Acquisition of Noncontiguous Class Attributes from Web Search Queries

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

Abstract

Previous methods for extracting attributes (e.g., *capital*, *population*) of classes (*Empires*) from Web documents or search queries assume that relevant attributes occur verbatim in the source text. The extracted attributes are short phrases that correspond to quantifiable properties of various instances (*ottoman empire*, *roman empire*, *mughal empire*) of the class. This paper explores the extraction of noncontiguous class attributes (*manner (it) claimed legitimacy of rule*), from fact-seeking and explanation-seeking queries. The attributes cover properties that are not always likely to be extracted as short phrases from inherently-noisy queries.

1 Introduction

Motivation: Resources such as Wikipedia (Remy, 2002) and Freebase (Bollacker et al., 2008) aim at organizing knowledge around classes (*Food ingredients*, *Astronomical objects*, *Religions*) and their instances (*wheat flower*, *uranus*, *hinduism*). Due to inherent limitations associated with maintaining and expanding human-curated resources, their content may be incomplete. For example, attributes representing the *energy* (or *energy per 100g*) or *solubility in water* are available in both Wikipedia and Freebase for many instances of *Food ingredients* (e.g., for *olive oil*, *honey*, *fennel*). But the attributes are missing for some instances (e.g., *cornmeal*). Moreover, structured information about *how long (it) lasts unopened* or *manner (it) helps in weight loss* is generally missing for *Food ingredients*, from both resources. Such information is also often absent from among the attributes acquired from either documents or queries by previous extraction methods (Paşca et al., 2007; Van Durme et al., 2008). Previously extracted attributes tend to be short, often nominal, phrases like *nutritional value* and *taste*. Even when extracted attributes are not nominal (Paşca, 2012), they remain relatively short phrases such as *good for skin*. As such, previous attributes have limited ability to capture the finer-grained properties being asked about in queries such as *"how long does olive oil last unopened"* and *"how does honey help in weight loss"*. The presence of such queries suggests that such information is relevant to Web users. Identifying noncontiguous properties, or attributes of interest to Web users, helps filling some of the gaps in existing knowledge resources, which otherwise could not be filled by attributes extracted with previous methods.

Contributions: The contributions of this paper are twofold. First, it introduces a method for the acquisition of noncontiguous class attributes, from fact or explanation-seeking Web search queries like *"how long does olive oil last unopened"* or *"how does honey help in weight loss"*. The resulting attributes are more diverse than, and therefore subsume, the scope of attributes extracted by previous methods. Indeed, previous methods are unlikely to extract attributes as specific as *length/duration (it) lasts unopened* and *manner (it) helps in weight loss*, for the instances *olive oil* and *honey* of the class *Food ingredients*. Conversely, previously extracted attributes like *nutritional value* and *solubility in water* are roughly equivalent to the finer-grained *nutritional value (it) has* and *reason (it) dissolves in water*, extracted from the queries *"what nutritional value does honey have"* and *"why does glucose dissolve in water"* respectively. Second, the noncontiguous attributes can be simultaneously interpreted as binary relations pertaining to instances and classes. The relations (*helps in weight loss*) connect an instance (*honey*) or, more generally, a class (*Food ingredients*), on one hand; and a loosely-typed unknown argument (*manner*) whose value is of interest to Web users, on the other hand. Because

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 386–394,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

Web users already inquire about the value of one of their arguments, the extracted relations are more likely to be relevant for the respective instances and classes, than relations extracted from arbitrary document sentences (Fader et al., 2011).

2 Noncontiguous Attributes

Intuitions: Users tend to formulate their Web search queries based on knowledge that they already possess at the time of the search (Paşca, 2007). Therefore, search queries play two roles simultaneously: in addition to requesting new information, they indirectly convey knowledge in the process. In particular, attributes correspond to quantifiable properties of instances and their classes. The extraction of attributes from queries starts from the intuition that, if an attribute A is relevant for a class C, then users are likely to ask for the value of the attribute A, for various instances I of the class C. If *nutritional value* and *diameter* are relevant attributes of the classes *Food ingredients* and *Astronomical objects* respectively, it is likely that users submit queries to inquire about the values of the attributes for instances of the two classes. Such queries could take the form *"what is the (nutritional value)$_A$ of (olive oil)$_I$"* and *"what is the (diameter)$_A$ of (jupiter)$_I$"*; or the more compact *"(nutritional value)$_A$ of (olive oil)$_I$"* and *"(diameter)$_A$ of (jupiter)$_I$"*. In this case, the attributes are relatively short phrases (*nutritional value*, *diameter*), and are expected to appear as contiguous phrases within queries. Previous methods on attribute extraction from queries specifically target this type of attributes. In fact, some methods apply dedicated extraction patterns (e.g., A *of* I) over either queries (Paşca et al., 2007) or documents (Tokunaga et al., 2005). Other methods expand manually-provided seed sets of attributes, with other phrases that co-occur with instances within queries, in similar contexts as the seed attributes do (Paşca, 2007).

While simpler properties are often mentioned in queries as short, contiguous phrases, finer-grained properties often are not. Queries seeking the *reason for solidification* for some *Food ingredients* could, but rarely do, contain the attribute verbatim (*"what is the reason for the solidification of honey"*). Instead, queries are more likely to inquire about the expected value, while specifying the instance and the properties encoded by the attribute (*"(why)$_A$ does (honey)$_I$ (solidify)$_A$"*). Readable descriptions (names) of the attributes can be recovered from the queries, by assembling the type of the expected value and the properties together (*reason (it) solidifies*). Thus, fact and explanation-seeking queries are an intriguing source of noncontiguous attributes that are not restricted to short phrases, and are not required to occur as contiguous phrases in queries.

Acquisition from Queries: The extraction method proposed in this paper takes as input a set of target classes, each of which is available as a set of instances that belong to the class; and a set of anonymized queries independent from one another. As illustrated in Figure 1, the method selects queries that contain an instance of a class together with what is deemed to be likely a noncontiguous attribute, and outputs ranked lists of attributes for each class. The extraction consists in several stages:

- selection of a subset of queries that contain an instance in a form that suggests the queries ask for the value of a noncontiguous attribute of the instance;
- extraction of noncontiguous attributes, from query fragments that describe the property of interest and the type of its expected value;
- aggregation and ranking of attributes of individual instances of a class, into attributes of a class.

Extraction Patterns: In order to determine whether a query contains an attribute of a class, the query is matched against the extraction patterns from Table 1. The use of patterns in attribute extraction has been previously suggested in (Paşca et al., 2007; Tokunaga et al., 2005), where the pattern *what is the* A *of* I extracts noun-phrase A attributes of instances I from queries and documents. In our case, the patterns are constructed such that they match fact-seeking and explanation-seeking questions that likely inquire about the value of a relevant property of an instance I of the class C. For example, the first pattern from Table 1 matches queries such as *"when did everquest become free to play"* and *"when was radon discovered as an element"*, which inquire about the date or time when certain events affected certain properties of the instances *everquest* and *radon* respectively. Instances I of the class C may be available as non-disambiguated items, that is, as strings (*java*) whose meaning is otherwise unknown; or as disambiguated items, that is, as strings associ-

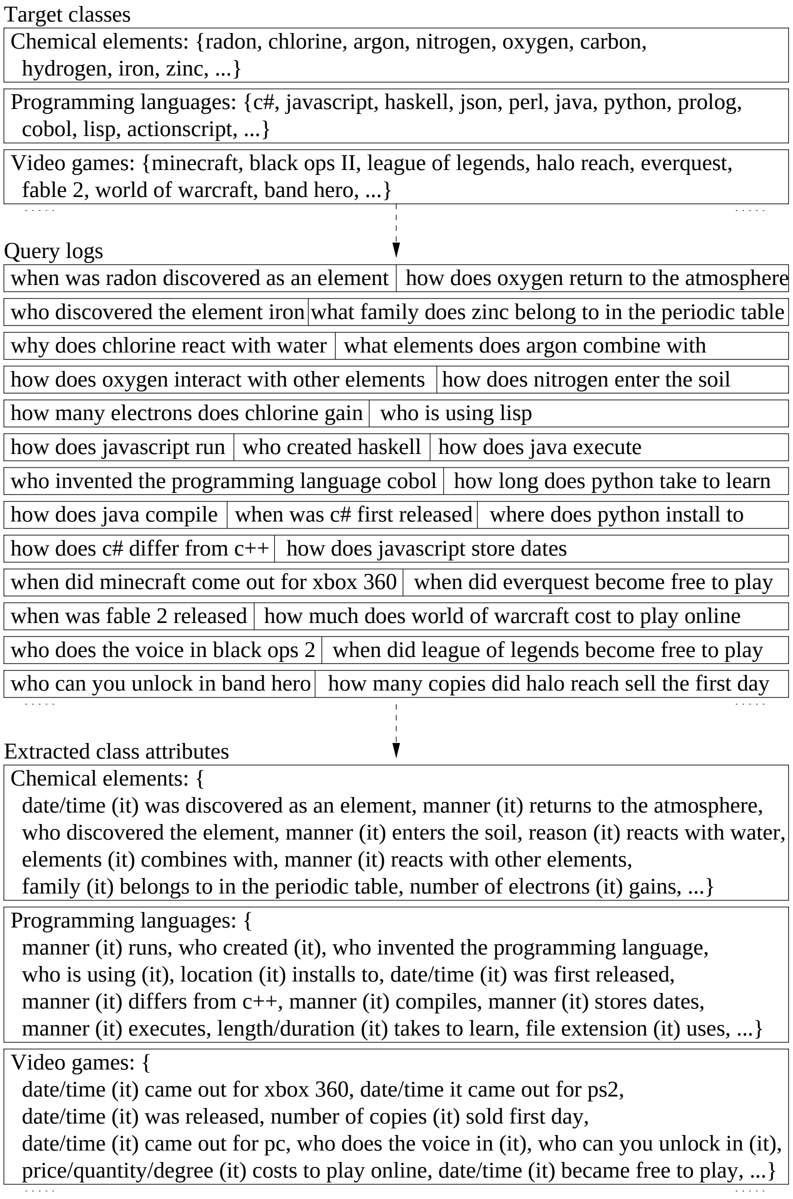
Target classes
Chemical elements: {radon, chlorine, argon, nitrogen, oxygen, carbon, hydrogen, iron, zinc, ...}
Programming languages: {c#, javascript, haskell, json, perl, java, python, prolog, cobol, lisp, actionscript, ...}
Video games: {minecraft, black ops II, league of legends, halo reach, everquest, fable 2, world of warcraft, band hero, ...}
Query logs
when was radon discovered as an element
how does oxygen return to the atmosphere
who discovered the element iron
what family does zinc belong to in the periodic table
why does chlorine react with water
what elements does argon combine with
how does oxygen interact with other elements
how does nitrogen enter the soil
how many electrons does chlorine gain
who is using lisp
how does javascript run
who created haskell
how does java execute
who invented the programming language cobol
how long does python take to learn
how does java compile
when was c# first released
where does python install to
how does c# differ from c++
how does javascript store dates
when did minecraft come out for xbox 360
when did everquest become free to play
when was fable 2 released
how much does world of warcraft cost to play online
who does the voice in black ops 2
when did league of legends become free to play
who can you unlock in band hero
how many copies did halo reach sell the first day
Extracted class attributes
Chemical elements: {
date/time (it) was discovered as an element, manner (it) returns to the atmosphere,
who discovered the element, manner (it) enters the soil, reason (it) reacts with water,
elements (it) combines with, manner (it) reacts with other elements,
family (it) belongs to in the periodic table, number of electrons (it) gains, ...}
Programming languages: {
manner (it) runs, who created (it), who invented the programming language,
who is using (it), location (it) installs to, date/time (it) was first released,
manner (it) differs from c++, manner (it) compiles, manner (it) stores dates,
manner (it) executes, length/duration (it) takes to learn, file extension (it) uses, ...}
Video games: {
date/time (it) came out for xbox 360, date/time it came out for ps2,
date/time (it) was released, number of copies (it) sold first day,
date/time (it) came out for pc, who does the voice in (it), who can you unlock in (it),
price/quantity/degree (it) costs to play online, date/time (it) became free to play, ...}

Figure 1: Overview of extraction of noncontiguous attributes from Web search queries

ated with pointers to knowledge base entries with a disambiguated meaning (*Java (programming language)*). In the first case, the matching of a query fragment, on one hand, to the portion of an extraction pattern corresponding to an instance I, on the other hand, consists in simple string matching. In the second case, the matching requires that the disambiguation of the query fragment, in the context of the query, matches the desired disambiguated meaning of I from the pattern. The subset of queries matching any of the extraction patterns, for any instances I of a class C, are the queries that contribute to extracting noncontiguous attributes of the class C.

Collecting Attributes of Individual Instances: A small set of rules optionally converts wh-prefixes into coarse-grained types of the expected values (e.g., *how long* into *length/duration*; or *when* into *date/time*). In the case of *what*-prefixed queries, the adjacent noun phrase, if any, is considered to be the expected type (*"what nutritional value .."* into *nutritional value*). Similar rules have been employed for shallow analysis of open-domain questions (Dumais et al., 2002). The predicate verbs in the remainder of the query are updated, to match the tense specified by the auxiliary verb (e.g., *"when did .."*), if any, following the wh-prefix. Thus, the verb *come* is converted to the past tense *came*, in the case of the query *"when did minecraft come out for xbox 360"*. An

Extraction Pattern → Examples of Matched Queries
when [does|did|do|was|were] [a|an|the|<nothing>] I A → when did everquest become free to play
why [does|did|do|was|were] [a|an|the|<nothing>] I A → why does chlorine interact with water
where [does|did|do|was|were] [a|an|the|<nothing>] I A → where does radon occur naturally
how [does|did|do|was|were] [a|an|the|<nothing>] I A → how does nitrogen enter the soil
who [does|did|do|was|were] [a|an|the|<nothing>] I A → who did claude monet study under
how A [does|did|do|was|were] [a|an|the|<nothing>] I A → how fast does oxygen dissolve in water
who A I → who invented the programming language cobol (Note: A does not start with [is|are|was|were])
what A [does|did|do|was|were] [a|an|the|<nothing>] I A → what elements does argon combine with
which A [does|did|do|was|were] [a|an|the|<nothing>] I A → which ports does minecraft use

Table 1: The extraction patterns match queries that are likely to inquire about the value of a noncontiguous attribute of an instance (I=a required instance; A=a required non-empty sequence of arbitrary tokens)

attribute is constructed from the concatenation of the wh-prefix or expected type (*date/time*); the slot pronoun *it*, in lieu of the instance (*date/time (it)*); and the query remainder after tense conversion (*date/time (it) came out for xbox 360*). If the linking verb following the wh-prefix is a form of *be* (e.g., *was*), then the linking verb is also retained after the slot pronoun, to form a more coherent attribute (*date/time (it) was first released*). Since constructed attributes are noun phrases, they are more consistent with, and can be more easily inserted among, existing attributes in structured data repositories (infobox entries of articles in Wikipedia, or property names or topics in Freebase).

Aggregation into Class Attributes: Attributes of a class C are aggregated from attributes of individual instances I of the class. An attribute A is deemed more relevant for C if the attribute is extracted for more of the instances I of the class C, and for fewer instances I that do not belong to the class C. Concretely, the score of an attribute for a class is the lower bound of the Wilson score interval (Brown et al., 2001) where the number of positive observations is the number of queries for which the attribute A is extracted for some instance I in the class C, $|\{Query(I,A)\}_{I \in C}|$; and the number of negative observations is the number of queries for which the attribute A is extracted for some instances I outside of the class C, $|\{Query(I,A)\}_{I \notin C}|$. The scores are internally computed at 95% confidence. Attributes of each class are ranked in decreasing order of their scores.

Reduction of Near-Duplicate Attributes: Due to lexical variations across queries from which attributes are extracted, some of the attributes are equivalent or nearly equivalent to one another. For example, *gained independence, won its independence* and *gained its freedom* of the class *Countries* are roughly equivalent, although they employ distinct tokens. The diversity and potential usefulness of a ranked list of attributes can be increased, if groups of near-duplicate attributes are identified in the list, and merged together.

A lower-ranked attribute is marked as a near-duplicate of a higher-ranked (i.e., earlier) attribute from the list, if all tokens from the lower-ranked attribute match either tokens from the higher-ranked attribute (*gained independence* vs. *won its independence*), or tokens from synonyms of phrases from the earlier attribute (*gained independence* vs. *won its independence*; or *takes to show symptoms* vs. *takes to come out*). Stop words, which include linking verbs, pronouns, determiners, conjunctions, wh-prefixes and prepositions, are not required to match. Synonyms may be either derived from existing lexical resources (e.g., WordNet (Fellbaum, 1998)), or mined from large document collections (Madnani and Dorr, 2010). Lower-ranked near-duplicate attributes are merged with the higher-ranked ones from the ranked list, thus improving the diversity of the list.

3 Experimental Setting

Textual Data Sources: The experiments rely on a random sample of around 1 billion fully-anonymized queries in English, submitted to a general-purpose Web search engine. Each query is available independently from other queries, and is accompanied by its frequency of occurrence in the query logs.

Target Classes: Table 2 shows the set of 40 target classes for evaluating the attributes extracted from queries. In an effort to reuse experimental setup proposed in previous work, each of the 40 manually-compiled classes introduced in (Paşca, 2007) is mapped into the Wikipedia category that best matches it. For example, the evaluation classes *Aircraft Model*, *Movie*, *Religion* and *Ter-*

Class (Examples of Instances)
Actors (keanu reeves, milla jovovich, ben affleck), Aircraft (boeing 737, bombardier crj200, embraer 170), Animated characters (bugs bunny, pink panther (character), yosemite sam), Association football clubs (a.s. roma, fluminense football club, real madrid), Astronomical objects (alpha centauri, jupiter, delta corvi), Automobiles (nissan gt-r, tesla model s, toyota prius), Awards (grammy award, justin winsor prize (library), palme d'or), Battles and operations of world war ii (battle of midway, operation postmaster, battle of milne bay), Chemical elements (plutonium, radon, hydrogen), Cities (rio de janeiro, osaka, chiang mai), Companies (best buy, aveeno, pepsico), Countries (costa rica, rwanda, south korea), Currencies by country (japanese yen, swiss franc, korean won), Digital cameras (canon eos 400d, nikon d3000, pentax k10d), Diseases and disorders (anorexia nervosa, hyperlysinemia, repetitive strain injury), Drugs (fluticasone propionate, phentermine, tramadol), Empires (ottoman empire, roman empire, mughal empire), Films (the fifth element, mockingbird don't sing, ten thousand years older), Flowers (trachelospermum jasminoides, lavandula stoechas, evergreen rose), Food ingredients (carrot, olive oil, fennel), Holidays (good friday, easter, halloween), Hurricanes in North America (hurricane katrina, hurricane wilma, hurricane dennis), Internet search engines (google, baidu, lycos), Mobile phones (nokia n900, htc desire, samsung s5560), Mountains (mount rainier, cerro san luis obispo, steel peak), National Basketball Association teams (los angeles lakers, cleveland cavaliers, indiana pacers), National parks (yosemite national park, orang national park, tortuguero national park), Newspapers (the economist, corriere del trentino, seattle medium), Organizations designated as terrorist (taliban, shining path, eta), Painters (claude monet, domingo antonio velasco, tarcisio merati), Programming languages (javascript, prolog, obliq), Religious faiths traditions and movements (confucianism, fudoki, omnism), Rivers (danube, pingo river, viehmoorgraben), Skyscrapers (taipei 101, 15 penn plaza, eqt plaza), Sports events (tour de france, 1984 scottish cup final, rotlewi versus rubinstein), Stadiums (fenway park, chengdu longquanyi, stade geoffroy-guichard), Treaties (treaty of versailles, franco-indian alliance, treaty of cordoba), Universities and colleges (cornell university, nugaal university, gale college), Video games (minecraft, league of legends, everquest), Wine (madeira wine, yellow tail (wine), port wine)

Table 2: Set of 40 Wikipedia categories used as target classes in the evaluation of attributes

rroristGroup from (Paşca, 2007) are mapped into the Wikipedia categories *Aircraft*, *Films*, *Religious faiths traditions and movements* and *Organizations designated as terrorist* respectively. The name of the Wikipedia category only serves as a convenience label for its target class, and is not otherwise exploited in any way during the evaluation. Instead, a target class consists in a set of titles of Wikipedia articles, of which sample titles (e.g., the Wikipedia article titled *nissan gt-r*) are shown in lowercase for each class (e.g., *Automobiles*) in Table 2. The set of instances of a class is selected from all articles listed under the respective category in Wikipedia, or listed under sub-categories of the respective category.

Label	Examples of Attributes
vital	Astronomical objects: manner (it) generates its energy
	Food ingredients: temperature (it) solidifies
	Religion: date/time (it) became a religion
okay	Astronomical objects: manner (it) became a constellation
	Food ingredients: reason (it) sparks in the microwave
	Religion: manner (it) feels about abortion
wrong	Astronomical objects: reason (it) has arms
	Food ingredients: manner (it) cleans pennies
	Religion: who owns (it)

Table 3: Correctness labels manually assigned to attributes extracted for various classes

The target classes contain between 41 (for *National Basketball Association teams*) and 66,934 (for *Films*) instances, with an average of 10,730 instances per class.

Synonym Repository: A synonym repository extracted separately from Web documents contains mappings from each of around 60,000 phrases in English, to lists of their synonym phrases. For example, the top synonyms available for the phrases *turn off* and *contagious* are [*switch off*, *extinguish*, *turn out*, ..] and [*infectious*, *catching*, *communicable*, ..] respectively.

Parameter Settings: Queries that match any of the extraction patterns from Table 1 are syntactically parsed (Petrov et al., 2010). As a prerequisite, the portion I of the patterns from the table must match a disambiguated instance from a query.

A variation of the tagger introduced in (Cucerzan, 2007) maps query fragments to their disambiguated, corresponding Wikipedia instances (i.e., to Wikipedia articles). The tagger is simplified to select the longest instance mentions, and does not use gazetteers or queries for training. Depending on the sources of textual data available for training, any taggers (Cucerzan, 2007; Ratinov et al., 2011; Pantel et al., 2012) that disambiguate text fragments relative to Wikipedia entries can be employed.

4 Evaluation Results

Attribute Accuracy: The top 50 attributes, from the ranked lists extracted for each target class, are manually assigned correctness labels. As shown in Table 3, an attribute is marked as *vital*, if it must be present among representative attributes of the

Class	Precision of Extracted Attributes			
	%vital	%okay	%wrong	Score
Awards	29	14	7	0.72
Chemical elements	46	2	2	0.94
Companies	42	1	7	0.85
Food ingredients	31	9	10	0.71
Programming languages	31	7	12	0.69
Stadiums	42	5	3	0.89
Video games	33	14	3	0.80
...				
Avg-All-Classes	33	10	7	0.76

Table 4: Accuracy of top 50 class attributes extracted from fact-seeking and explanation-seeking queries, over the evaluation set of 40 target classes

class; *okay*, if it provides useful but non-essential information; and *wrong*, if it is incorrect (Paşca, 2007). For example, the attributes *manner (it) generates its energy*, *manner (it) became a constellation* and *reason (it) has arms* are annotated as *vital*, *okay* and *wrong* respectively for the class *Astronomical objects*. To compute the precision score over a set of attributes, the correctness labels are converted to numeric values: *vital* to 1.0, *okay* to 0.5, and *wrong* to 0.0. Precision is the sum of the correctness values of the attributes, divided by the number of attributes.

Table 4 summarizes the precision scores over the evaluation set of target classes. The scores vary from one class to another, for example 0.71 for *Food ingredients* but 0.94 for *Chemical elements*. The average score is 0.76, indicating that attributes extracted from fact and explanation-seeking queries have encouraging levels of accuracy. The results already take into account the detection of near-duplicate attributes. More precisely, the highest-ranked attribute in each group of near-duplicate attributes, examples of which are shown in Table 5, is retained and evaluated; the lower-ranked attributes from each group are not considered in the evaluation. Attributes like *number of passengers (it) can hold*, *number of passengers it fits* and *number of passengers it seats* are nearly equivalent, but are still not marked as near-duplicates for the class *Aircraft*, when they should. Conversely, the attribute *location (it) lives* is marked as a near-duplicate of *location (it) lives in new york*, when it should not. Nevertheless, a significant number of near-duplicates, which would otherwise crowd the ranked lists of attributes with redundant information, are identified and discarded.

Target Class: Group of Near-Duplicate Attributes
Actors: movies (it) plays in, played in, acts in, acted in, played, played on
Automobiles: date (it) was first manufactured, first produced, first made
Battles and operations of World War II: reason (it) happened, took place, occurred
Chemical elements: manner (it) returns to the atmosphere, gets back into the atmosphere, got into the atmosphere, gets into the atmosphere, enters the environment, enters the atmosphere
Companies: location (it) makes its products, manufactures its products, produces its products, gets its products, makes its products, manufactures their products
Companies: date/time (it) began outsourcing, started outsourcing, outsourced
Countries: date (it) got its independence, gained independence, gained its independence, got independence, got their independence, won its independence, achieved independence, received its independence, gained its freedom
Diseases and disorders: length/duration (it) takes to show symptoms, takes to show up, takes to show, takes to appear, takes to manifest, takes to come out

Table 5: Groups of near-duplicate attributes identified for various classes. Attributes within a group are ranked according to their individual scores. Removing all but the first attribute of each group, from the ranked list of attributes of the respective class, improves the diversity of the list

Discussion: The set of patterns shown in Table 1 is extensible. Moreover, the patterns are subject to errors. They may cause false matches, resulting in erroneous extractions. The extent to which this occurs is indirectly measured in the overall precision results. The modification of some of the patterns, or the addition of new ones, would likely affect the expected coverage and precision of the extracted attributes. If a pattern is particularly noisy, it is likely to cause systematic errors, and therefore produce attributes of lower quality.

Since attributes in Wikipedia and Freebase are initially entered manually by human editors, their correctness is virtually guaranteed. As for attributes extracted automatically, previous comparisons indicate that attributes tend to have higher quality when extracted from queries instead of documents (Paşca, 2007). Indeed, a set of extraction patterns applied to text produces attributes whose average precision at rank 50 is 0.44 when extracted from documents, vs. 0.63 from queries (Paşca et al., 2007). More importantly, previously available or extracted attributes are virtually always simple, short noun phrases like *nutritional value*, *taste* or *solubility in water*. Even if not confined to noun phrases, they are still short,

Run: [Ranked Attributes for a Sample of Classes]
Class: Automobiles:
D: [(it) goes on sale, (it) will go on sale, (it) is an engineering playground, (it) will be available in japan, (it) shows up in japan, (it) is a technical tour de force, (it) unveiled at tas 2008, (it) runs a 7:38, (it) is a unique car, (it) uses a premium midship package, (it) features an all-new 3.8-litre, (it) is one of the fastest cars, (it) made a quick drive-by, ..]
Q: [price/quantity/degree (it) weights, year (it) was banned from bathurst, manner (it) launch control works, engine (it) has, kind of engine (it) has, price/quantity/degree (it) costs in japan, number of horsepower (it) has, price/quantity/degree horsepower (it) has, number of seats (it) has, speed (it) goes, who designed (it), ..]
Class: Mobile phones:
D: [(it) was announced on september 17 2008, (it) ceased with version, (it) was scheduled to be released in late 2010, (it) also supports qt (toolkit), (it) supports hardware capable, (it) can synchronize with microsoft outlook, (it) also supports python (programming language), ..]
Q: [date/time (it) came out in australia, who carries (it), reason (it) keeps rebooting, colours (it) comes in, video format (it) supports, date/time (it) was released, date/time (it) came out in the uk, length/duration (it)'s battery lasts, who sells (it), how much (it) costs, ..]
Class: Mountains:
D: [(it) is an active volcano, (it) is in the distance, (it) is the highest peak in cascade range, (it) is 14,410 feet, (it) was established in 1899, (it) comes into view, (it) was established as a national park, ..]
Q: [date/time (it) last erupted, manner (it) erupted in 1882, manner (it) formed, date/time (it) first became active, manner (it) got its name, number of eruptions (it) had, type of magma (it) has, reason (it) became a national park, kind of animals (it) has, ..]

Table 6: Top relations extracted for a sample of target classes via open-domain relations from documents (D) or via attributes from queries (Q)

like *vegan*, *healthy* or *gluten free* (Van Durme et al., 2008; Paşca, 2012). In comparison, attributes extracted in this paper accommodate properties that are sometimes awkward or even impossible to express through short phrases.

Noncontiguous Attributes as Relations: Noncontiguous attributes extracted from fact-seeking queries are embodiments of relations linking the instances mentioned in the queries, on one hand, and the values being requested by the queries, on the other hand. Therefore, the method proposed in this paper can also be regarded as a method for the acquisition of relevant relations of various classes. The extracted relations specify the left argument (i.e., the instance) and the linking relation name (i.e., the attribute). They only specify the type of the, but not the actual, right argument (i.e., the value being requested).

An additional experiment compares the accuracy of relations extracted as noncontiguous attributes from queries, vs. relations extracted by a previous open-domain method (Fader et al., 2011) from 500 million Web documents. The previous method, including its extraction patterns and its ranking scheme, is designed with instances rather than classes in mind. For fairness to the method in (Fader et al., 2011), the evaluation procedure is slightly adjusted. The set of instances associated with each target class, over which the two methods are evaluated, is reduced to a single representative instance selected a-priori. The instances are shown as the first instances in parentheses for each class in the earlier Table 2. Thus, the class attributes are extracted using only the instances *keanu reeves*, *boeing 737* and *bugs bunny* in the case of the classes *Actors*, *Aircraft* and *Animated characters* respectively.

Table 6 suggests that noncontiguous attributes extracted from queries tend to capture higher-quality relations than arbitrary relations extracted from documents. Because fact-seeking queries inquire about the value of some relations (attributes) of an instance, the relations themselves tends to be more relevant than relations extracted from arbitrary document sentences. Nevertheless, relations derived from queries likely serve as a useful complement, rather than replacement, of relations from documents. The former only discover what relations may be relevant; the latter also identify their occurrences within text.

5 Related Work

Sources of text from which relations (Zhu et al., 2009; Carlson et al., 2010; Lao et al., 2011) and, more specifically, attributes can be extracted include Web documents and data in human-compiled encyclopedia. In Web documents, attributes are available within unstructured (Tokunaga et al., 2005; Paşca et al., 2007), structured (Raju et al., 2008) and semi-structured text (Yoshinaga and Torisawa, 2007), layout formatting tags (Wong et al., 2008), itemized lists or tables (Cafarella et al., 2008). In human-compiled encyclopedia (Wu and Weld, 2010), data relevant to attribute extraction includes infoboxes and category labels (Nastase and Strube, 2008; Hoffart et al., 2013) associated with Wikipedia articles. In order to acquire class attributes, a common strategy is to first acquire attributes of instances, then aggregate or propagate (Talukdar and Pereira,

2010) attributes, from instances to the classes to which the instances belong. The role of Web search queries, as an alternative textual data source to Web documents in open-domain information extraction, has been investigated in the tasks of attribute extraction (Paşca, 2007; Paşca, 2012), as well as in collecting sets of related instances (Jain and Pennacchiotti, 2010).

To increase diversity within a ranked list of attributes, the extraction method in this paper employs a synonym vocabulary to approximately identify groups of near-duplicate attributes. As reported for previous methods, the resulting lists may still contain lexically different but semantically equivalent attributes. Scenarios where detecting all equivalent attributes is important may benefit from other techniques for paraphrase acquisition (Madnani and Dorr, 2010).

Sophisticated techniques are sometimes employed to identify the type of the expected answers of open-domain questions (Pinchak et al., 2009). In comparison, the loose typing of the values of our noncontiguous attributes is mostly coarse-grained. It relies on wh-prefixes (*when*, *how long*, *where*, *how*) and possibly subsequent words (*what nutritional value*) from the queries, to determine whether the values are expected to be a *date/time*, *length/duration*, *location*, *manner*, *nutritional value* etc.

Relations extracted from document sentences (e.g., *"Claude Monet was born in Paris"*) are tuples of an instance (*claude monet*), a text fragment acting as the lexicalized relation (*was born in*), and another instance (*paris*) (cf. (Fader et al., 2011; Mausam et al., 2012)). For convenience, the relation and second instance may be concatenated, as in *was born in paris* for *claude monet*. But document sentences mentioning an instance do not necessarily refer to properties of the instance that people other than the author of the document are likely to inquire about. Consequently, even top-ranked extracted relations occasionally include less informative ones, such as *comes into view* for *mount rainier*, *is on the table* for *madeira wine*, or *allows for features* for *javascript* (Fader et al., 2011). Comparatively, relations extracted via noncontiguous attributes from queries tend to refer to properties that have values that Web users inquire about in their search queries. Therefore, the relations extracted from queries are more likely to refer to salient properties, such as *date/time (it) had its last eruption* for *mount rainier*; *length/duration (it) lasts* for *madeira wine*; and *manner (it) stores date information* for *javascript*.

6 Conclusion

By requesting values for attributes of individual instances, fact-seeking and explanation-seeking queries implicitly assert the relevance of the properties encoded by the attributes, for the respective instances and their classes. The extracted attributes are not required to take the form of contiguous short phrases in the source queries, thus allowing for the acquisition of a broader range of attributes than those extracted by previous methods. Furthermore, since Web users are interested in their values, the relations to which the extracted attributes refer tend to be more relevant than relations extracted from arbitrary documents using previous methods. Current work explores the role of distributional similarities in expanding extracted attributes for narrow classes; and the extraction of noncontiguous attributes and relations from natural-language queries without a wh-prefix (e.g., *cars driven by james bond*).

Acknowledgments

The author would like to thank Efrat Farkash and Michael Kleyman for assistance with the synonym repository.

References

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 2008. 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.

L. Brown, T. Cai, and A. DasGupta. 2001. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–117.

M. Cafarella, A. Halevy, D. Wang, E. Wu, and Y. Zhang. 2008. 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.

A. Carlson, J. Betteridge, R. Wang, E. Hruschka, and T. Mitchell. 2010. 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.

S. Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, pages 708–716, Prague, Czech Republic.

S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. 2002. Web question answering: Is more always better? In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR-02)*, pages 207–214, Tampere, Finland.

A. Fader, S. Soderland, and O. Etzioni. 2011. 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.

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

J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. 2013. YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence Journal. Special Issue on Artificial Intelligence, Wikipedia and Semi-Structured Resources*, 194:28–61.

A. Jain and M. Pennacchiotti. 2010. 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.

N. Lao, T. Mitchell, and W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, pages 529–539, Edinburgh, Scotland.

N. Madnani and B. Dorr. 2010. Generating phrasal and sentential paraphrases: a survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-12)*, pages 523–534, Jeju Island, Korea.

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

M. Paşca, B. Van Durme, and N. Garera. 2007. The role of documents vs. queries in extracting class attributes from text. In *Proceedings of the 16th International Conference on Information and Knowledge Management (CIKM-07)*, pages 485–494, Lisbon, Portugal.

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

M. Paşca. 2012. Attribute extraction from conjectural queries. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING-12)*, Mumbai, India.

P. Pantel, T. Lin, and M. Gamon. 2012. Mining entity types from query logs via user intent modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-12)*, pages 563–571, Jeju Island, Korea.

S. Petrov, P. Chang, M. Ringgaard, and H. Alshawi. 2010. 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.

C. Pinchak, D. Lin, and D. Rafiei. 2009. 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.

S. Raju, P. Pingali, and V. Varma. 2008. An unsupervised approach to product attribute extraction. In *Proceedings of the 31st International Conference on Research and Development in Information Retrieval (SIGIR-08)*, pages 35–42, Singapore.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-11)*, pages 1375–1384, Portland, Oregon.

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

P. Talukdar and F. Pereira. 2010. 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.

K. Tokunaga, J. Kazama, and K. Torisawa. 2005. Automatic discovery of attribute words from Web documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 106–118, Jeju Island, Korea.

B. Van Durme, T. Qian, and L. Schubert. 2008. Class-driven attribute extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, pages 921–928, Manchester, United Kingdom.

T. Wong, W. Lam, and T. Wong. 2008. An unsupervised framework for extracting and normalizing product attributes from multiple Web sites. In *Proceedings of the 31st International Conference on Research and Development in Information Retrieval (SIGIR-08)*, pages 35–42, Singapore.

F. Wu and D. Weld. 2010. 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.

N. Yoshinaga and K. Torisawa. 2007. Open-domain attribute-value acquisition from semi-structured texts. In *Proceedings of the 6th International Semantic Web Conference (ISWC-07), Workshop on Text to Knowledge: The Lexicon/Ontology Interface (OntoLex-2007)*, pages 55–66, Busan, South Korea.

J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. Wen. 2009. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th World Wide Web Conference (WWW-09)*, pages 101–110, Madrid, Spain.