Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

Turning Web Text and Search Queries into Factual Knowledge: Hierarchical Class Attribute Extraction

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

Abstract

A seed-based framework for textual information extraction allows for weakly supervised acquisition of open-domain class attributes over conceptual hierarchies, from a combination of Web documents and query logs. Automatically-extracted labeled classes, consisting of a label (e.g., *painkillers*) and an associated set of instances (e.g., *vicodin*, *oxycontin*), are linked under existing conceptual hierarchies (e.g., *brain disorders* and *skin diseases* are linked under the concepts *BrainDisorder* and *SkinDisease* respectively). Attributes extracted for the labeled classes are propagated upwards in the hierarchy, to determine the attributes of hierarchy concepts (e.g., *Disease*) from the attributes of their subconcepts (e.g., *BrainDisorder* and *SkinDisease*).

Introduction

Background

Taking advantage of increasing amounts of publicly available text, Web-based information extraction generally focuses on the acquisition of instances and/or facts of predefined types. Since it is unfeasible to manually enumerate the instances and types of facts that may be relevant for all knowledge domains, current research efforts aim at collecting instances and relations of many different types and with minimal supervision (Banko et al. 2007). The inherent challenges of the task are illustrated by the difficulty of accurately pinpointing instances of complex types (e.g., book names, sayings etc.) within text documents (Downey, Broadhead, and Etzioni 2007); and the need to identify the most relevant relations of each instance out of many candidates (Davidov, Rappoport, and Koppel 2007).

Contributions

This paper introduces a weakly-supervised method for acquiring hierarchical open-domain class attributes from unstructured text. Initially, the attributes capture relevant properties (e.g., *side effects* and *maximum dose*) of a labeled class. A labeled class consists of a class label (e.g., *painkillers*) and a set of class instances (e.g., *vicodin*, *oxycontin*). Both the labeled classes and their attributes are extracted from a combination of Web documents and query logs. The extraction relies on a very small amount of supervision, in the form of a couple of Is-A extraction patterns widely used in information extraction literature (Hearst 1992) and as few as 5 seed attributes provided for only one class. The flat set of labeled classes is linked into existing conceptual hierarchies (e.g., *brain disorders* and *skin diseases* are linked under the concepts *BrainDisorder* and *SkinDisease* respectively). Thus, the attributes extracted for the labeled classes are propagated upwards in the hierarchy, for instance to determine the attributes of the hierarchy concept *Disease* from the attributes of its subconcepts (e.g., *BrainDisorder* and *SkinDisease*).

Our extraction method makes several contributions. First, it produces a flat set of more than 9,000 open-domain classes containing a total of around 200,000 instances. Although this is an intermediate result rather than final goal of the paper, it is significant since large sets of classes constitute useful resources for an array of applications (Pantel and Ravichandran 2004), including seed-based information extraction techniques. Second, the method extracts attributes for thousands of open-domain, automatically-acquired classes. The amount of supervision is limited to five seed attributes provided for only one reference class. In comparison, the largest previous study in attribute extraction reports results on a set of 40 manually-assembled classes, and requires five seed attributes to be provided as input for each class (Paşca 2007). Third, the method is the first to pursue the extraction of class attributes over conceptual hierarchies, rather than over a flat set of classes. A simple algorithm for propagating attributes along conceptual hierarchies is instrumental in generating attributes at precision levels of 0.64 at rank 10, 0.57 at rank 20, and 0.46 at rank 50, for open-domain concepts available within a widely-used language processing resource, namely WordNet (Fellbaum 1998). Fourth, the extraction method operates on a combination of both Web documents and search query logs, to acquire knowledge that is expected to be meaningful and suitable for later use. In contrast, the textual data sources used in previous studies in large-scale information extraction are either Web documents (Banko et al. 2007) or, recently, query logs (Paşca 2007), but not both.

Hierarchical Class Attribute Extraction

Extraction of Labeled Class Instances

Overview: Figure 1 shows how Web textual data is used to acquire hierarchical open-domain class attributes through the sequential extraction of: 1) open-domain, labeled classes of instances, by applying a few extraction patterns to unstructured text within documents, while guiding the extraction based on the contents of query logs (bottom-left in Figure 1); 2) class attributes that capture quantifiable properties of those classes, by mining query logs while guiding the ex-

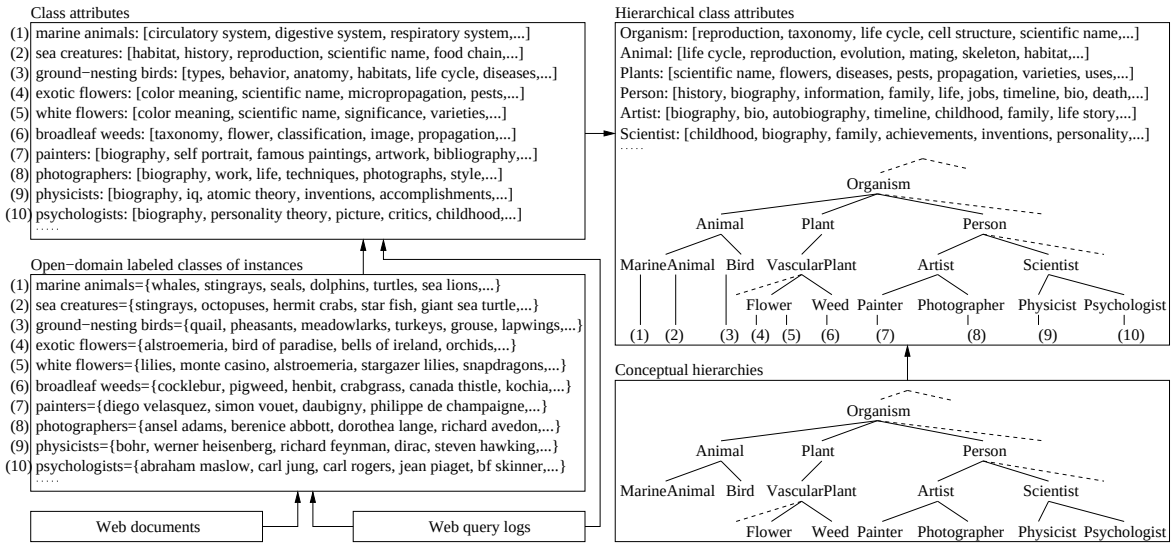

Figure 1: Overview of weakly-supervised extraction of hierarchical open-domain class attributes

traction based on a few attributes provided as seed examples (top-left in the figure); and 3) hierarchical class attributes, by propagating the attributes over existing conceptual hierarchies, after automatically linking labeled class instances under hierarchy concepts (top-right in the figure).

The extraction of labeled classes of instances introduces three innovations over previous work on extracting conceptual hierarchies from text (Hearst 1992; Snow, Jurafsky, and Ng 2006), with respect to: the number of extraction patterns; the use of query logs to uniformly handle the extraction of simple and complex instances; and the introduction of inexpensive heuristics for producing cleaner labeled classes.

Pattern-Based Extraction: For simplicity, the number of Is-A patterns is aggressively reduced to only two patterns, which can be summarized as:

$\langle$[..] $\mathcal{C}$ [such as|including] $\mathcal{I}$ [and|,|.]$\rangle$,

where $\mathcal{I}$ is a potential instance (e.g., *Venezuelan equine encephalitis*) and $\mathcal{C}$ is a potential class label for the instance (e.g., *zoonotic diseases*), for example in the sentence: *"The expansion of the farms increased the spread of zoonotic diseases such as Venezuelan equine encephalitis [..]"*.

Instance Filtering: Once a pattern match is found, the exact boundaries of the class label $\mathcal{C}$ and the instance $\mathcal{I}$ must be precisely identified within the document sentence. The class label is simply approximated from the part-of-speech tags of the sentence words, as a base (i.e., non-recursive) noun phrase whose last component is a plural-form noun. In the example sentence from above, the class label is *zoonotic diseases*, which consists of a plural-form noun and a preceding modifier. If no such phrase is found, the pattern match is discarded. In comparison, boundary detection for potential instances is more challenging, especially for phrases on which common-sense heuristics (e.g., English instances are short sequences of capitalized words) fail due to phrase length (e.g., for book and movie titles) or use of general-vocabulary, non-capitalized words (e.g., for sayings, proverbs and names of marine mammals). Intuitively, if an entity is prominent, Web search users will (eventually) ask about it. Thus, we hypothesize that relevant instances of any kind must occur as search queries containing an instance and nothing else. In practice, the right boundaries of the instances $\mathcal{I}$ in the extraction patterns are identified by simply checking that the sequence of words within the pattern that corresponds to the potential instance $\mathcal{I}$ can be found as an entire query in query logs. During matching, all string comparisons are case-insensitive. If no such query is found, the pattern match is discarded. Since most queries are typed in lower case by their users, the collected data is uniformly converted to lower case.

Class Label Filtering: Since the collected class labels are base noun phrases, their head nouns can be approximated as the last words within each class label. A heuristic identifies which head noun occurs most frequently across the potential class labels $\mathcal{C}$ of an instance $\mathcal{I}$, then discards the labels whose head nouns are not the most frequent head noun. For example, since the most frequent head of the labels associated with *australia* is *countries*, class labels such as *commonwealth countries* and *asia pacific countries* are retained, whereas *regional players*, *exporters* or *economic powers* are discarded. In the process, valid labels that are useful in describing the class of an instance may be discarded (e.g., *asia pacific nations* for the instance *australia*), thus promoting precision of the class labels at the expense of lower recall. Whereas reduced recall is an undesirable side effect, we feel that it only slightly diminishes the usefulness of the extracted data. Indeed, although it is straightforward to obtain lists of instances for the seven or so coarse-grained classes (Person, Organization, Location etc.) that were the focus of named entity extraction for decades, no such lists are readily available for hundreds (Talukdar et al. 2006), let alone thousands of diverse, fine-grained, open-domain classes covering the many domains of potential interest to Web search users.

After filtering, the resulting pairs of an instance and a label are arranged into instance sets (e.g., {*rabies*, *west nile virus*, *leptospirosis*,...}), each associated with a class label (e.g., *zoonotic diseases*).

Linking Labeled Classes to Conceptual Hierarchies

Manually-constructed language resources such as WordNet provide reliable, wide-coverage upper-level conceptual hierarchies, by grouping together phrases with the same mean-

ing (e.g., *analgesic*, *painkiller* and *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). Automatically-extracted labeled classes of instances lend themselves as natural candidates for extending the conceptual hierarchies available within WordNet and any other similar, hand-built resources, for two reasons. First, due to the effort required to manually maintain and extend its conceptual hierarchies, WordNet cannot systematically cover fine-grained concepts of specific domains. Some of the gaps under various concepts (e.g., Protein, Disorder and Antibiotic) can be easily filled with automatically-extracted class labels (e.g., *cellular proteins*, *behavioral disorders* and *oral antibiotics*). Second, WordNet is not meant to be an encyclopedic resource. Consequently, the automatically-extracted instances can supplement the instances encoded explicitly in WordNet, since the latter are either (rarely) exhaustive (e.g., 127 instances exist in WordNet for the concept *AfricanCountry*), or (sometimes) merely representative (e.g., there are 3 instances for *SearchEngine*), or (usually) completely missing (e.g., there are no instances for *CarMaker*).

To determine the points of insertion of automatically-extracted labeled classes under 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. Although the development of an intricate method for choosing the correct sense is theoretically possible, we employ the more practical heuristic of always selecting the first (that is, most frequent) sense of the label in WordNet as point of insertion. Our choice is motivated by three factors. First, WordNet senses are often too fine-grained, making the task of choosing the correct sense difficult even for humans (Palmer, Dang, and Fellbaum 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, Kasneci, and Weikum 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 that *companies* has in WordNet, which corresponds to companies as institutions created to conduct business. In the process, lexically distinct but semantically equivalent class labels (e.g., *marine animals* and *sea creatures*) and their associated instances may be judiciously inserted under the same WordNet concept that captures the shared meaning of the class labels.

Hierarchical Extraction of Class Attributes

Flat-Set Extraction: The labeled classes of instances collected automatically from Web documents are passed as input to the second extraction phase (top-left in Figure 1), which acquires class attributes by mining a collection of Web search queries. The attributes capture properties that are relevant to the class. The extraction of attributes exploits the set of class instances rather than the associated class label, and has four stages:

1) identification of a noisy pool of candidate attributes, as remainders of queries that also contain a class instance. In the case of the class *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 search-signature vector representations for each candidate attribute, based on queries (e.g., *"cast 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 "templatized" queries. The latter are automatically derived from the original queries, by replacing specific attributes and instances with common placeholders, e.g., *"X for Y"*;

3) construction of a reference internal search-signature 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 class (e.g., *movies*), by computing similarity scores between their individual vector representations and the reference vector of the seed attributes.

The result of the four stages is a ranked list of attributes (e.g., [*opening song*, *cast*,...]) for each class (e.g., *movies*).

In a departure from previous work, the instances of each input class are automatically generated as described earlier, rather than manually assembled. Furthermore, the amount of supervision is limited to seed attributes being provided for only one of the classes, whereas (Paşca 2007) requires seed attributes for each class. To this effect, the extraction includes modifications such that only one reference vector is constructed internally from the seed attributes during the third stage, rather one such vector for each class in (Paşca 2007); and similarity scores are computed cross-class by comparing vector representations of individual candidate attributes against the only reference vector available during the fourth stage, rather than with respect to the reference vector of each class in (Paşca 2007).

Hierarchical Propagation: As discussed in the previous section, the labeled classes and their associated sets of instances are linked under various concepts from existing conceptual hierarchies. Therefore, the attributes extracted from query logs for the labeled classes can be iteratively propagated upwards in the conceptual hierarchies. The computation proceeds from the bottom towards the top of the hierarchies. The formula that computes the score of some attribute $\mathcal{A}$ for an intermediate hierarchy concept $\mathcal{H}$ promotes attributes that have higher scores for more of the direct (i.e., non-inherited) hierarchy subconcepts $\mathcal{H}_C \subset \mathcal{H}$ of the concept $\mathcal{H}$:

$$S(Att(\mathcal{H}, \mathcal{A})) = \frac{\sum_{\mathcal{H}_C} S(Att(\mathcal{H}_C, \mathcal{A}))}{1 + |\{\mathcal{H}_C : \mathcal{H}_C \subset \mathcal{H}\}|}$$

The computed scores define the relative ranking of attributes for each hierarchy concept. As illustrated earlier in the top-right part of Figure 1, the ranked lists of attributes of higher-level hierarchy concepts such as *Organism* are thus computed from attributes of subconcepts such as *Plant* and *Person*, which are in turn computed from attributes of subconcepts such as *Artist* and *Scientist*.

The two-stage computation of attributes, for labeled classes and then for hierarchy concepts, is a practical implementation of two theoretical observations. First, a concept is traditionally a placeholder for a set of instances that share similar properties (Dowty, Wall, and Peters 1980). Therefore, the attributes of our labeled classes are computed

by analyzing candidate attributes of the respective class instances. Second, a superconcept captured properties that are common across its subconcepts (Dowty, Wall, and Peters 1980). In our case, the attributes of a hierarchy concept are iteratively computed from attributes of its subconcepts, starting from the bottom of the hierarchies where the labeled classes are linked under hierarchy concepts.

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 collection of queries is 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). The document collection consists of around 100 million documents in English, as available in a Web repository snapshot from 2006. The textual portion of the documents is cleaned of HTML, tokenized, split into sentences and part-of-speech tagged using the TnT tagger (Brants 2000).

Parameters for Extracting Labeled Classes: The extraction method collects labeled classes of instances from the input documents. During pattern matching, the instance boundaries are approximated by checking that the collected instances occur among to the top five million queries with the highest frequency within the input query logs. The extracted data is further filtered by discarding classes with fewer than 25 instances, and retaining the top 100 instances in each class. The labeled classes 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 Class Attributes: The amount of supervision for extracting attributes of labeled classes is limited to 5 seed attributes (*population*, *area*, *president*, *flag* and *climate*) provided for only one of the extracted labeled classes, namely *european countries*. Internally, the ranking of attributes uses the Jensen-Shannon divergence to compute similarity scores between internal representations of seed attributes, on one hand, and each of the candidate attributes, on the other hand. The top 50 attributes extracted for each class are retained for the upward propagation towards higher-level WordNet concepts under which the class labels are linked.

Evaluation

Quantitative Results

The extracted set of labeled classes consists of 9,537 class labels, each of them associated with 25 to 100 instances, with an average of 54 instances per class. The labeled classes exhibit great variation from the point of view of their popularity within query logs, measured by the sum of the frequencies of the input queries that fully match any of the instances of each class (e.g., the queries *british open* for *sports events*, or *san jose mercury news* for *newspapers*). As illustrated in Figure 2, the corresponding frequency sums per class vary considerably, ranging from 13.8 million at rank 1 for *sites*, down to 0.9 million at rank 1,501 for *hip-hop artists*. More importantly, the extracted classes cover a wide range of domains of interest such as health for *flu symptoms*, entertainment for *classic shows*, finance for *operating expenses*, or philosophy for *critical theorists*.

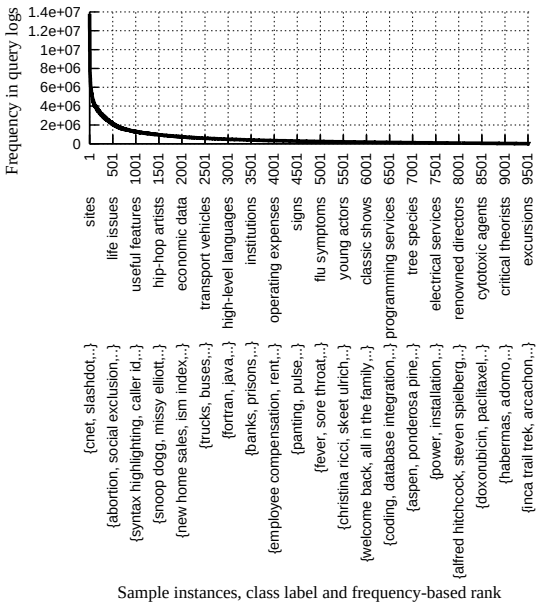

Figure 2: Popularity of the extracted labeled classes, measured by the aggregated frequency of queries that are full, case-insensitive matches of any of the instances in each class

Since some class labels (e.g., *isps* and *patent apps*) are not found in WordNet even after removing all their modifiers, only a subset of 9,519 of the 9,537 labeled classes, containing a total of 199,571 unique instances, are linked under WordNet concepts.

Qualitative Results

Experimental Runs: The experiments consist of six different runs, which correspond to different choices for the source of conceptual hierarchies and class instances linked to those hierarchies, as illustrated in Table 1. In the first two runs, N_1 and N_2, the class instances are those available within the latest version of WordNet (3.0) itself via HasInstance relations. N_1 is a subset of N_2, obtained by discarding instances that belong to concepts with fewer than 25 instances. In the third and fourth runs, K_1 and K_2, the class instances are available in an earlier version of WordNet (2.1) that was extended as part of previous work (Snow, Jurafsky, and Ng 2006). K_1 does not include the HasInstance instances already available in WordNet, whereas K_2 includes them. The last two runs from Table 2, E_1 and E_2, correspond to the fully-fledged extraction method described in this paper. In E_2, class labels are linked to the first sense available at the point of insertion in WordNet. The manual assessment of the points of insertion selected for a random sample of 100 class labels indicates a precision of 0.8. Comparatively, E_1 is stricter, as it discards class labels and their associated instances whenever multiple, rather than only one, senses exist in WordNet at the point of insertion.

Target Hierarchy Concepts: The performance of attribute extraction is computed over a set of target concepts chosen to contain a large enough number of concepts (25) to properly ensure varied experimentation on several dimensions, while taking into account the time intensive nature of man-

Description	Source of Hierarchy and Instances					
	N_1	N_2	K_1	K_2	E_1	E_2
WordNet version	3.0	3.0	2.1	2.1	3.0	3.0
Discard inst. of ambiguous concepts?	-	-	-	-	√	-
Discard small (≤25 inst.) instance sets?	√	-	-	-	√	√
Include inst. from WordNet?	√	√	-	√	-	-
Include inst. from elsewhere?	-	-	√	√	√	√
Total instances ($\times 10^3$)	13.5	17.4	110.2	127.6	48.0	199.5
Total classes	136	945	2,465	3,078	1,422	9,519

Table 1: Source of conceptual hierarchy and class instances for various hierarchical attribute extraction runs

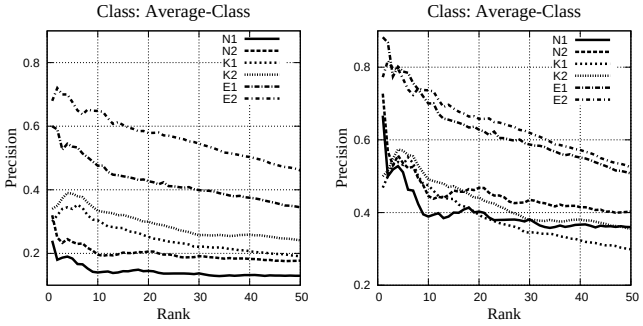

Figure 3: Accuracy of the attributes extracted for various runs, as an average over the entire set of 25 target concepts (first graph) and as an average over (variable) subsets of the 25 target concepts for which some attributes were extracted in each run (second graph)

ual accuracy judgments often required in the evaluation of information extraction systems (Banko et al. 2007). 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 is mapped into exactly one WordNet concept (synset). For instance, 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"*. [1] 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, Sekine, and Grishman 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.

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 3 plots the precision at ranks 1 through 50 for the ranked lists of attributes extracted by various runs. In the first graph from the figure, 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 second graph, only target concepts for which some attributes were extracted are included in the precision scores of a given run. Thus, the first graph properly penalizes a run for failing to extract any attributes for some target concepts, whereas the second graph does not include any such penalties.

Several conclusions can be drawn after inspecting the results. First, the more restrictive runs (N_1, K_1 and E_1) have lower precision scores across all ranks in the first graph of Figure 3 than their than less restrictive counterparts (N_2, K_2 and E_2 respectively). In other words, adding more restrictions may improve precision but hurts recall of class instances, which results in lower average precision scores for the attributes. Second, N_1 and N_2 have the lowest precision scores in the first graph, which is in line with the relatively small number of instances available in the original WordNet, as discussed earlier and confirmed by the counts from Table 1. Third, the runs using our automatically-extracted labeled classes (E_1 and E_2) clearly outperform not only N_1 and N_2, but also K_1 and K_2. Concretely, in the left graph, the precision scores at rank 10 are 0.33 for K_2, 0.47 (that is, a 42% improvement over K_2) for E_1 and 0.64 (94% improvement over K_2) for E_2. The total counts of unique instances and classes listed in Table 1 for the six runs may suggest that the superior performance with automatically-extracted classes (E_1 and E_2) is simply due to the higher coverage of those runs, which must reduce the number of concepts for which no attributes can be extracted. However, this is not the case. Indeed, the precision scores are higher for E_2 than for K_1 and K_2 in the second graph as well, that is, even without taking into account target concepts without any extracted attributes. In fact, even the precision scores for the more restrictive run E_1 are higher across all ranks than the precision of both K_1 and K_2, for both the first and the second graph, although the coverage of E_1 is clearly smaller than the coverage of K_1 and K_2 as shown in Table 1. The different levels of attribute accuracy can be explained by the different quality and usefulness of the input class instances in the various

[1] The target concepts are also uniquely mapped to WordNet 2.1 synsets whose internal offsets are different but are semantically equivalent to their WordNet 3.0 counterparts, with respect to component phrases, associated definitions, and localization within the conceptual hierarchy.

runs. In the case of N_1 and N_2, although the class instances are theoretically perfect since they are part of the manually-created WordNet, their usefulness in practice suffers due to the ambiguity of the instances. For example, WordNet contains *constable* (as an alternative to *john constable*) as an instance of a painter, and *buena vista* as an instance of a pitched battle, which are certainly valid when considered in the context of WordNet, but are ambiguous in isolation, resulting in spurious matches over search queries during attribute extraction. On the other hand, the instances available within the pre-existing WordNet extension (Snow, Jurafsky, and Ng 2006) behind the runs K_1 and K_2 are sometimes listed for the wrong concepts. For instance, *laura serrano* and *mario miranda* are incorrectly listed in K_1 and K_2 as instances of a boxer in the sense of a dog breed, whereas *advanced micro devices* and *david chase* occur as instances of makers or creators in the religious sense. The automatically-extracted labeled classes in E_1 and E_2 are certainly not free of errors either, but when aggregated over many classes, they do offer more useful representative instances of various concepts, at least for the task of class attribute extraction.

Comparison to Previous Results

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. Under those conditions, the accuracy of attributes extracted from text reaches 0.90 at rank 10, 0.85 at rank 20, and 0.76 at rank 50, when measured over selected flat sets of classes (Paşca 2007). In contrast, the current method operates on automatically-extracted, open-domain classes of instances. Furthermore, by dropping the requirement of manually providing a small set of seed attributes for each target class, and relying on only a few seed attributes specified for one reference class, we acquire class attributes without the need of first determining what the classes should be, what instances they should contain, and from which resources the instances should be collected. To our knowledge, the method presented in this paper is the first to pursue the extraction of class attributes over conceptual hierarchies, rather than over a flat set of classes. As such, it is related to previous work on ontologizing relations acquired from text (Pennacchiotti and Patrick 2006), which focused on PartOf and CauseOf relations rather than class attributes.

Conclusion

This paper introduces an extraction framework for mining a combination of both documents and search query logs. Web-derived labeled classes of instances allow for the extraction of attributes over existing conceptual hierarchies, without a-priori restrictions to specific domains of interest and with very little supervision. The quality of the extracted hierarchical attributes reaches 0.64 at rank 10 and 0.46 at rank 50, and thus exceeds the quality of attributes extracted based on previous resources of class instances. Current work aims at assigning various attributes to their most appropriate locations within the conceptual hierarchies, rather than computing ranked lists of attributes for each hierarchy concept.

Acknowledgments

The author thanks Razvan Bunescu for comments on an earlier version of the paper, and Bijun He for emergency technical assistance before the initial submission.

References

Banko, M.; Cafarella, M. J.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artifi - cial Intelligence (IJCAI-07)*, 2670–2676.

Brants, T. 2000. TnT - a statistical part of speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, 224–231.

Davidov, D.; Rappoport, A.; and Koppel, M. 2007. Fully unsupervised discovery of concept-specifi c relationships by Web mining. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, 232–239.

Downey, D.; Broadhead, M.; and Etzioni, O. 2007. Locating complex named entities in Web text. In *Proceedings of the 20th International Joint Conference on Artifi cial Intelligence (IJCAI-07)*, 2733–2739.

Dowty, D.; Wall, R.; and Peters, S. 1980. *Introduction to Montague Semantics*. Springer.

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

Gao, W.; Niu, C.; Nie, J.; Zhou, M.; Hu, J.; Wong, K.; and Hon, H. 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)*, 463–470.

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

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

Paş ca, M. 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)*, 101–110.

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

Pantel, P., and Ravichandran, D. 2004. Automatically labeling semantic classes. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, 321–328.

Pennacchiotti, M., and Patrick, P. 2006. Ontologizing semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, 793–800.

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

Snow, R.; Jurafsky, D.; and Ng, A. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, 801–808.

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

Talukdar, P.; Brants, T.; Liberman, M.; and Pereira, F. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, 141–148.