

Acquisition of Open-Domain Classes via Intersective Semantics

Marius Paşca
1600 Amphitheatre Parkway
Mountain View, California 94043
mars@google.com

ABSTRACT

A weakly-supervised method acquires fine-grained class labels that do not occur verbatim in the input data or underlying text collection. The method generates more specific class labels (*gold mining companies listed on the toronto stock exchange*) that capture the semantics of the underlying classes, out of pairs of input class labels (*companies listed on the toronto stock exchange*, *gold mining companies*) available for an instance (*Golden Star Resources*). When applied to Wikipedia articles and their categories, the method generates new categories for existing articles, and expands existing categories with additional articles.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; I.2.6 [Artificial Intelligence]: Learning

Keywords

Open-domain information extraction; fine-grained classes; knowledge acquisition; class labels

1. INTRODUCTION

Background: Semantic classes encoded in knowledge repositories are often lexicalized through class labels. They conveniently summarize the properties shared among the instances of each class. Consequently, research efforts aiming at compiling knowledge about instances emphasize the acquisition of as many relevant, lexicalized class labels as possible.

That class labels play a prominent role in open-domain knowledge acquisition is supported by the presence of class labels in popular, human-compiled resources that encode open-domain instances. Depending on the resource, the class labels may take different forms. Class labels may be

members of synonym sets in hypernym classes available for each instance in WordNet [16]; or categories available in each article in Wikipedia; or types available for each Freebase topic [5, 30]. However, the presence of class labels in otherwise distinct methods and resources serves comparable goals. They compactly describe the meaning of the underlying semantic class, by effectively summarizing some of the relevant properties of the instance; serve as conceptual bridges among instances sharing common properties; and allow for the organization of instances and their classes into ontologies or hierarchies. The importance of class labels is also illustrated by the growing emphasis on open-domain information extraction [14]. Its goal is to extract open-domain classes, instances and relations from textual data [3, 7, 52]. Concretely, it extracts sets of instances [43, 36, 50, 22] associated with thousands or more [48, 53, 13, 24] open-domain class labels. In open-domain extraction, the input text collection may be a large set of Web documents, in which case an extracted class label is a contiguous piece of text (phrase, title, column heading) from some document. Alternatively, the input may be human-compiled data, in which case an extracted class label is one of thousands of pre-existing WordNet synsets [44], Wikipedia categories [45, 31, 28] or Freebase types [30, 46]. Subsequently, the connectivity between existing instances, on one hand, and existing class labels, on the other hand, may be further increased by transferring class labels of an instance to its related instances [46, 25].

Motivation: With existing extraction methods, for a class label to be output for one or more instances, the class label must necessarily occur verbatim in the input data or text collection. But even in very large text collections, specific (as opposed to general) class labels like “*non-alcoholic mexican beverages*”, “*hip hop female singers*”, “*gold mining companies listed on the toronto stock exchange*” are rarely, if ever, mentioned verbatim for any instances. In the relatively few cases when they are mentioned, they may not be associated with their instances (*Jumex*, *Tamala Jones*, *Golden Star Resources*) in ways that lend themselves to the extraction of the respective pairs of an instance and its class label by current algorithms. Consequently, instances encoded in current knowledge resources or extracted with current extraction methods offer limited coverage of specific class labels.

Contributions: This paper is the first to explore the role of what could be referred to as intersective semantics in the extraction of open-domain information in general, and class labels of open-domain instances in particular. It intersects pairwise input instance sets associated with more

general class labels (“*companies listed on the toronto stock exchange*”, “*gold mining companies*”), into output instance sets associated with more specific class labels (“*gold mining companies listed on the toronto stock exchange*”). Unlike in previous work, new class labels are generated rather than merely selected from existing class labels. As such, they are not required to occur verbatim in the input data or underlying text collection. Experimental results over Wikipedia demonstrate that previously-unseen class labels can be generated at specificity levels unmatched by previous methods. Precision over randomly selected output samples of an instance and its generated class label reaches 70%. It increases to 85%, when the generated class labels are further intersected with a large set of Web search queries. Generated class labels provide additional coverage, in scenarios where class labels stored for an instance are made available directly, for example to enhance Web search results with structured data. Class labels also constitute additional evidence in tasks such as answering list-seeking queries [2], or in answer typing [38] in question answering.

2. GENERATING CLASS LABELS

Intersective Semantics: A class label is a descriptive phrase employed to name a semantic class, where the semantic class contains a set of instances sharing common properties. Perhaps because instances and classes are particularly important building blocks in any knowledge repository, class labels are frequently studied and extracted from text separately [48, 50, 46, 53] from other properties and relations that may apply to the same instances and classes. But class labels, on one hand, and properties and relations, on the other hand, are not the distinct, unrelated pieces of knowledge about a class that previous work may suggest they are. On the contrary, we argue that the naming of class labels is strongly influenced by the very properties shared among the instances of the class:

Hypothesis 1: Let S be a semantic class, $\{I\}$ its set of instances, and $\{P\}$ the properties shared among the instances. A class label C of the class S serves as a compact, lexical approximation of the properties $\{P\}$. The name of a more accurate (i.e., specific) class label C of the class S summarizes more, ideally all, of the properties $\{P\}$.

Consider the instances within the set $\{\text{Croatia, Algeria, Greece, Italy, Slovenia, Tunisia, Spain, ..}\}$. They share various properties: they are all *locations* in the world - more precisely, they are all *countries*, and all have a border with the Mediterranean Sea. Out of multiple class labels, including “*locations*”, “*countries*”, “*locations bordering the mediterranean sea*” or “*countries bordering the mediterranean sea*”, the latter is one of the more specific that best summarize the properties of the instances.

The addition of new properties to a class specializes it into a more specific class. The naming of its class label takes into account both old properties and new properties:

Hypothesis 2: Let P' be a new property applied to the set of instances $\{I\}$ of the class S . The subset of instances $\{I'\} \subset \{I\}$ that satisfies the additional property P' corresponds to a more specific class S' . The name of a class label C' of the class S' can be obtained from the class label C of S , by adding a reference to the property P' .

Considering the above example, if the instance set is restricted to the items located in Europe, then the instance set shrinks to $\{\text{Croatia, Greece, Italy, Slovenia, Spain, ..}\}$. The

```

Input: instance  $\mathcal{I}$ 
.   pair of class labels  $C_1, C_2$  of  $\mathcal{I}$ 
.   large repository of Web search queries  $\{\mathcal{Q}\}$ 
Output: set of generated class labels  $\{\mathcal{G}\}$  of  $\mathcal{I}$ 
Variables:  $\mathcal{F}_1, \mathcal{F}_2$  = prefix of  $C_1, C_2$ 
.    $\mathcal{X}_1, \mathcal{X}_2$  = infix of  $C_1, C_2$ 
.    $\mathcal{O}_1, \mathcal{O}_2$  = postfix of  $C_1, C_2$ 
.    $\{\mathcal{P}_1\}, \{\mathcal{P}_2\}$  = set of subphrases of  $C_1, C_2$ 
.    $\mathcal{Y}_1, \mathcal{Y}_2$  = subphrase of  $C_1, C_2$ 
.    $\mathcal{G}_A, \mathcal{G}_B, \mathcal{G}_C, \mathcal{G}_D, \mathcal{G}_X$  = concatenations of
.   prefixes, infixes, postfixes of  $C_1$  and  $C_2$ 

Steps:
00.  $\{\mathcal{G}\} = \emptyset$ 
01.  $\mathcal{X}_1 = []$ ;  $\mathcal{X}_2 = []$ 
02.  $\mathcal{H}_1 = \text{HeadToken}(C_1)$ ;  $\mathcal{H}_2 = \text{HeadToken}(C_2)$ 
03.  $\{\mathcal{P}_1\} = \text{SubphrasesWithHead}(C_1, \mathcal{H}_1)$ 
04.  $\{\mathcal{P}_2\} = \text{SubphrasesWithHead}(C_2, \mathcal{H}_2)$ 
05. For  $\mathcal{Y}_1 \in \{\mathcal{P}_1\}$  and  $\mathcal{Y}_2 \in \{\mathcal{P}_2\}$ 
06.   If  $\text{CompatiblePhrases}(\mathcal{Y}_1, \mathcal{Y}_2)$ 
07.      $\mathcal{X}_1 = \text{LongestPhrase}(\mathcal{X}_1, \mathcal{Y}_1)$ 
08.      $\mathcal{X}_2 = \text{LongestPhrase}(\mathcal{X}_2, \mathcal{Y}_2)$ 
09. If  $(\mathcal{X}_1 \neq [])$  and  $(\mathcal{X}_2 \neq [])$ 
10.    $[\mathcal{F}_1, \mathcal{X}_1, \mathcal{O}_1] = \text{SplitToPrefixInfixPostfix}(C_1, \mathcal{X}_1)$ 
11.    $[\mathcal{F}_2, \mathcal{X}_2, \mathcal{O}_2] = \text{SplitToPrefixInfixPostfix}(C_2, \mathcal{X}_2)$ 
12.   For  $\mathcal{X}$  in  $\{\mathcal{X}_1, \mathcal{X}_2\}$ 
13.      $\mathcal{G}_A = \text{Concat}(\text{Concat}(\mathcal{F}_1, \mathcal{F}_2), \mathcal{X}, \text{Concat}(\mathcal{O}_1, \mathcal{O}_2))$ 
14.      $\mathcal{G}_B = \text{Concat}(\text{Concat}(\mathcal{F}_2, \mathcal{F}_1), \mathcal{X}, \text{Concat}(\mathcal{O}_1, \mathcal{O}_2))$ 
15.      $\mathcal{G}_C = \text{Concat}(\text{Concat}(\mathcal{F}_1, \mathcal{F}_2), \mathcal{X}, \text{Concat}(\mathcal{O}_2, \mathcal{O}_1))$ 
16.      $\mathcal{G}_D = \text{Concat}(\text{Concat}(\mathcal{F}_2, \mathcal{F}_1), \mathcal{X}, \text{Concat}(\mathcal{O}_2, \mathcal{O}_1))$ 
17.      $\mathcal{G}_X = \text{MostReadablePhrase}(\mathcal{G}_A, \mathcal{G}_B, \mathcal{G}_C, \mathcal{G}_D)$ 
18.     If  $\text{PlausibleClassLabel}(\mathcal{G}_X, \{\mathcal{Q}\})$ 
19.       Insert  $\mathcal{G}_X$  in  $\{\mathcal{G}\}$ 
20. Return  $\{\mathcal{G}\}$ 

```

Figure 1: Generic merging of pairs of class labels into generated class labels

class label of the new, more specific class can be obtained from the class label of the more general class “*countries bordering the mediterranean sea*”, by adding a reference to the new property: “*european countries bordering the mediterranean sea*”.

Hypothesis 3: Let S_1 and S_2 be two semantic classes, $\{I_1\}$ and $\{I_2\}$ their sets of instances, and $\{P_1\}$ and $\{P_2\}$ the properties shared among the instances. A more specific class S_{12} , specializing classes S_1 or S_2 or both, can be constructed by intersecting the instance sets $\{I_1\} \cap \{I_2\}$, which will be associated with the union of properties $\{P_1\} \cup \{P_2\}$. The name of the class label C_{12} of the class S_{12} approximates the properties $\{P_1\} \cup \{P_2\}$. The name can be generated by merging the class labels C_1 of S_1 , with C_2 of S_2 .

Consider two classes, whose instance sets are $\{\text{Croatia, Algeria, Greece, Italy, Slovenia, Tunisia, Spain, ..}\}$, with the class label “*countries bordering the mediterranean sea*”; and $\{\text{Croatia, Germany, Poland, Greece, Serbia, Italy, Slovenia, Portugal, Spain, Denmark, ..}\}$, with the class label “*european countries*”. A more specific class can be created from the pair of more general classes, by intersecting their instance sets into $\{\text{Croatia, Greece, Italy, Slovenia, Spain, ..}\}$; and merging their class labels into “*european countries bordering the mediterranean sea*”.

Merging Flat-Set Class Labels: A class consists in a set of instances associated with class labels. Intuitively, to construct a more specific class out of a pair of given classes, the two instance sets are intersected; and the two class labels are merged into a more specific class label. Figure 1 presents an algorithm for merging pairs of class labels of an instance, into generated class labels applying to the instance.

In Steps 2 to 4, the input class labels are converted into sets of spans $\{\mathcal{P}_1\}, \{\mathcal{P}_2\}$. A span is a phrase within the class

label, whose head token is also the head token of the class label. The head token is derived from the syntactic parse tree of the class label.

Steps 5 to 8 identify two infixes $\mathcal{X}_1, \mathcal{X}_2$ among the spans of each of the two input class labels C_1, C_2 . The infixes $\mathcal{X}_1, \mathcal{X}_2$ are computed as the longest spans $\mathcal{V}_1 \in \{\mathcal{P}_1\}, \mathcal{V}_2 \in \{\mathcal{P}_2\}$ within the two class labels, which are deemed compatible with each other. The compatibility between two spans estimates whether it is appropriate to merge them together. It is a particular case of semantic relatedness [41, 32]. It can be approximated in multiple ways. A stricter approximation would require the phrases to be identical (“1962 [films]” vs. “[films] by french directors”) or synonymous (“silent [films]” vs. “award-winning [movies]”). Conversely, a looser approximation would require the spans to subsume each other (“zulu-language [films]” vs. “[documentary films] about women”; or “ethiopian [women]” vs. “living [people]”; or “1980s [establishments]” vs. “[technology companies]”), or be more loosely related phrases (“[prelates]” vs. “romanian anti communist [clergy]”).

In Steps 10 and 11, the infixes split each of the two input class labels C_1, C_2 into an infix $\mathcal{X}_1, \mathcal{X}_2$ preceded by a prefix $\mathcal{F}_1, \mathcal{F}_2$ and followed by a postfix $\mathcal{O}_1, \mathcal{O}_2$. The prefix and postfix may be empty.

If compatible infixes have been identified within the two input class labels, a new class label is generated from the class labels in Steps 12 to 19. The infix of the generated class label is one of the infixes of the input class labels. The prefix and postfix of the generated class label are obtained by pairwise merging of the prefixes and postfixes of the input class labels respectively. Specifically, the generated prefix is one out of two possible concatenations of the input prefixes; and the generated postfix is one out of two possible concatenations of the input postfixes. For example, for the input class labels “silent [films]” and “award-winning [movies]”, the generated prefix is one of *silent award-winning* or *award-winning silent*, the generated infix is *films* or *movies*, and the generated postfix is empty. Thus, Steps 13 to 16 simply consider the four possible concatenations $\mathcal{G}_A, \mathcal{G}_B, \mathcal{G}_C, \mathcal{G}_D$ of a generated prefix followed by a generated infix followed by a generated postfix. The most readable of these concatenations is retained as the generated class label \mathcal{G}_X in Step 17. The readability of a concatenation is estimated with an ngram language model [6] compiled from a collection of Web documents. As an alternative to more complex smoothing strategies, the computation of language-model scores relies on a simple backoff strategy (cf. [6]). The concatenation \mathcal{G}_X with the highest language-model score is selected over the other possible concatenations in Step 17. Note that one of the four possible concatenations is always selected in Step 17, even if the entire concatenation does not occur in its entirety in the language-model training data. If the concatenation \mathcal{G}_X is deemed to be a plausible class label in Step 18, then it is retained in the set of generated class labels $\{\mathcal{G}\}$. Concretely, the presence of a class label as a full-length query in the set of queries $\{\mathcal{Q}\}$ can be treated as an indication that the class label is plausible, since Web users found it to be relevant enough to submit it as a search query. The use of search queries for estimating plausibility imposes practical bounds on the specificity of class labels that can be generated. Most search queries are relatively short. Therefore, very long class labels are unlikely to be generated, since they are not likely to occur even within a very large set of search

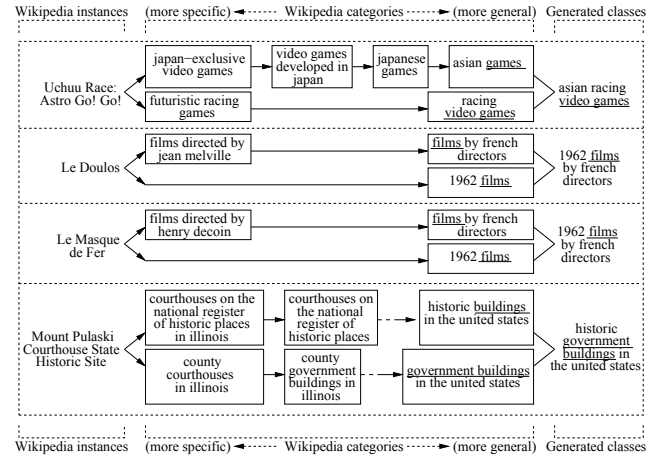


Figure 2: Merging pairs of Wikipedia parent categories and ancestor categories of an instance, into generated class labels of the instance. One of the infixes of the categories being merged, shown underlined, is selected as infix of the generated class label. Parent categories are Wikipedia categories listed at the bottom of a Wikipedia article. Ancestor categories are parent categories or, recursively, parent categories of parent categories.

queries. Alternative methods for assessing plausibility may be considered, such as a threshold on the language-model scores computed in the previous step.

Merging Hierarchical Class Labels: The algorithm in Figure 1 applies directly to input class labels available for each instance simply as a flat set of class labels. But if input class labels are organized hierarchically, the algorithm equally applies to both “parent” class labels, and to “ancestor” class labels that recursively generalize the input class labels in the hierarchy. For each pair of input class labels, additional pairs of their more general class labels are also considered. For example, in addition to merging “*prelates*” and “*romanian anti communist clergy*”, the algorithm also attempts to merge one of “*prelates*”, “*christian religious leaders*”, “*religious leaders*”, with one of “*romanian anti communist clergy*”, “*romanian clergy*”, “*romanian people*”.

Generating Wikipedia Categories: The method for generating class labels can be applied to extend any knowledge resource that associates human-readable class labels to various instances. Among such sources, one of the more popular ones is Wikipedia. In this case, an input class label is an existing Wikipedia category, whereas a generated class label corresponds to a generated Wikipedia category. Previous studies already illustrate Wikipedia’s role in knowledge acquisition [45, 31, 54, 52] and information retrieval [21, 42]. Our contribution related to Wikipedia is to infer new, finer-grained categories, from categories already listed at the bottom of Wikipedia articles. When compared to other knowledge resources, Wikipedia exhibits a few peculiar features, such as the presence of infoboxes. In order to avoid dependence on features specific to Wikipedia, neither infoboxes nor the body of Wikipedia articles are used.

The left part of Figure 2 shows examples of Wikipedia instances, i.e., titles of Wikipedia articles (e.g., *Uchuu Race: Astro Go! Go!*). In the figure, instances are connected to the

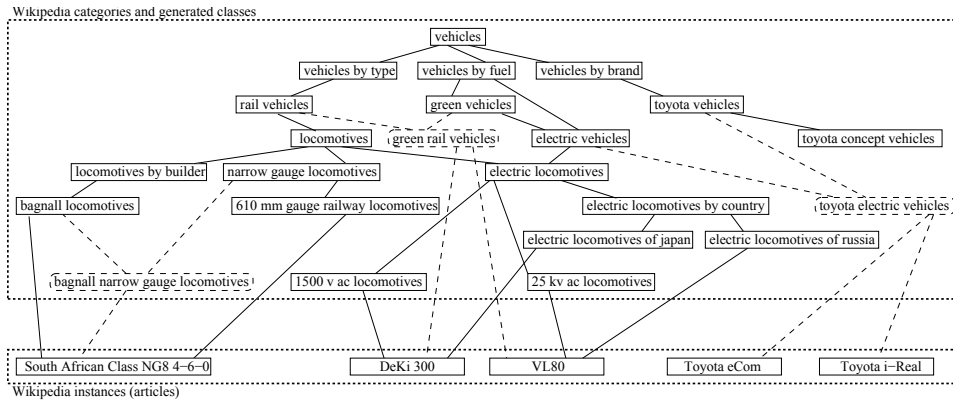


Figure 3: Acquisition of generated class labels for individual Wikipedia articles, from pairs of parent and ancestor categories listed in Wikipedia for each article. Parent categories are Wikipedia categories listed at the bottom of a Wikipedia article. Ancestor categories are parent categories or, recursively, parent categories of parent categories (solid boxes in upper part of the figure=Wikipedia categories; solid boxes in lower part of the figure=Wikipedia instances; solid edges between boxes=instance-to-category or category-to-category links within Wikipedia; dashed boxes=generated class labels; dashed edges below dashed boxes=added links between a Wikipedia instance and a generated class label; dashed edges above dashed boxes=added links between a generated class label and one of the two Wikipedia categories from which it was generated)

right to their Wikipedia parent categories (“*japan-exclusive video games*”) and, further to the right, to incrementally more general ancestor categories derived from the Wikipedia category network [40] (“*video games developed in japan*”, “*japanese games*”, “*asian games*”). Ancestor categories are parent categories or, recursively, parent categories of parent categories in Wikipedia. The right part of the figure shows new class labels (new Wikipedia categories) obtained from pairs of ancestor categories. For example, “*asian games*” and “*racing video games*” are merged into “*asian racing video games*”, by merging the infixes *games* and *video games* of the two class labels respectively.

Although new categories are generated independently from one Wikipedia instance to another, the same category may be generated for multiple instances. For example, “*1962 films by french directors*” is generated for both *Le Doulos* and *Le Masque de Fer* in Figure 2. In turn, generated categories become part of, and extend, the larger Wikipedia category network. In Figure 3, generated categories (e.g., “*toyota electric vehicles*”) are connected downwards to their Wikipedia instances (*Toyota eCom*, *Toyota i-Real*, etc.), and upwards to the Wikipedia categories from which they are created (“*electric vehicles*”, “*toyota vehicles*”).

3. EXPERIMENTAL SETTING

Raw Data Sources: The experiments benefit from three sources of raw textual data: a snapshot of all Wikipedia articles in English, as available in May 2012; a sample of around 200 million Web documents in English; and around 1 billion anonymized Web search queries in English.

Derived Data Sources: The raw content of Wikipedia articles is distilled into mappings from an article title (e.g., *Mount Pulaski Courthouse State Historic Site*) to the set of categories listed at the bottom of the article (e.g., “*government buildings completed in 1848*”, “*abraham lincoln national heritage area*”, “*county courthouses in illinois*”). The mappings from a sample of 50 articles are inspected during development, and therefore removed from evaluations of

accuracy. In all mappings, the categories are syntactically parsed [37] to identify their head tokens. Consistently with treatment in previous work [40], Wikipedia categories containing any of the subphrases *article(s)*, *category(ies)*, *infobox(es)*, *pages*, *redirects*, *stubs*, *templates*, *wikiproject* and *use mdy dates* are deemed to have internal bookkeeping as sole purpose, and therefore are discarded. Categories like “*abraham lincoln national heritage area*”, whose head tokens are not plural-form nouns, are less likely to form good class labels. As in [40], such categories are discarded.

The Wikipedia category network, containing mappings from more specific to more general Wikipedia categories, is separately collected from categories listed at the bottom of Wikipedia category pages (e.g., from the page titled *Category:Government buildings completed in 1848*).

Unstructured text within the set of Web documents is the source for an ngram language model using a simple-backoff smoothing strategy [6], and for a phrase similarity repository derived following [27, 34]. The repository contains ranked lists of up to 1000 most distributionally similar phrases, for each of around 1 million phrases. For example, the top distributionally similar phrases for *video games* are *videogames*, *computer games*, *violent video games*, *games*, *comic books*, *electronic games* etc. The similarity between two phrases is the cosine similarity of their context feature vectors [27].

Using the phrase similarity repository, a separate, expanded set of possible queries is created out of the original set of Web search queries. Following [33], replacements of ngrams within queries, with their most similar phrases, produce other possible queries whose plausibility is checked against the input set of queries. The expanded set of queries contains around 125 billion queries. Either the original set of queries or the expanded set of queries can be used as the input set of queries, in the algorithm in Figure 1.

Extraction Parameters: The generic steps in the algorithm from Figure 1 are instantiated as follows. Infixes of class labels being merged are deemed compatible in Step 6, if the infixes are distributionally similar to each other in

Run	Class Labels	I per C		Run	Class Labels	I per C	
		A	M			A	M
R_W	472,237	310	8	$R_{G \cap \neg W}$	4,576,369	10	2
$R_{W \cap Q}$	136,259	579	15	$R_{G \cap \neg W \cap Q}$	33,207	35	3
$R_{W \cap S}$	272,694	416	10	$R_{G \cap \neg W \cap S}$	389,878	27	2

Table 1: Number of (unique) class labels extracted in various runs (I per C=number of instances per class label; A=average; M=median)

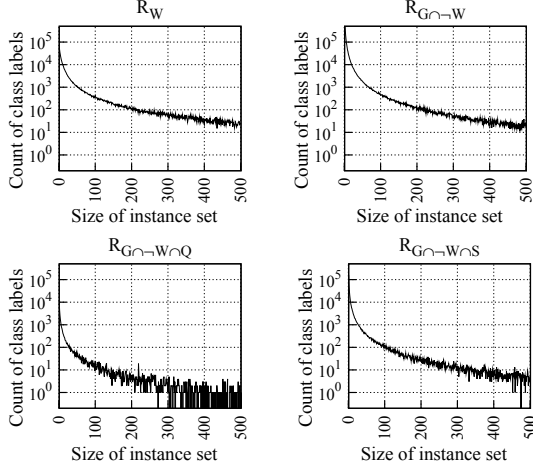


Figure 4: Distribution of the size of instance sets, shown as count of class labels whose instance set has a given size. Computed over source categories from Wikipedia (first graph) vs. generated class labels that are not Wikipedia categories (remaining graphs)

the phrase similarity repository; and if they share the same head token (e.g., *video games* and *games*). As for generated class labels, they are deemed plausible in Step 18 if they are full-length queries in the set of queries. The algorithm is implemented as a sequence of MapReduce [10] stages over the Wikipedia snapshot.

Experimental Runs: Experimental runs correspond to combinations of constraints that must be satisfied by the class labels extracted or generated from Wikipedia data. Individual constraints are denoted as follows: W , to require class labels to be in the set of existing categories from Wikipedia (and, on the flip side, $\neg W$ to require class labels to be other than existing Wikipedia categories); G , to generate class labels with our method, without any checks for plausibility of generated class labels; Q , to require class labels to be queries from the original input set of queries; S , to require class labels to be queries from the expanded set of queries. These individual constraints are combined into several experimental runs, in particular:

- R_W , a baseline run that simply outputs categories already listed in Wikipedia for the various Wikipedia instances, without further modifications;
- R_G , an aggressive (over-generative) run that generates class labels without any checks for plausibility;
- $R_{G \cap W}$ and $R_{G \cap \neg W}$, obtained from R_G by restricting generated class labels to those that are (W) or are not ($\neg W$) existing Wikipedia categories;

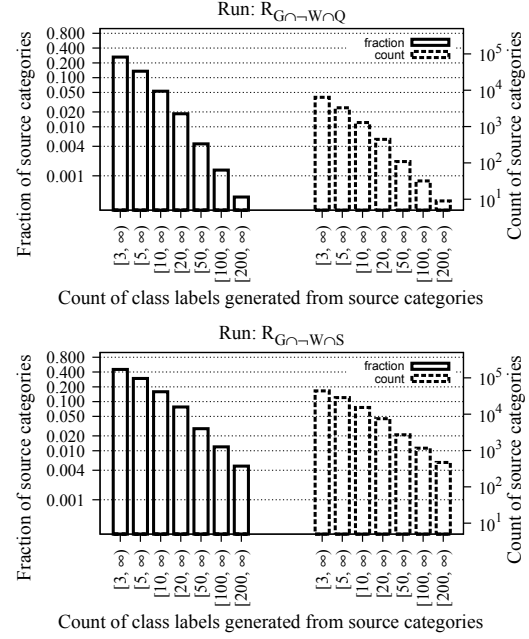


Figure 5: Distribution of source categories, shown as fraction (solid lines, left axis) and count (dotted lines, right axis) of source categories for which the count of generated class labels falls under particular intervals. Counts of class labels are computed over generated class labels that are not Wikipedia categories, and are either original queries (upper graph) or extended queries (lower graph). For example, 10 class labels or more are generated from each of 5.3% (1,292) source categories, in the upper graph; or from each of 15.8% (15,312), in the lower graph

- $R_{G \cap Q}$ and $R_{G \cap S}$, more conservative runs than R_G that require generated class labels to be original queries (Q) or expanded queries (S);

- $R_{G \cap \neg W \cap Q}$ and $R_{G \cap \neg W \cap S}$, runs that further restrict plausible generated class labels to those that are not among existing Wikipedia categories already.

4. EVALUATION RESULTS

Relative Coverage: As illustrated in Table 1, the number of new, generated class labels (run $R_{G \cap \neg W}$) is an order of magnitude higher than the number of categories available in Wikipedia (run R_W) from which they are generated. If class labels are intersected with the sets of queries, 25% or 150% additional class labels are generated on top of existing class labels, when intersecting with the original set Q ($R_{W \cap Q}$ vs. $R_{G \cap \neg W \cap Q}$) or expanded set of queries S ($R_{W \cap S}$ vs. $R_{G \cap \neg W \cap S}$) respectively. This corresponds to increasing coverage by the respective percentages, in a scenario where list-seeking queries were answered directly with instances associated with class labels that fully match the queries.

Since generated class labels are finer-grained, they are associated with fewer instances on average than their source categories from Wikipedia. Figure 4 provides a more detailed view into the distribution of class labels in various runs, from the point of view of the size of their instance sets.

Figures 5 and 6 illustrate the “productivity” of source categories from Wikipedia, in contributing to the generation of

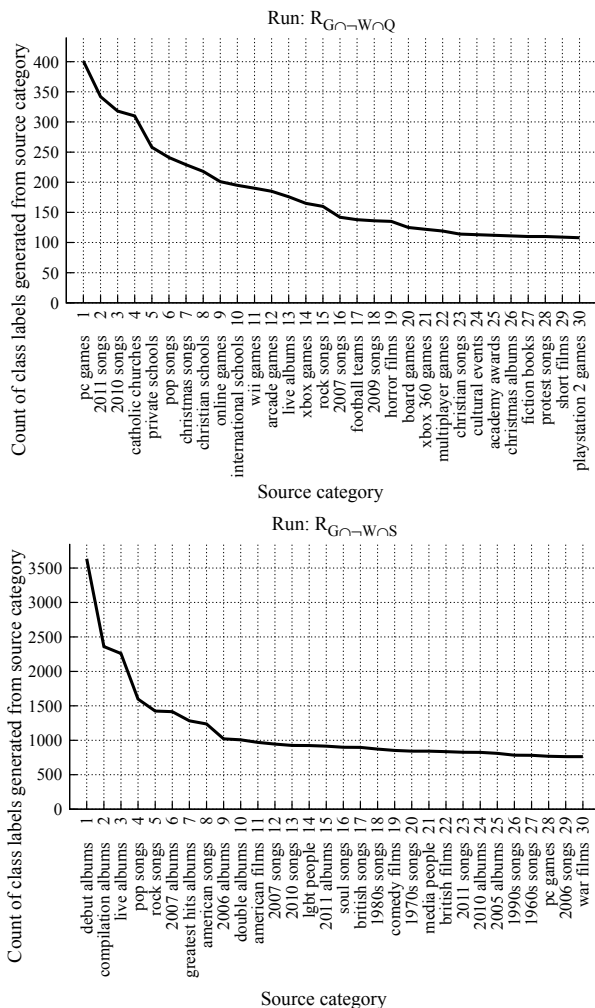


Figure 6: Top source categories from Wikipedia, whose intersection with some other Wikipedia categories produces the highest number of generated class labels. Counts are computed over generated class labels that are not Wikipedia categories, and are either original queries (first graph) or extended queries (second graph)

class labels. The graphs in Figure 5 show the fraction and count of source categories, on the vertical axis, that generate a number of class labels within a particular interval, on the horizontal axis. More class labels are generated, when they are required to appear in the expanded set S as opposed to original set Q of queries. The same phenomenon is visible in Figure 6. The top 30 most productive Wikipedia categories, which contribute to generating the largest number of class labels, lead to 401 (“*pc games*”) down to 108 (“*playstation 2 games*”) generated class labels for $R_{G \cap W \cap Q}$, vs. 3,634 (“*debut albums*”) down to 763 (“*war films*”) for $R_{G \cap W \cap S}$. **Recovering Existing Instances:** Although its goal is to generate new class labels, the method may generate class labels that are themselves Wikipedia categories. For example, merging the two input Wikipedia categories “*high schools in nevada*” and “*private schools in nevada*” for the instance *The Meadows School* produces “*private high schools in nevada*”. The latter is already a category of *The Meadows School* in

Sample of Class Labels
11th-century natural disasters, 1971 christmas albums, 1976 elections in germany, aerospace museums in iowa, agriculture companies of the united kingdom, american people of madeiran descent, buddhist monasteries in france, cambodian documentary films, china records live albums, defunct airlines of sierra leone, guatemalan people of italian descent, history books about central america, ice hockey people from west virginia, kenyan people of czech descent, market towns in wales, music museums in austria, naval battles of world war i involving japan, open air museums in canada, pakistani people of dutch descent, private high schools in nevada, religious museums in thailand, slovenian expatriates in china, solar power stations in japan, student protests in australia, tangerine dream live albums, technology museums in ohio, toll bridges in massachusetts, ucla bruins baseball coaches, uruguayan computer scientists, vietnam war aircraft carriers of the united states

Table 2: Ability of generated class labels to recover instances already available for them in Wikipedia. Illustrated through a random sample of generated class labels, whose instance sets include all instances already available in Wikipedia for the respective class labels

Wikipedia. Such cases are an opportunity to measure coverage, with respect to the ability of generated class labels to recover instances already available for them in Wikipedia.

To illustrate, consider the Wikipedia category “*private schools in nevada*”. It has two instances in Wikipedia, both of which are also among the four instances to which the generated class label “*private schools in nevada*” is associated. Separately, the Wikipedia category “*austrian computer scientists*” has ten instances in Wikipedia, seven of which are present among the nine instances of the generated class label with the same name “*austrian computer scientists*”. Therefore, the generated class labels “*austrian computer scientists*” and “*private schools in nevada*” recover 70% (7 out of 10) and 100% (2 out of 2) respectively, of the instances already available for them in Wikipedia. In particular, the set of instances available in Wikipedia for the category “*private schools in nevada*” is completely recovered by the generated class label with the same name.

Table 2 shows additional examples of Wikipedia categories whose sets of instances are completely recovered by the generated class labels. More generally, only a fraction of the instances available in Wikipedia categories are recovered via generated class labels, as illustrated in Table 3. Over the set of generated class labels that are Wikipedia categories, the average proportion of Wikipedia instances recovered by generated class labels is 63.3%.

Adding New Instances: Over the same set of generated class labels that are Wikipedia categories, Table 4 measures the complementary phenomenon of adding new instances not already available in Wikipedia for the respective class labels. In this context, a new instance is one that already exists in Wikipedia, but is associated with categories other than the particular class label being considered. For example, the set of instances associated with the category “*technology companies of japan*” in Table 4 increases from only one instance (*Neurowear*) already available in Wikipedia, to 306 instances (e.g., *AliceSoft*, *Now Production*) after generating class labels. Over generated class labels that are Wikipedia

Rcvr. Int.	Frac. Cl.	Rcvr. Int.	Frac. Cl.
[0.0, 0.0]	0.044	[0.6, 0.8)	0.132
(0.0, 0.2)	0.130	[0.8, 1.0)	0.124
(0.2, 0.4)	0.117	[1.0, 1.0]	0.323
(0.4, 0.6)	0.131		

Table 3: Ability of generated class labels to recover instances already available for them in Wikipedia. Illustrated through the fraction of generated class labels that are Wikipedia categories, that fall under various intervals of instance recovery coverage ratios. For each generated class label, the recovery coverage ratio is computed as the ratio between the number of instances already available in Wikipedia that are recovered by the generated class label, and the number of instances already available in Wikipedia (Rcvr. Int.=interval over instance recovery coverage ratios; Frac. Cl.=fraction of class labels)

#	Class Label	Cvg./↗
1	english christian religious leaders	1,012×
2	2010s rock songs	634×
3	irish christian religious leaders	570×
4	european people of irish descent	508×
5	victorian-era ships of the united kingdom	435×
6	christian organizations based in the united states	340×
7	infantry divisions of world war ii	321×
8	technology companies of japan	306×
9	6th-century european people	279×
10	7th-century european people	247×
11	international educational organizations	212×
12	merchant ships of scotland	166×
13	turkish people of spanish descent	148×
14	service companies of the soviet union	134×
15	italian catholic bishops	131×

Table 4: Ability of generated class labels to increase coverage (Cvg./↗) by generating instances not already available for them in Wikipedia. Illustrated through the list of generated class labels that contain the most new instances, relative to the instances already available in Wikipedia for the respective class labels (×=how many times the number of new instances is larger than the number of instances already available in Wikipedia)

categories, the average ratio of Wikipedia instances newly added by generated class labels to the respective categories is 72.8%.

Accuracy of Generated Class Labels: Since new pairs of a generated class label and an instance are by definition not already in Wikipedia, their quality cannot be assessed automatically relative to Wikipedia. A separate gold standard dedicated to the task is not available either. Instead, in accordance with other previous studies on open-domain information extraction [49, 52, 53, 28], extraction samples are manually inspected.

Random pairs of a generated class label and an instance are manually annotated as *correct*, *questionable* or *incorrect*. Examples of questionable pairs are “*cultural events in italy*” for *Alessandria Challenger*, which is a sports event; and “*british explorers of north america*” for *John Franklin*,

Run	Precision				
	Correctness Counts				Score
	T	C	Q	I	
$R_{G\cap W}$	200	119	45	36	0.707
$R_{G\cap W\cap Q}$	200	172	6	22	0.875
$R_{G\cap W\cap S}$	200	179	1	20	0.897

Table 5: Accuracy computed over random samples of pairs of a generated class label that is not a Wikipedia category, and one of the instances of the class label. For each run, the sample of pairs is drawn by first selecting a weighted random sample of 200 class labels, where the weight is the size of (i.e., number of instances in) a class label; then selecting one random instance for each class label (T=total; C=correct; Q=questionable; I=incorrect)

Run	Precision				
	Correctness Counts				Score
	T	C	Q	I	
$R_{G\cap W}$	200	161	9	30	0.827

Table 6: Accuracy computed over a random sample of pairs of a generated class that is a Wikipedia category, and one of the instances of the class label. The sample of pairs is drawn by first selecting a weighted random sample of 200 class labels, where the weight is the size of (i.e., number of instances in) a class label; then selecting one random instance for each class label, where the instance is not already available in Wikipedia as an instance for the class label (T=total; C=correct; Q=questionable; I=incorrect)

who was a person born in England who explored Canada. To compute the accuracy over a set of annotated pairs, the correctness labels *correct*, *questionable* and *incorrect* are converted to the numeric values 1.0, 0.5 and 0 respectively. Precision over the set is the sum of the correctness values, divided by the size of the set. Note that alternative metrics like the cumulative gain and its discounted variants [23] could also be measured, if the evaluation data took the form of ranked lists rather than sets.

Precision is computed for two different scenarios: generated class labels that are not Wikipedia categories, and generated class labels that are Wikipedia categories. The first scenario corresponds to creating new Wikipedia categories and populating them from scratch with instances, in Table 5. The accuracy of generated class labels is 0.707, 0.875 or 0.897, depending on whether generated class labels are not checked for plausibility (run $R_{G\cap W}$) or are required to be original queries (run $R_{G\cap W\cap Q}$) or extended queries (run $R_{G\cap W\cap S}$). As expected, runs $R_{G\cap W\cap Q}$ and $R_{G\cap W\cap S}$ have higher precision than run $R_{G\cap W}$. More interestingly, runs $R_{G\cap W\cap Q}$ and $R_{G\cap W\cap S}$ have similar precision. In other words, the increase in coverage brought by run $R_{G\cap W\cap S}$ relative to $R_{G\cap W\cap Q}$, illustrated earlier in Table 1, does not cause a drop in precision.

The second scenario for computing precision corresponds to populating existing Wikipedia categories with additional instances, in Table 6. The accuracy for generated class labels is 0.827 ($R_{G\cap W}$). Thus, around 8 out of every 10 newly added instances to existing Wikipedia categories are correct.

Pairs of (Generated Class Label: Instance)
Generated class labels are Wikipedia categories:
(african-american rock singers: Gary U.S. Bonds); (canadian rock musicians: Martha Johnson (singer)); (entertainment companies of japan: Inti Creates); (hip hop double albums: Paid in Full (soundtrack)); (manufacturing companies of lebanon: Farra Design Center); (music publishing companies of germany: Kitty-Yo); (roman catholic schools in australia: Good Shepherd Catholic College, Mount Isa); (syrian people of british descent: Asma al-Assad)
Generated class labels are not Wikipedia categories:
(2007 wii games: Monster Trux: Arenas); (australian indie musicians: Calerway); (british colonial infantry regiments: 10th Jats); (charter high schools in the united states: George Walton Comprehensive High School); (fantasy novel characters: Old Man Willow); (hip hop female singers: Tamala Jones); (metal mining companies: Zhaojin Mining); (non-alcoholic mexican beverages: ToniCol); (stealth xbox games : The Great Escape (2003 video game))

Table 7: Sample of correct class labels generated for various Wikipedia instances and not already available as Wikipedia categories for the respective instances

Table 7 gives examples of correct extractions for both the first scenario and the second scenario.

Discussion: Aggressively generating class labels in $R_{G \cap W}$ increases the number of Wikipedia categories from Table 1 by a factor of 10. Whether their precision of 0.707 in Table 5 is adequate depends on the application. Requiring that generated class labels be queries results in the more conservative runs $R_{G \cap W \cap Q}$ and $R_{G \cap W \cap S}$. In particular, the intersection of generated class labels with the expanded set of queries proves particularly useful. It allows for generated categories to roughly double (more precisely, increase by 82%) the number of Wikipedia categories in Table 1, while retaining a precision score above 0.80 in Table 5.

The correctness of class labels is guaranteed in Table 6, where the class labels are existing Wikipedia categories; but not in Table 5. Nevertheless, the accuracy of runs $R_{G \cap W \cap Q}$ and $R_{G \cap W \cap S}$ in Table 5 is higher than for $R_{G \cap W}$ in Table 6. Two factors contribute to this situation. First, requiring the generated class labels to be original (Q) or expanded (S) queries has positive impact on the actual correctness of class labels generated in $R_{G \cap W \cap Q}$ and $R_{G \cap W \cap S}$. Second, it may be more difficult to populate an existing class label with additional instances beyond the more popular ones that are already known, as in $R_{G \cap W}$; than it is to populate a new class label with its instances from scratch, as in $R_{G \cap W \cap Q}$ and $R_{G \cap W \cap S}$.

Error Analysis: Several types of errors, causing incorrect class labels to be generated for some instances, are more frequently encountered in the experiments.

In the first type of errors, the input Wikipedia categories, from which class labels are incorrectly generated, are themselves incorrect. Although Wikipedia data is manually created and verified, the Wikipedia category network has errors with respect to how it hierarchically organizes Wikipedia categories. In some cases, the organization is not in fact hierarchical [40]. In Wikipedia, the category “*amtrak stations in utah*” is linked under “*amtrak stations*”, “*amtrak stations*” is linked under “*railway stations by company*”, and “*railway*

stations by company” is linked under “*railway companies*”. The first two links are proper hierarchical (subsumption, or IsA) links, but the last link is not. In other cases, even when links are hierarchical, they may still be incorrect due to what one could refer to as OR-subsumption (as opposed to AND-subsumption). To our knowledge, this phenomenon has not been reported in previous work using or extending Wikipedia [40, 45, 52, 32]. The OR-subsumption phenomenon consists in more specific categories being linked under a set of more general categories, where the rather unusual semantics seems to be that only one link or another, but not all, hold simultaneously for any instance of the more specific categories. Examples include the category “*people of the tudor period*”. It is linked under “*15th-century irish people*”, “*15th-century english people*” and “*15th-century welsh people*”, among other categories. Clearly, the meaning of these links is that an instance of “*people of the tudor period*” can be an instance of “*15th-century irish people*” or “*15th-century english people*” or “*15th-century welsh people*”, but not all of them simultaneously. Since OR-links are not distinctly marked as such in Wikipedia, they are treated similarly to other links. They cause the inclusion of incorrect input class labels like “*15th-century irish people*” for the instance *William Finch (merchant)*.

The second type of errors consists in merging input categories that are individually correct, but produce incorrect generated class labels for various reasons. In some cases, the input class labels refer to incompatible properties that do not make much sense in combination. For example, the instance *Hub Pernoll* is listed in Wikipedia under the categories “*portland beavers players*” and “*aberdeen grays players*”. Similarly, the instance *Kirk Douglas* is listed under “*cecil b. demille award golden globe winners*” and “*academy award winners*”. But it is incorrect to generate the class labels “*aberdeen harbor grays portland beavers players*” or “*cecil b. demille award academy award winners*”. Note that the same class labels could be judged as more useful, or at least more readable, if conjunctions were added (e.g., “*aberdeen harbor grays and portland beavers players*”). In other cases, incorrect infixes may be selected, as in merging “*elementary schools in british columbia*” and “*high schools in british columbia*” into “*elementary high schools in british columbia*”.

5. RELATED WORK

In a confirmation of their key role in any attempt at representing, organizing and serving world knowledge, class labels and/or their instances are the backbone of knowledge resources. The application of our method to Wikipedia is transitively beneficial to other resources derived from it, including DBpedia [4], Yago [45, 20] and Freebase [5].

The acquisition of open-domain classes, instances and relations has attracted much attention [26, 1, 17, 22, 52, 18, 29]. In previous methods for extracting sets of instances associated with class labels [3, 50, 25], the sets of instances and/or class labels are organized as flat sets or hierarchically, relative to inferred hierarchies [24] or manual hierarchies such as WordNet [44] or the category network within Wikipedia [51, 39]. Semi-structured text from Web documents is a complementary resource to unstructured text, for the purpose of extracting relations in general [7], and class labels and their instances in particular [47, 9]. Comparatively, our method does not require any separate collections of Web documents, whether unstructured or semi-

Method: Constraints on Extracted Class Labels
Class labels are confined to a restricted vocabulary:
[46]: Existing class labels, such as Wikipedia categories, of some other instance(s)
[44]: Elements of WordNet synonym sets
[28]: Wikipedia categories
Class labels are not confined to a restricted vocabulary:
[53]: Contiguous phrases in text documents; match a Hearst[19] pattern with their instances
[24]: Contiguous phrases in text documents; match a doubly-anchored Hearst[19] pattern with their instances
[35]: Contiguous phrases in text documents; occur in the proximity of their instances

Table 8: Constraints satisfied by class labels extracted by a sample of extraction methods

structured, as a necessary data source from which class labels and their instances are extracted. Instead, our method simply merges class labels already available for an instance, into new class labels of that instance.

The vocabulary of class labels potentially assigned to an instance by previous methods may be restricted or unrestricted. In the case of a restricted vocabulary, the vocabulary of class labels is confined to a closed set provided manually as input [50, 8]. The set is often derived from Wikipedia or similar resources [45, 46, 28]. New pairs of a class label and an instance are extracted by expanding the sets of instances associated with each class label [48, 50, 34, 55], with additional related instances; or by propagating class labels from an instance to its related instances over a graph [46]. Relatedness is often estimated via distributional similarities [34, 25]. For a class label to be newly assigned to an instance, the class label must already be present in the input data for some other instance. In contrast, our method often generates class labels not present in the input data for any of the instances. In the case of an unrestricted vocabulary, the vocabulary of class labels is acquired itself along with the instances [35, 44, 3, 48, 24, 15]. The extracted class labels take the simpler syntactic form of head nouns preceded by modifiers, e.g., *cities*, *european cities* [13]; *artists*, *strong acids* [35]; *outdoor activities*, *prestigious private schools* [48]; *methaterians*, *aquatic birds* [24]. Part of the reason is that class labels and other phrases that are relatively complex nouns are known to be difficult to detect and pick out precisely from surrounding text [12]. In contrast, the class labels extracted in our method exhibit greater syntactic variation, including but not limited to head nouns preceded by modifiers. Some of the practical constraints exhibited by class labels extracted with various methods are summarized in Table 8. To our knowledge, none of the previous methods assigns new class labels to existing instances with existing class labels, where the new class labels are not already existing class labels of some other instances. Furthermore, previous methods do not produce class labels as specific as, e.g., *“10th-century serbian executed reigning monarchs”*, *“1883 ships built in the san francisco bay area”*, *“packaging companies listed on the zagreb stock exchange”* and others generated by our method.

A number of previous studies specifically address the expansion of instances and/or their class labels available within knowledge resources. Examples include the filtering and reorganization of existing class labels, and the expansion of

their sets of instances in the case of Wikipedia [40], WordNet [44] or both [45, 39], as well as Freebase [46]. For example, [44] links new instances acquired from unstructured text to existing WordNet concepts, whereas [46] propagates existing class labels among semantically-related Freebase instances. None of these studies acquire previously-unseen class labels. For example, class labels that are newly propagated to an instance in [46] must have been present verbatim in the input data for some other instance. The acquisition of fine-grained class labels and their instances is related to typed search [11] or entity search [2] where class labels submitted as queries may be answered by instances available for the class labels. It is also related to answering list queries (*“what is the list of drought-tolerant plants from venezuela”*).

The extraction method in [31] analyzes Wikipedia categories, with the goal of turning implicit relations that they may encode into explicit ones. Among other extracted relations, [31] may extract relations of the same type as in this paper, namely IsA relations between a Wikipedia instance and a class label (more precisely, a Wikipedia category) not already available for the respective instance in Wikipedia. There are several differences between the two methods. First, the vocabulary of class labels in [31] is restricted to the categories already available in Wikipedia, whereas our method generates class labels that sometimes are, but usually are not already available in Wikipedia. Second, categories added to an instance in [31] are more general than the source categories from which they are inferred. For example, the category *“albums”* is inferred for the instance *Kind of Blue*, based on the availability of the parent category *“miles davis albums”* and its own parent category *“albums by artist”* in Wikipedia. In comparison, our method merges pairs of existing categories into generated class labels that are more specific than existing categories. Third, the method in [31] is tailored and applies only to Wikipedia, as it makes specific assumptions about the edges in the category network within Wikipedia. In comparison, our method is equally applicable to class labels from sources other than Wikipedia. Concretely, it applies to any sources that associate multiple class labels to various instances. It does require such sources to satisfy a few constraints. At least some of their class labels must be complex nouns rather than simple nouns (e.g., *“software companies”* rather than *“companies”*), such that they can be merged with one another. Furthermore, enough of their instances must be associated with multiple class labels rather than with only one, such that pairs of such class labels can be formed and considered for merging. If the class labels of some source are not organized hierarchically, then our method still applies to that source, but does not attempt to merge class labels that recursively generalize other class labels in the hierarchy.

An opposite approach to generating fine-grained class labels from existing class labels might be to first reduce existing class labels to non-overlapping, “atomic” class labels. Such atomic class labels would later be potentially mixed and merged, in response to user queries asking for fine-grained class labels. However, such an approach would incur prohibitive latency associated to inferring new class labels from existing ones on the fly. It would also merely compute a list of ranked candidate instances in response to a fine-grained class label, rather than explicitly asserting that certain instances do in fact apply to a certain fine-grained class label. In fact, to our knowledge, none of the previ-

ous extraction methods attempts to reduce extracted class labels to atomic class labels either. Instead, they may, and often strive to, extract lexically overlapping class labels, such as “software companies”, “german companies”, “german software companies” for the same instance.

6. CONCLUSION

A common strategy for scaling up the acquisition of open-domain knowledge is to take advantage of larger input data sources. This paper proposes an alternative strategy: taking advantage of knowledge that was already extracted, to infer additional knowledge. Disparate pieces of evidence (i.e., input class labels) are merged together into finer-grained assertions (generated class labels) than what could be potentially extracted from even large text collections. Current work explores the connection between class labels of an instance, on one hand, and relations and properties of an instance, on the other hand; the generation of iteratively more specific class labels, by applying the algorithm from Figure 1 in multiple iterations; and more accurately identifying pairs of input class labels that are not compatible and therefore would generate incorrect class labels.

7. REFERENCES

- [1] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM Conference on Knowledge Discovery and Data Mining (KDD-07)*, pages 76–85, San Jose, California, 2007.
- [2] K. Balog, M. Bron, and M. de Rijke. Category-based query modeling for entity search. In *Proceedings of the 32nd European Conference on Information Retrieval (ECIR-10)*, pages 319–331, Milton Keynes, United Kingdom, 2010.
- [3] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2670–2676, Hyderabad, India, 2007.
- [4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the Web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.
- [5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 International Conference on Management of Data (SIGMOD-08)*, pages 1247–1250, Vancouver, Canada, 2008.
- [6] T. Brants, A. Popat, P. Xu, F. Och, and J. Dean. Large language models in machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-07)*, pages 858–867, Prague, Czech Republic, 2007.
- [7] M. Cafarella, A. Halevy, D. Wang, E. Wu, and Y. Zhang. WebTables: Exploring the power of tables on the Web. In *Proceedings of the 34th Conference on Very Large Data Bases (VLDB-08)*, pages 538–549, Auckland, New Zealand, 2008.
- [8] A. Carlson, J. Betteridge, R. Wang, E. Hruschka, and T. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the 3rd ACM Conference on Web Search and Data Mining (WSDM-10)*, pages 101–110, New York, 2010.
- [9] B. Dalvi, W. Cohen, and J. Callan. Websets: Extracting sets of entities from the Web using unsupervised information extraction. In *Proceedings of the 5th ACM Conference on Web Search and Data Mining (WSDM-12)*, pages 243–252, Seattle, Washington, 2012.
- [10] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI-04)*, pages 137–150, San Francisco, California, 2004.
- [11] G. Demartini, T. Iofciu, and A. de Vries. Overview of the INEX 2009 Entity Ranking track. In *Initiative for the Evaluation of XML Retrieval Workshop*, pages 254–264, Brisbane, Australia, 2009.
- [12] D. Downey, M. Broadhead, and O. Etzioni. Locating complex named entities in Web text. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2733–2739, Hyderabad, India, 2007.
- [13] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the Web: an experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.
- [14] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 3–10, Barcelona, Spain, 2011.
- [15] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, pages 1535–1545, Edinburgh, Scotland, 2011.
- [16] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press, 1998.
- [17] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32nd International Conference on Research and Development in Information Retrieval (SIGIR-09)*, pages 267–274, Boston, Massachusetts, 2009.
- [18] R. Gupta and S. Sarawagi. Joint training for open-domain extraction on the Web: Exploiting overlap when supervision is limited. In *Proceedings of the 4th ACM Conference on Web Search and Data Mining (WSDM-11)*, Hong Kong, China, 2011.
- [19] M. Hearst. 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, 1992.
- [20] J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence. Special Issue on Artificial Intelligence, Wikipedia and Semi-Structured Resources*, 194:28–61, 2013.
- [21] J. Hu, G. Wang, F. Lochovsky, J. Sun, and Z. Chen. Understanding user’s query intent with Wikipedia. In *Proceedings of the 18th World Wide Web Conference (WWW-09)*, pages 471–480, Madrid, Spain, 2009.
- [22] A. Jain and M. Pennacchiotti. Open entity extraction from Web search query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 510–518, Beijing, China, 2010.
- [23] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.
- [24] Z. Kozareva and E. Hovy. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*, pages 1110–1118, Cambridge, Massachusetts, 2010.
- [25] Z. Kozareva, K. Voevodski, and S. Teng. Class label enhancement via related instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, pages 118–128, Edinburgh, Scotland, 2011.
- [26] D. Lin. Automatic retrieval and clustering of similar words.

- In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98)*, pages 768–774, Montreal, Quebec, 1998.
- [27] D. Lin and X. Wu. Phrase clustering for discriminative learning. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 1030–1038, Singapore, 2009.
- [28] T. Lin, Mausam, and O. Etzioni. No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-12)*, pages 893–903, Jeju Island, Korea, 2012.
- [29] F. Mesquita, J. Schmidek, and D. Barbosa. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP-13)*, pages 447–457, Seattle, Washington, 2013.
- [30] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 1003–1011, Singapore, 2009.
- [31] V. Nastase and M. Strube. Decoding Wikipedia categories for knowledge acquisition. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1219–1224, Chicago, Illinois, 2008.
- [32] R. Navigli and S. Ponzetto. BabelRelate! a joint multilingual approach to computing semantic relatedness. In *Proceedings of the 26th National Conference on Artificial Intelligence (AAAI-12)*, pages 108–114, Toronto, Canada, 2012.
- [33] M. Pasca. Asking what no one has asked before: Using phrase similarities to generate synthetic web search queries. In *Proceedings of the 20th International Conference on Information and Knowledge Management (CIKM-11)*, Glasgow, United Kingdom, 2011.
- [34] P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 938–947, Singapore, 2009.
- [35] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting 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)*, pages 113–120, Sydney, Australia, 2006.
- [36] M. Pennacchiotti and P. Pantel. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 238–247, Singapore, 2009.
- [37] S. Petrov, P. Chang, M. Ringgaard, and H. Alshawi. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*, pages 705–713, Cambridge, Massachusetts, 2010.
- [38] C. Pinchak, D. Lin, and D. Rafiei. Flexible answer typing with discriminative preference ranking. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, pages 666–674, Athens, Greece, 2009.
- [39] S. Ponzetto and R. Navigli. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 2083–2088, Pasadena, California, 2009.
- [40] S. Ponzetto and M. Strube. 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, 2007.
- [41] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th World Wide Web Conference (WWW-06)*, pages 377–386, Edinburgh, Scotland, 2006.
- [42] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. Topical clustering of search results. In *Proceedings of the 5th ACM Conference on Web Search and Data Mining (WSDM-12)*, pages 223–232, Seattle, Washington, 2012.
- [43] S. Sekine and H. Suzuki. Acquiring ontological knowledge from query logs. In *Proceedings of the 16th World Wide Web Conference (WWW-07), Posters*, pages 1223–1224, Banff, Canada, 2007.
- [44] R. Snow, D. Jurafsky, and A. Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 801–808, Sydney, Australia, 2006.
- [45] F. Suchanek, G. Kasneci, and G. Weikum. 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, 2007.
- [46] P. Talukdar and F. Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 1473–1481, Uppsala, Sweden, 2010.
- [47] P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pages 582–590, Honolulu, Hawaii, 2008.
- [48] B. Van Durme and M. Pasca. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1243–1248, Chicago, Illinois, 2008.
- [49] B. Van Durme, T. Qian, and L. Schubert. Class-driven attribute extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, pages 921–928, Manchester, United Kingdom, 2008.
- [50] R. Wang and W. Cohen. Automatic set instance extraction using the Web. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 441–449, Singapore, 2009.
- [51] F. Wu and D. Weld. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the 17th World Wide Web Conference (WWW-08)*, pages 635–644, Beijing, China, 2008.
- [52] F. Wu and D. Weld. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 118–127, Uppsala, Sweden, 2010.
- [53] W. Wu, H. Li, H. Wang, and K. Zhu. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the 2012 International Conference on Management of Data (SIGMOD-12)*, pages 481–492, Scottsdale, Arizona, 2012.
- [54] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, and M. Ishizuka. Unsupervised relation extraction by mining Wikipedia texts using information from the Web. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 1021–1029, Singapore, 2009.
- [55] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. Wen. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th World Wide Web Conference (WWW-09)*, pages 101–110, Madrid, Spain, 2009.