathbf{y}^{(i)}$ that corresponds to concept $x_j^{(i)}$ and $x_k^{(i)}$ is $y_{jk}^{(i)} \in \{0,1\}$, where $y_{jk}^{(i)} = 0$, if $x_j^{(i)}$ and $x_k^{(i)}$ are in the same cluster; 1, otherwise. We use Mean Squared Error as the loss function and the optimization function is defined as:

$$\min_S \sum_{j=1}^{|\mathbf{x}^{(i)}|} \sum_{k=1}^{|\mathbf{x}^{(i)}|} \left(y_{jk}^{(i)} - \sqrt{\Phi\left(x_j^{(i)}, x_k^{(i)}\right)^T S^{-1} \Phi\left(x_j^{(i)}, x_k^{(i)}\right)} \right)^2, \text{ subject to } S \succcurlyeq 0.$$

where $\Phi(x_j^{(i)}, x_k^{(i)})$ represents a set of pairwise underlying feature functions; S is the parameter matrix, which weighs the feature functions.

Given the learned parameter matrix *S*, OntoCop then applies the newly-learned distance function to obtain distance scores for the ungrouped concepts in the ontology. The *K*-medoids clustering algorithm further clusters the ungrouped concepts based on these scores and produces the next version of the ontology. The ungrouped concepts are the top level concepts in the existing ontology. OntoCop clusters concepts at one level during each iteration; the entire concept hierarchy is built in a bottom-up manner. OntoCop then presents the modified ontology to the human and waits for the next round of manual guidance.

Though *concept extraction* and *relation formation,* OntoCop builds ontologies interactively according to manual guidance. Figure 2 shows part of an ontology for the *Mercury* comments, and how to use it for navigation and drilldown. A policymaker viewing the ontology (Figure 2, left pane) quickly sees the issues raised by the public. She may be familiar with many of these issues, but "*asthma attack*" may be unexpected. A click on "*asthma attack*" leads to a list of comments that discuss this topic (Figure 2, middle pane). A click on any comment displays its text with the concept conveniently highlighted (Figure 2, right pane). This combination of navigation and drilldown helps a policymaker to quickly understand what issues the public raised, and what was said about each issue, thus improves the responsiveness of regulatory agencies during the rulemaking process.

Experiments

We collaborated with the Qualitative Data Analysis Program (QDAP) at the University of Pittsburgh's University Center for Social and Urban Research (UCSUR) to conduct a user evaluation. Twelve professional coders familiar with the problem domain participated in the experiment. They were divided into two groups: four for the manual group and eight for the interactive group. Both groups were asked to construct an ontology with the same concept candidates until they felt satisfied with their work or reached a 90-minute limit. Three public comment datasets were used in the experiments: *Wolf (RIN-1018-AU53), Polar Bear (RIN-1018-AV19)* and *Mercury (OAR-2002-0056).*[1] Table 1 gives an overview, including the total number of comments received, the number of unique comments, duration of the comment period, and the topic, of these three public comment datasets.

Table 1: Overview of Three Public Comment Datasets.

Dataset	#Comments	#Unique Comments	Duration	Topic
Wolf	282,992	59,109	02/07-08/07	Delisting the northern rocky mountain population of gray wolves from the federal list of endangered and threatened wildlife.
Polar Bear	624,947	73,989	01/07-10/07	Listing the polar bear as threatened throughout its range under the Act (72 FR 1064).
Mercury	536,975	104,146	02/04-06/04	Proposing national emission standards for hazardous air pollutants.

Quality of Interactively vs. Manually Constructed Ontologies

This experiment investigates whether OntoCop 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. Cohen's Kappa statistic is used to assess the intercoder agreement. Table 2 shows the averaged intercoder agreement for parent/child pairs in three public comment datasets. 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. The significant test results are in the range of $t < 2$ and $p > 0.01$, which shows no statistically significant differences between pairs of manually-built ontologies and interactively-built ontologies.

Table 2: Average Intercoder Agreement on Parent/Child Pairs.

Dataset	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

[1] http://erulemaking.cs.cmu.edu/data.php.

Table 3: Average Manual Editing Costs.

	Add	Delete	Move	Rename	Undo	Total
Manual	57	200	2807	71	19	3153
Interactive	21	129	1694	40	8	1890

Table 4: Ontology Construction Duration.

	Wolf	Polar Bear	Mercury	Average
Manual	1:24	1:22	1:33	1:27
Interactive: human	0:33	0:29	0:30	0.31
computer	0:33	0:05	0:35	0:24

Costs of Interactively vs. Manually Constructed Ontologies

We compare the construction logs for users from both manual and interactive groups. Table 3 shows the number of manual edits of building ontologies for the three datasets. The edits include adding a concept, moving a concept by drag & drop, deleting a concept, changing name for a concept and undoing previous actions. In total, the interactive users use 40% less editing actions to produce the same quality ontologies. A one-tailed t-test shows a significant reduction, $t=10$ and $p < 0.001$, in the interactive runs in terms of editing costs as compared to the manual runs. It demonstrates that OntoCop is significantly more cost effective than the manual work. Table 4 shows the actual time needed to construct an ontology for both manual and interactive runs. It also shows the amount of time spent by the human and the amount of time spent by the machine in the interactive runs. In general, the 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 OntoCop greatly saves a human user's time to construct ontologies.

Conclusion

OntoCop helps a person to construct an ontology that provides a way of understanding the "lay of the land" in a corpus, and then "drilling down" into specific topic areas. Unlike most prior research on ontology construction, OntoCop is designed for corpora that have a very task-specific focus, such as public comments submitted to regulatory agencies during Notice and Comment Rulemaking. Such domains contain many similar concepts, rare concepts, and rare relationships. OntoCop relies on human guidance, and a set of feature functions that give it a stronger ability to disambiguate the subtle differences among similar concepts. Experimental results on three corpora of public comments show that interactive learning not only produces useful, domain-specific ontologies, but also saves time and human effort, thus accelerating the process of developing task-specific ontologies.

Acknowledgements

This research was supported by NSF grant IIS-0704210. Any opinions, findings, conclusions, or recommendations expressed in this paper are of the authors, and do not necessarily reflect those of the sponsor.

References

1. H. Yang and J. Callan. Near-duplicate Detection for eRulemaking. In Proceedings of the 15th National Conference on Digital Government Research (Dg.o), 2005.
2. D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In Proceedings of the 40th Annual Meeting for the Association for Computational Linguistics (ACL), 2002.
3. P. Pantel and D. Ravichandran. Automatically labeling semantic classes. In Proceedings of Human Language Technology conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL), 2004.
4. C. Fellbaum. WordNet:An Electronic Lexical Database. MIT Press, 1998.
5. M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics (COLING), 1992.
6. P. C. Mahalanobis. On the Generalised Distance in Statistics. In Proceedings of the National Institute of Sciences of India 2 (1): 49–55, 1936.

Hui Yang is a PhD student at Carnegie Mellon University's Language Technologies Institute. Contact her at huiyang@cs.cmu.edu.
Jamie Callan is a professor at Carnegie Mellon University's Language Technologies Institute and Heinz School of Public Policy and Management. Contact him at callan@cs.cmu.edu.