Outclassing Wikipedia in Open-Domain Information Extraction: Weakly-Supervised Acquisition of Attributes over Conceptual Hierarchies

Marius Paşca
Google Inc.
Mountain View, California 94043
mars@google.com

Abstract

A set of labeled classes of instances is extracted from text and linked into an existing conceptual hierarchy. Besides a significant increase in the coverage of the class labels assigned to individual instances, the resulting resource of labeled classes is more effective than similar data derived from the manually-created Wikipedia, in the task of attribute extraction over conceptual hierarchies.

1 Introduction

Motivation: Sharing basic intuitions and long-term goals with other tasks within the area of Web-based information extraction (Banko and Etzioni, 2008; Davidov and Rappoport, 2008), the task of acquiring class attributes relies on unstructured text available on the Web, as a data source for extracting generally-useful knowledge. In the case of attribute extraction, the knowledge to be extracted consists in quantifiable properties of various classes (e.g., *top speed*, *body style* and *gas mileage* for the class of *sports cars*).

Existing work on large-scale attribute extraction focuses on producing ranked lists of attributes, for target classes of instances available in the form of flat sets of instances (e.g., *ferrari modena*, *porsche carrera gt*) sharing the same class label (e.g., *sports cars*). Independently of how the input target classes are populated with instances (manually (Paşca, 2007) or automatically (Paşca and Van Durme, 2008)), and what type of textual data source is used for extracting attributes (Web documents or query logs), the extraction of attributes operates at a lexical rather than semantic level. Indeed, the class labels of the target classes may be not more than text surface strings (e.g., *sports cars*) or even artificially-created labels (e.g., *CartoonChar* in lieu of *cartoon characters*). Moreover, although it is commonly accepted that *sports cars* are also *cars*, which in turn are also *motor vehicles*, the presence of *sports cars* among the input target classes does not lead to any attributes being extracted for *cars* and *motor vehicles*, unless the latter two class labels are also present explicitly among the input target classes.

Contributions: The contributions of this paper are threefold. First, we investigate the role of classes of instances acquired automatically from unstructured text, in the task of attribute extraction over concepts from existing conceptual hierarchies. For this purpose, ranked lists of attributes are acquired from query logs for various concepts, after linking a set of more than 4,500 open-domain, automatically-acquired classes containing a total of around 250,000 instances into conceptual hierarchies available in WordNet (Fellbaum, 1998). In comparison, previous work extracts attributes for either manually-specified classes of instances (Paşca, 2007), or for classes of instances derived automatically but considered as flat rather than hierarchical classes, and manually associated to existing semantic concepts (Paşca and Van Durme, 2008). Second, we expand the set of classes of instances acquired from text, thus increasing their usefulness in attribute extraction in particular and information extraction in general. To this effect, additional class labels (e.g., *motor vehicles*) are identified for existing instances (e.g., *ferrari modena*) of existing class labels (e.g., *sports cars*), by exploiting IsA relations available within the conceptual hierarchy (e.g., *sports cars* are also *motor vehicles*). Third, we show that large-scale, automatically-derived classes of in-

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 639–647,
Athens, Greece, 30 March – 3 April 2009.

stances can have as much as, or even bigger, practical impact in open-domain information extraction tasks than similar data from large-scale, high-coverage, manually-compiled resources. Specifically, evaluation results indicate that the accuracy of the extracted lists of attributes is higher by 8% at rank 10, 13% at rank 30 and 18% at rank 50, when using the automatically-extracted classes of instances rather than the comparatively more numerous and a-priori more reliable, human-generated, collaboratively-vetted classes of instances available within Wikipedia (Remy, 2002).

2 Attribute Extraction over Hierarchies

Extraction of Flat Labeled Classes: Unstructured text from a combination of Web documents and query logs represents the source for deriving a flat set of labeled classes of instances, which are necessary as input for attribute extraction experiments. The labeled classes are acquired in three stages:

1) extraction of a noisy pool of pairs of a class label and a potential class instance, by applying a few Is-A extraction patterns, selected from (Hearst, 1992), to Web documents:

(*fruits, apple*), (*fruits, corn*), (*fruits, mango*), (*fruits, orange*), (*foods, broccoli*), (*crops, lettuce*), (*flowers, rose*);

2) extraction of unlabeled clusters of distributionally similar phrases, by clustering vectors of contextual features collected around the occurrences of the phrases within Web documents (Lin and Pantel, 2002):

{*lettuce, broccoli, corn, ..*},

{*carrot, mango, apple, orange, rose, ..*};

3) merging and filtering of the raw pairs and unlabeled clusters into smaller, more accurate sets of class instances associated with class labels, in an attempt to use unlabeled clusters to filter noisy raw pairs instead of merely using clusters to generalize class labels across raw pairs (Paşca and Van Durme, 2008):

fruits={*apple, mango, orange, ..*}.

To increase precision, the vocabulary of class instances is confined to the set of queries that are most frequently submitted to a general-purpose Web search engine. After merging, the resulting pairs of an instance and a class label are arranged into instance sets (e.g., {*ferrari modena, porsche carrera gt*}), each associated with a class label (e.g., *sports cars*).

Linking Labeled Classes into Hierarchies: Manually-constructed language resources such as WordNet provide reliable, wide-coverage upper-level conceptual hierarchies, by grouping together phrases with the same meaning (e.g., {*analgesic, painkiller, pain pill*}) into sets of synonyms (synsets), and organizing the synsets into conceptual hierarchies (e.g., *painkillers* are a subconcept, or a hyponym, of *drugs*) (Fellbaum, 1998). To determine the points of insertion of automatically-extracted labeled classes into hand-built WordNet hierarchies, the class labels are looked up in WordNet using built-in morphological normalization routines. When a class label (e.g., *age-related diseases*) is not found in WordNet, it is looked up again after iteratively removing its leading words (e.g., *related diseases*, and *diseases*) until a potential point of insertion is found where one or more senses exist in WordNet for the class label.

An efficient heuristic for sense selection is to uniformly choose the first (that is, most frequent) sense of the class label in WordNet, as point of insertion. Due to its simplicity, the heuristic is bound to make errors whenever the correct sense is not the first one, thus incorrectly linking *academic journals* under the sense of *journals* as personal diaries rather than periodicals, and *active volcanoes* under the sense of *volcanoes* as fissures in the earth, rather than mountains formed by volcanic material. Nevertheless, choosing the first sense is attractive for three reasons. First, WordNet senses are often too fine-grained, making the task of choosing the correct sense difficult even for humans (Palmer et al., 2007). Second, choosing the first sense from WordNet is sometimes better than more intelligent disambiguation techniques (Pradhan et al., 2007). Third, previous experimental results on linking Wikipedia classes to WordNet concepts confirm that first-sense selection is more effective in practice than other techniques (Suchanek et al., 2007). Thus, a class label and its associated instances are inserted under the first WordNet sense available for the class label. For example, *silicon valley companies* and its associated instances (*apple, hewlett packard* etc.) are inserted under the first of the 9 senses of *companies* in WordNet, which corresponds to companies as institutions created to conduct business.

In order to trade off coverage for higher precision, the heuristic can be restricted to link a class label under the first WordNet sense available, as

before, but only when no other senses are available at the point of insertion beyond the first sense. With the modified heuristic, the class label *internet search engines* is linked under the first and only sense of *search engines* in WordNet, but *silicon valley companies* is no longer linked under the first of the 9 senses of *companies*.

Extraction of Attributes for Hierarchy Concepts: The labeled classes of instances linked to conceptual hierarchies constitute the input to the acquisition of attributes of hierarchy concepts, by mining a collection of Web search queries. The attributes capture properties that are relevant to the concept. The extraction of attributes exploits the sets of class instances rather than the associated class labels. More precisely, for each hierarchy concept for which attributes must be extracted, the instances associated to all class labels linked under the subhierarchy rooted at the concept are collected as a union set of instances, thus exploiting the transitivity of IsA relations. This step is equivalent to propagating the instances upwards, from their class labels to higher-level WordNet concepts under which the class labels are linked, up to the root of the hierarchy. The resulting sets of instances constitute the input to the acquisition of attributes, which consists of four stages:

1) identification of a noisy pool of candidate attributes, as remainders of queries that also contain one of the class instances. In the case of the concept *movies*, whose instances include *jay and silent bob strike back* and *kill bill*, the query *"cast jay and silent bob strike back"* produces the candidate attribute *cast*;

2) construction of internal vector representations for each candidate attribute, based on queries (e.g., *"cast selection for kill bill"*) that contain a candidate attribute (*cast*) and a class instance (*kill bill*). These vectors consist of counts tied to the frequency with which an attribute occurs with a given "templatized" query. The latter replaces specific attributes and instances from the query with common placeholders, e.g., *"X for Y"*;

3) construction of a reference internal vector representation for a small set of seed attributes provided as input. A reference vector is the normalized sum of the individual vectors corresponding to the seed attributes;

4) ranking of candidate attributes with respect to each concept, by computing the similarity between their individual vector representations and the reference vector of the seed attributes.

The result of the four stages, which are described in more detail in (Paşca, 2007), is a ranked list of attributes (e.g., [*opening song*, *cast*, *characters*,...]) for each concept (e.g., *movies*).

3 Experimental Setting

Textual Data Sources: The acquisition of open-domain knowledge relies on unstructured text available within a combination of Web documents maintained by, and search queries submitted to the Google search engine. The textual data source for extracting labeled classes of instances consists of around 100 million documents in English, as available in a Web repository snapshot from 2006. Conversely, the acquisition of open-domain attributes relies on a random sample of fully-anonymized queries in English submitted by Web users in 2006. The sample contains about 50 million unique queries. Each query is accompanied by its frequency of occurrence in the logs. Other sources of similar data are available publicly for research purposes (Gao et al., 2007).

Parameters for Extracting Labeled Classes: When applied to the available document collection, the method for extracting open-domain classes of instances from unstructured text introduced in (Paşca and Van Durme, 2008) produces 4,583 class labels associated to 258,699 unique instances, for a total of 869,118 pairs of a class instance and an associated class label. All collected instances occur among to the top five million queries with the highest frequency within the input query logs. The data is further filtered by discarding labeled classes with fewer than 25 instances. The classes, examples of which are shown in Table 1, are linked under conceptual hierarchies available within WordNet 3.0, which contains a total of 117,798 English noun phrases grouped in 82,115 concepts (or synsets).

Parameters for Extracting Attributes: For each target concept from the hierarchy, given the union of all instances associated to class labels linked to the target concept or one of its subconcepts, and given a set of five seed attributes (e.g., {*quality*, *speed*, *number of users*, *market share*, *reliability*} for *search engines*), the method described in (Paşca, 2007) extracts ranked lists of attributes from the input query logs. Internally, the ranking of attributes uses Jensen-Shannon (Lee, 1999) to compute similarity scores between internal rep-

Class Label	Class Size	Class Instances
accounting systems	40	flexcube, myob, oracle financials, peachtree accounting, sybiz
antimicrobials	97	azithromycin, chloramphenicol, fusidic acid, quinolones, sulfa drugs
civilizations	197	ancient greece, chaldeans, etruscans, inca, indians, roman republic
elementary particles	33	axions, electrons, gravitons, leptons, muons, neutrons, positrons
farm animals	61	angora goats, burros, cattle, cows, donkeys, draft horses, mule, oxen
forages	27	alsike clover, rye grass, tall fescue, sericea lespedeza, birdsfoot trefoil
ideologies	179	egalitarianism, laissez-faire capitalism, participatory democracy
social events	436	academic conferences, afternoon teas, block parties, masquerade balls

Table 1: Examples of instances within labeled classes extracted from unstructured text, used as input for attribute extraction experiments

resentations of seed attributes, on one hand, and each of the newly acquired attributes, on the other hand. Depending on the experiments, the amount of supervision is thus limited to either 5 seed attributes for each target concept, or to 5 seed attributes (*population*, *area*, *president*, *flag* and *climate*) provided for only one of the extracted labeled classes, namely *european countries*.

Experimental Runs: The experiments consist of four different runs, which correspond to different choices for the source of conceptual hierarchies and class instances linked to those hierarchies, as illustrated in Table 2. In the first run, denoted N, the class instances are those available within the latest version of WordNet (3.0) itself via HasInstance relations. The second run, Y, corresponds to an extension of WordNet based on the manually-compiled classes of instances from categories in Wikipedia, as available in the 2007-w50-5 version of Yago (Suchanek et al., 2007). Therefore, run Y has the advantage of the fact that Wikipedia categories are a rich source of useful and accurate knowledge (Nastase and Strube, 2008), which explains their previous use as a source for evaluation gold standards (Blohm et al., 2007). The last two runs from Table 2, E_s and E_a, correspond to the set of open-domain labeled classes acquired from unstructured text. In both E_s and E_a, class labels are linked to the first sense available at the point of insertion in WordNet. In E_s, the class labels are linked only if no other senses are available at the point of insertion beyond the first sense, thus promoting higher linkage precision at the expense of fewer links. For example, since the phrases *impressionists*, *sports cars* and *painters* have 1, 1 and 4 senses available in WordNet respectively, the class labels *french impressionists* and *sports cars* are linked to the respective WordNet concepts, whereas the class label *painters* is not. Comparatively, in E_a, the class labels are uniformly linked to the first sense available in WordNet, regardless of whether other senses may or may not be available. Thus, E_a trades off potentially lower precision for the benefit of higher linkage recall, and results in more of the class labels and their associated instances extracted from text to be linked to WordNet than in the case of run E_s.

Description	Source of Hierarchy and Instances			
	N	Y	E_s	E_a
Include instances from WordNet?	√	√	-	-
Include instances from elsewhere?	-	√	√	√
#Instances ($\times 10^3$)	14.3	1,296.5	108.0	257.0
#Class labels	945	30,338	1,315	4,517
#Pairs of a class label and instance ($\times 10^3$)	17.4	2,839.8	191.0	859.0

Table 2: Source of class instances for various experimental runs

4 Evaluation

4.1 Evaluation of Labeled Classes

Coverage of Class Instances: In run N, the input class instances are the component phrases of synsets encoded via HasInstance relations under other synsets in WordNet. For example, the synset corresponding to {*search engine*}, defined as *"a computer program that retrieves documents or files or data from a database or from a computer network"*, has 3 HasInstance instances in WordNet, namely *Ask Jeeves*, *Google* and *Yahoo*. Table 3 illustrates the coverage of the class instances extracted from unstructured text and linked to WordNet in runs E_s and E_a respectively, relative to all 945 WordNet synsets that contain HasInstance instances. Note that the coverage scores are conservative assessments of actual coverage, since a run (i.e., E_s or E_a) receives credit for a WordNet instance only if the run contains an instance that is a full-length, case-insensitive match (e.g., *ask*

Concept		HasInstance Instances within WordNet		Cvg	
Synset	Offset	Examples	Count	E_s	E_a
{existentialist, existentialist, philosopher, existential philosopher}	10071557	Albert Camus, Beauvoir, Camus, Heidegger, Jean-Paul Sartre	8	1.00	1.00
{search engine}	06578654	Ask Jeeves, Google, Yahoo	3	1.00	1.00
{university}	04511002	Brown, Brown University, Carnegie Mellon University	44	0.61	0.77
{continent}	09254614	Africa, Antarctic continent, Europe, Eurasia, Gondwanaland, Laurasia	13	0.54	0.54
{microscopist}	10313872	Anton van Leeuwenhoek, Anton van Leuwenhoek, Swammerdam	6	0.00	0.00
Average over all 945 WordNet concepts that have HasInstance instance(s)			18.71	0.21	0.40

Table 3: Coverage of class instances extracted from text and linked to WordNet (used as input in runs E_s and E_a respectively), measured as the fraction of WordNet HasInstance instances (used as input in run N) that occur among the class instances (Cvg=coverage)

jeeves) of the WordNet instance. On average, the coverage scores for class instances of runs E_s and E_a relative to run N are 0.21 and 0.40 respectively, as shown in the last row in Table 3. Comparatively, the equivalent instance coverage for run Y, which already includes most of the WordNet instances by design (cf. (Suchanek et al., 2007)), is 0.59.

Relative Coverage of Class Labels: The linking of class labels to WordNet concepts allows for the expansion of the set of classes of instances acquired from text, thus increasing its usefulness in attribute extraction in particular and information extraction in general. To this effect, additional class labels are identified for existing instances, in the form of component phrases of the synsets that are superconcepts (or hypernyms, in WordNet terminology) of the synset under which the class label of the instance is linked in WordNet. For example, since the class label *sports cars* is linked under the WordNet synset {*sports car*, *sport car*}, and the latter has the synset {*motor vehicle, automotive vehicle*} among its hypernyms, the phrases *motor vehicles* and *automotive vehicles* are collected as new class labels [1] and associated to existing instances of *sports cars* from the original set, such as *ferrari modena*. No phrases are collected from a selected set of 10 top-level WordNet synsets, including {*entity*} and {*object, physical object*}, which are deemed too general to be useful as class labels. As illustrated in Table 4, a collected pair of a new class label and an existing instance either does not have any impact, if the pair already occurs in the original set of labeled classes; or expands existing classes, if the class label already occurs in the original set of labeled classes but not in association to the instance; or creates new classes of instances, if the class label is not part of the original set. The latter two cases aggregate to increases in coverage, relative to the pairs from the original sets of labeled classes, of 53% for E_s and 304% for E_a.

Already in original labeled classes:	
painters	alfred sisley
european countries	austria
Expansion of existing labeled classes:	
animals	avocet
animals	northern oriole
scientists	howard gardner
scientists	phil zimbardo
Creation of new labeled classes:	
automotive vehicles	acura nsx
automotive vehicles	detomaso pantera
creative persons	aaron copland
creative persons	yoshitomo nara

Table 4: Examples of additional class labels collected from WordNet, for existing instances of the original labeled classes extracted from text

4.2 Evaluation of Attributes

Target Hierarchy Concepts: The performance of attribute extraction is assessed over a set of 25 target concepts also used for evaluation in (Paşca, 2008). The set of 25 target concepts includes: *Actor*, *Award*, *Battle*, *CelestialBody*, *ChemicalElement*, *City*, *Company*, *Country*, *Currency*, *DigitalCamera*, *Disease*, *Drug*, *FictionalCharacter*, *Flower*, *Food*, *Holiday*, *Mountain*, *Movie*, *NationalPark*, *Painter*, *Religion*, *River*, *SearchEngine*, *Treaty*, *Wine*. Each target concept represents exactly one WordNet concept (synset). For instance,

[1] For consistency with the original labeled classes, new class labels collected from WordNet are converted from singular (e.g., *motor vehicle*) to plural (e.g., *motor vehicles*).

one of the target concepts, denoted *Country*, corresponds to a synset situated at the internal offset 08544813 in WordNet 3.0, which groups together the synonymous phrases *country*, *state* and *land* and associates them with the definition *"the territory occupied by a nation"*. The target concepts exhibit variation with respect to their depths within WordNet conceptual hierarchies, ranging from a minimum of 5 (e.g., for *Food*) to a maximum of 11 (for *Flower*), with a mean depth of 8 over the 25 concepts.

Evaluation Procedure: The measurement of recall requires knowledge of the complete set of items (in our case, attributes) to be extracted. Unfortunately, this number is often unavailable in information extraction tasks in general (Hasegawa et al., 2004), and attribute extraction in particular. Indeed, the manual enumeration of all attributes of each target concept, to measure recall, is unfeasible. Therefore, the evaluation focuses on the assessment of attribute accuracy.

To remove any bias towards higher-ranked attributes during the assessment of class attributes, the ranked lists of attributes produced by each run to be evaluated are sorted alphabetically into a merged list. Each attribute of the merged list is manually assigned a correctness label within its respective class. In accordance with previously introduced methodology, an attribute is *vital* if it must be present in an ideal list of attributes of the class (e.g., *side effects* for *Drug*); *okay* if it provides useful but non-essential information; and *wrong* if it is incorrect (Paşca, 2007).

To compute the precision score over a ranked list of attributes, the correctness labels are converted to numeric values (*vital* to 1, *okay* to 0.5 and *wrong* to 0). Precision at some rank N in the list is thus measured as the sum of the assigned values of the first N attributes, divided by N.

Attribute Accuracy: Figure 1 plots the precision at ranks 1 through 50 for the ranked lists of attributes extracted by various runs as an average over the 25 target concepts, along two dimensions. In the leftmost graphs, each of the 25 target concepts counts towards the computation of precision scores of a given run, regardless of whether any attributes were extracted or not for the target concept. In the rightmost graphs, only target concepts for which some attributes were extracted are included in the precision scores of a given run. Thus, the leftmost graphs properly penalize a run

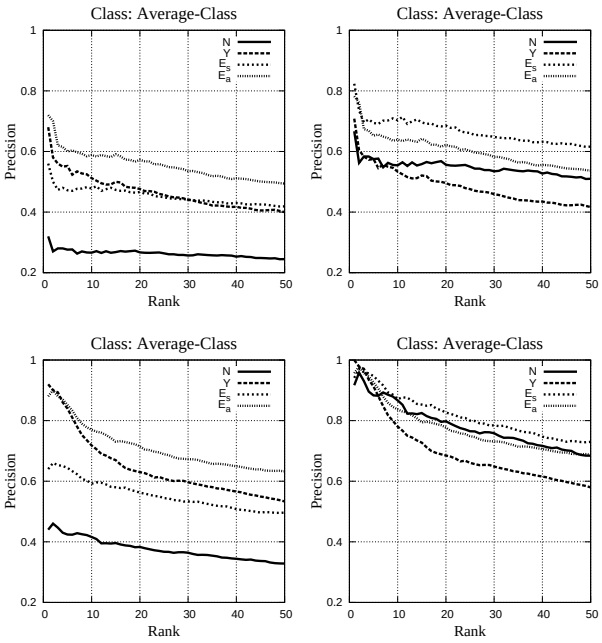

Figure 1: Accuracy of the attributes extracted for various runs, as an average over the entire set of 25 target concepts (left graphs) and as an average over (variable) subsets of the 25 target concepts for which some attributes were extracted in each run (right graphs). Seed attributes are provided as input for only one target concept (top graphs), or for each target concept (bottom graphs)

for failing to extract any attributes for some target concepts, whereas the rightmost graphs do not include any such penalties. On the other dimension, in the graphs at the top of Figure 1, seed attributes are provided only for one class (namely, *european countries*), for a total of 5 attributes over all classes. In the graphs at the bottom of the figure, there are 5 seed attributes for each of the 25 target concepts in the graphs at the bottom of Figure 1, for a total of 5×25=125 attributes.

Several conclusions can be drawn after inspecting the results. First, providing more supervision, in the form of seed attributes for all concepts rather than for only one concept, translates into higher attribute accuracy for all runs, as shown by the graphs at the top vs. graphs at the bottom of Figure 1. Second, in the leftmost graphs, run N has the lowest precision scores, which is in line with the relatively small number of instances available in the original WordNet, as confirmed by the counts from Table 2. Third, in the leftmost graphs, the more restrictive run E_s has lower precision scores across all ranks than its less restrictive counterpart E_a. In other words, adding more

Class	Precision											
	@10				@30				@50			
	N	Y	E_s	E_a	N	Y	E_s	E_a	N	Y	E_s	E_a
Actor	1.00	1.00	1.00	1.00	0.78	0.85	0.98	0.95	0.62	0.84	0.95	0.96
Award	0.00	0.50	0.95	0.85	0.00	0.35	0.80	0.73	0.00	0.29	0.70	0.69
Battle	0.80	0.90	0.00	0.90	0.76	0.80	0.00	0.80	0.74	0.72	0.00	0.73
CelestialBody	1.00	1.00	1.00	0.40	1.00	1.00	0.93	0.16	0.98	0.89	0.91	0.12
ChemicalElement	0.00	0.65	0.80	0.80	0.00	0.45	0.83	0.63	0.00	0.48	0.84	0.51
City	1.00	1.00	0.00	1.00	0.86	0.80	0.00	0.83	0.78	0.70	0.00	0.76
Company	0.00	1.00	0.90	1.00	0.00	0.90	0.93	0.88	0.00	0.77	0.82	0.80
Country	1.00	0.90	1.00	1.00	0.98	0.81	0.96	0.96	0.97	0.76	0.98	0.97
Currency	0.00	0.90	0.00	0.90	0.00	0.53	0.00	0.83	0.00	0.36	0.00	0.87
DigitalCamera	0.00	0.20	0.85	0.85	0.00	0.10	0.85	0.85	0.00	0.10	0.82	0.82
Disease	0.00	0.60	0.75	0.75	0.00	0.76	0.83	0.83	0.00	0.63	0.87	0.86
Drug	0.00	1.00	1.00	1.00	0.00	0.91	1.00	1.00	0.00	0.88	0.96	0.96
FictionalCharacter	0.80	0.70	0.00	0.55	0.65	0.48	0.00	0.38	0.42	0.41	0.00	0.34
Flower	0.00	0.65	0.00	0.70	0.00	0.26	0.00	0.55	0.00	0.16	0.00	0.53
Food	0.00	0.80	0.90	1.00	0.00	0.65	0.71	0.96	0.00	0.53	0.59	0.96
Holiday	0.00	0.60	0.80	0.80	0.00	0.50	0.48	0.48	0.00	0.37	0.41	0.41
Mountain	1.00	0.75	0.00	0.90	0.96	0.61	0.00	0.86	0.77	0.58	0.00	0.74
Movie	0.00	1.00	1.00	1.00	0.00	0.90	0.80	0.78	0.00	0.85	0.75	0.74
NationalPark	0.90	0.80	0.00	0.00	0.85	0.76	0.00	0.00	0.82	0.75	0.00	0.00
Painter	1.00	1.00	1.00	1.00	0.96	0.93	0.88	0.96	0.92	0.89	0.76	0.93
Religion	0.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.00	0.92	0.97
River	1.00	0.80	0.00	0.00	0.70	0.60	0.00	0.00	0.61	0.58	0.00	0.00
SearchEngine	0.40	0.00	0.25	0.25	0.23	0.00	0.35	0.35	0.32	0.00	0.43	0.43
Treaty	0.50	0.90	0.80	0.80	0.33	0.65	0.53	0.53	0.26	0.59	0.42	0.42
Wine	0.00	0.30	0.80	0.80	0.00	0.26	0.43	0.45	0.00	0.20	0.28	0.29
Average (over 25)	0.41	0.71	0.59	**0.77**	0.36	0.59	0.53	**0.67**	0.32	0.53	0.49	**0.63**
Average (over non-empty)	0.86	0.78	**0.87**	0.83	0.75	0.64	**0.78**	0.73	0.68	0.57	**0.73**	0.68

Table 5: Comparative accuracy of the attributes extracted by various runs, for individual concepts, as an average over the entire set of 25 target concepts, and as an average over (variable) subsets of the 25 target concepts for which some attributes were extracted in each run. Seed attributes are provided as input for each target concept

restrictions may improve precision but hurts recall of class instances, which results in lower average precision scores for the attributes. Fourth, in the leftmost graphs, the runs using the automatically-extracted labeled classes (E_s and E_a) not only outperform N, but one of them (E_a) also outperforms Y. This is the most important result. It shows that large-scale, automatically-derived classes of instances can have as much as, or even bigger, practical impact in attribute extraction than similar data from larger (cf. Table 2), manually-compiled, collaboratively created and maintained resources such as Wikipedia. Concretely, in the graph on the bottom left of Figure 1, the precision scores at ranks 10, 30 and 50 are 0.71, 0.59 and 0.53 for run Y, but 0.77, 0.67 and 0.63 for run E_a. The scores correspond to attribute accuracy improvements of 8% at rank 10, 13% at rank 30, and 18% at rank 50 for run E_a over run Y. In fact, in the rightmost graphs, that is, without taking into account target concepts without any extracted attributes, the precision scores of both E_s and E_a are higher than for run Y across most, if not all, ranks from 1 through 50. In this case, it is E_1 that produces the most accurate attributes, in a task-based demonstration that the more cautious linking of class labels to WordNet concepts in E_s vs. E_a leads to less coverage but higher precision of the linked labeled classes, which translates into extracted attributes of higher accuracy but for fewer target concepts.

Analysis: The curves plotted in the two graphs at the bottom of Figure 1 are computed as averages over precision scores for individual target concepts, which are shown in detail in Table 5. Precision scores of 0.00 correspond to runs for which no attributes are acquired from query logs, because no instances are available in the subhierarchy rooted at the respective concepts. For example, precision scores for run N are 0.00 for *Award* and *DigitalCamera*, among others concepts in Table 5, due to the lack of any HasInstance instances in WordNet for the respective concepts. The number of target concepts for which some attributes are extracted is 12 for run N, 23 for Y, 17 for E_s

and 23 for E_a. Thus, both run N and run E_s exhibit rather binary behavior across individual classes, in that they tend to either not retrieve any attributes or retrieve attributes of relatively higher quality than the other runs, causing E_s and N to have the worst precision scores in the last but one row of Table 5, but the best precision scores in the last row of Table 5.

The individual scores shown for E_s and E_a in Table 5 concur with the conclusion drawn earlier from the graphs in Figure 1, that Run E_s has lower precision than E_a as an average over all target concepts. Notable exceptions are the scores obtained for the concepts *CelestialBody* and *ChemicalElement*, where E_s significantly outperforms E_a in Table 5. This is due to confusing instances (e.g., *kobe bryant*) being associated with class labels (e.g., *nba stars*) that are incorrectly linked under the target concepts (e.g., *Star*, which is a subconcept of *CelestialBody* in WordNet) in E_a, but not linked at all and thus not causing confusion in E_s.

Run Y performs better than E_a for 5 of the 25 individual concepts, including *NationalPark*, for which no instances of *national parks* or related class labels are available in run E_a; and *River*, for which relevant instances in the labeled classes in E_a, but they are associated to the class label *river systems*, which is incorrectly linked to the WordNet concept *systems* rather than to *rivers*. However, run E_a outperforms Y on 12 individual concepts (e.g., *Award*, *DigitalCamera* and *Disease*), and also as an average over all classes (last two rows in Table 5).

5 Related Work

Previous work on the automatic acquisition of attributes for open-domain classes from text requires the manual enumeration of sets of instances and seed attributes, for each class for which attributes are to be extracted. In contrast, the current method operates on automatically-extracted classes. The experiments reported in (Paşca and Van Durme, 2008) also exploit automatically-extracted classes for the purpose of attribute extraction. However, they operate on flat classes, as opposed to concepts organized hierarchically. Furthermore, they require manual mappings from extracted class labels into a selected set of evaluation classes (e.g., by mapping *river systems* to *River*, *football clubs* to *SoccerClub*, and *parks* to *NationalPark*), whereas the current method maps class labels to concepts automatically, by linking class labels and their associated instances to concepts. Manually-encoded attributes available within Wikipedia articles are used in (Wu and Weld, 2008) in order to derive other attributes from unstructured text within Web documents. Comparatively, the current method extracts attributes from query logs rather than Web documents, using labeled classes extracted automatically rather than available in manually-created resources, and requiring minimal supervision in the form of only 5 seed attributes provided for only one concept, rather than thousands of attributes available in millions of manually-created Wikipedia articles. To our knowledge, there is only one previous study (Paşca, 2008) that directly addresses the problem of extracting attributes over conceptual hierarchies. However, that study uses labeled classes extracted from text with a different method; extracts attributes for labeled classes and propagates them upwards in the hierarchy, in order to compute attributes of hierarchy concepts from attributes of their subconcepts; and does not consider resources similar to Wikipedia, as sources of input labeled classes for attribute extraction.

6 Conclusion

This paper introduces an extraction framework for exploiting labeled classes of instances to acquire open-domain attributes from unstructured text available within search query logs. The linking of the labeled classes into existing conceptual hierarchies allows for the extraction of attributes over hierarchy concepts, without a-priori restrictions to specific domains of interest and with little supervision. Experimental results show that the extracted attributes are more accurate when using automatically-derived labeled classes, rather than classes of instances derived from manually-created resources such as Wikipedia. Current work investigates the impact of the semantic distribution of the classes of instances on the overall accuracy of attributes; the potential benefits of using more compact conceptual hierarchies (Snow et al., 2007) on attribute accuracy; and the organization of labeled classes of instances into conceptual hierarchies, as an alternative to inserting them into existing conceptual hierarchies created manually from scratch or automatically by filtering manually-generated relations among classes from Wikipedia (Ponzetto and Strube, 2007).

References

M. Banko and O. Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 28–36, Columbus, Ohio.

S. Blohm, P. Cimiano, and E. Stemle. 2007. Harvesting relations from the web - quantifiying the impact of filtering functions. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1316–1321, Vancouver, British Columbia.

D. Davidov and A. Rappoport. 2008. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 227–235, Columbus, Ohio.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

W. Gao, C. Niu, J. Nie, M. Zhou, J. Hu, K. Wong, and H. Hon. 2007. Cross-lingual query suggestion using query logs of different languages. In *Proceedings of the 30th ACM Conference on Research and Development in Information Retrieval (SIGIR-07)*, pages 463–470, Amsterdam, The Netherlands.

T. Hasegawa, S. Sekine, and R. Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain.

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 539–545, Nantes, France.

L. Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL-99)*, pages 25–32, College Park, Maryland.

D. Lin and P. Pantel. 2002. Concept discovery from text. In *Proceedings of the 19th International Conference on Computational linguistics (COLING-02)*, pages 1–7.

V. Nastase and M. Strube. 2008. Decoding Wikipedia categories for knowledge acquisition. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1219–1224, Chicago, Illinois.

M. Paşca and B. Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 19–27, Columbus, Ohio.

M. Paşca. 2007. Organizing and searching the World Wide Web of facts - step two: Harnessing the wisdom of the crowds. In *Proceedings of the 16th World Wide Web Conference (WWW-07)*, pages 101–110, Banff, Canada.

M. Paşca. 2008. Turning Web text and search queries into factual knowledge: Hierarchical class attribute extraction. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1225–1230, Chicago, Illinois.

M. Palmer, H. Dang, and C. Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163.

S. Ponzetto and M. Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1440–1447, Vancouver, British Columbia.

S. Pradhan, E. Loper, D. Dligach, and M. Palmer. 2007. SemEval-2007 Task-17: English lexical sample, SRL and all words. In *Proceedings of the 4th Workshop on Semantic Evaluations (SemEval-07)*, pages 87–92, Prague, Czech Republic.

M. Remy. 2002. Wikipedia: The free encyclopedia. *Online Information Review*, 26(6):434.

R. Snow, S. Prakash, D. Jurafsky, and A. Ng. 2007. Learning to merge word senses. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, pages 1005–1014, Prague, Czech Republic.

F. Suchanek, G. Kasneci, and G. Weikum. 2007. Yago: a core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of the 16th World Wide Web Conference (WWW-07)*, pages 697–706, Banff, Canada.

F. Wu and D. Weld. 2008. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the 17th World Wide Web Conference (WWW-08)*, pages 635–644, Beijing, China.