Queries as a Source of Lexicalized Commonsense Knowledge

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

Abstract

The role of Web search queries has been demonstrated in the extraction of attributes of instances and classes, or of sets of related instances and their class labels. This paper explores the acquisition of open-domain commonsense knowledge, usually available as factual knowledge, from Web search queries. Similarly to previous work in open-domain information extraction, knowledge extracted from text - in this case, from queries - takes the form of lexicalized assertions associated with open-domain classes. Experimental results indicate that facts extracted from queries complement, and have competitive accuracy levels relative to, facts extracted from Web documents by previous methods.

1 Introduction

Motivation: Open-domain information extraction methods (Etzioni et al., 2005; Pennacchiotti and Pantel, 2009; Wang and Cohen, 2009; Kozareva and Hovy, 2010; Wu et al., 2012) aim at distilling text into knowledge assertions about classes, instances and relations among them (Etzioni et al., 2011). Ideally, the assertions would complement or expand upon knowledge available in popular, human-created resources such as Wikipedia (Remy, 2002) and Freebase (Bollacker et al., 2008), reducing costs and scalability issues associated with manual editing, curation and maintenance of knowledge.

Candidate knowledge assertions extracted from text for various instances and classes (Banko et al., 2007; Cafarella et al., 2008; Wu and Weld, 2010) must satisfy several constraints in order to be useful. First, their boundaries must be correctly identified within the larger context (e.g., a document sentence) from which they are extracted. In practice, this is a challenge with arbitrary Web documents, where even instances and class labels that are complex nouns, and thus still shorter than candidate assertions, are difficult to precisely detect and pick out from surrounding text (Downey et al., 2007). This causes the extraction of assertions like *companies* may *"be in the process"*, *hurricanes* may *"run from june"*, or *video games* may *"make people"* (Fader et al., 2011). Second, the assertions must be correctly associated with their corresponding instance or class. In practice, tagging and parsing errors over documents of arbitrary quality may cause the extracted assertions to be associated with the wrong instances or classes. Examples are *video games* may *"watch movies"*, or *video games* may *"read a book"*. Third, the assertions, even if true, must refer to relevant properties or facts, rather than to statements of little or no practical interest to anyone. In practice, relevant properties may be difficult to distinguish from uninteresting statements in Web documents. Consequently, assertions extracted from Web documents include the facts that *companies* may *"say in a statement"*, or that *hurricanes* may *"be just around the corner"* or may *"be in effect"*.

Contributions: This paper explores the use of Web search queries, as opposed to Web documents, as a textual source from which knowledge pertaining to open-domain classes can be extracted. Previous explorations of the role of queries in information extraction include the acquisition of attributes of instances (Alfonseca et al., 2010) and of classes (Van Durme and Paşca, 2008); the acquisition of sets of related

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1081–1091,
October 25-29, 2014, Doha, Qatar.

instances (Sekine and Suzuki, 2007; Jain and Pennacchiotti, 2010) and their class labels (Van Durme and Paşca, 2008; Pantel et al., 2012); the disambiguation of instances mentioned in queries relative to entries in external knowledge repositories (Pantel and Fuxman, 2011) and its application in query expansion (Dalton et al., 2014); and the extraction of the most salient of the instances mentioned in a given Web document (Gamon et al., 2013). In comparison, this paper shows that queries also lend themselves to the acquisition of factual knowledge beyond attributes, like the facts that *companies* may *"buy back stock"*, *hurricanes* may *"need warm water"*, and *video games* may *"come out on tuesdays"*.

To extract knowledge assertions for diverse classes of interest to Web users, the method applies simple extraction patterns to queries. The presence of the source queries, from which the assertions are extracted, is in itself deemed evidence that the Web users who submitted the queries find the assertions to be relevant and not just random statements. Experimental results indicate that knowledge assertions extracted from queries complement, and have competitive accuracy levels relative to, knowledge extracted from Web documents by previous methods.

2 Extraction from Queries

Queries as Knowledge: 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.

A fact corresponds to a property that, together with other properties, help define the semantics of the class and its interaction with other classes. The extraction of factual knowledge from queries starts from the intuition that, if a fact F is relevant for a class C, then users are likely to ask for various aspects of the fact F, in the context of the class C. If *companies* may *"pay dividends"* or *"get audited"*, and such properties are relatively prominent for *companies*, then users eventually submit queries to inquire about the facts.

Often, queries will be simple concatenations of keywords: *"companies pay dividends"* or perhaps *"company dividends"*, *"audit companies"*. Since there are no restrictions on the linguistic structure

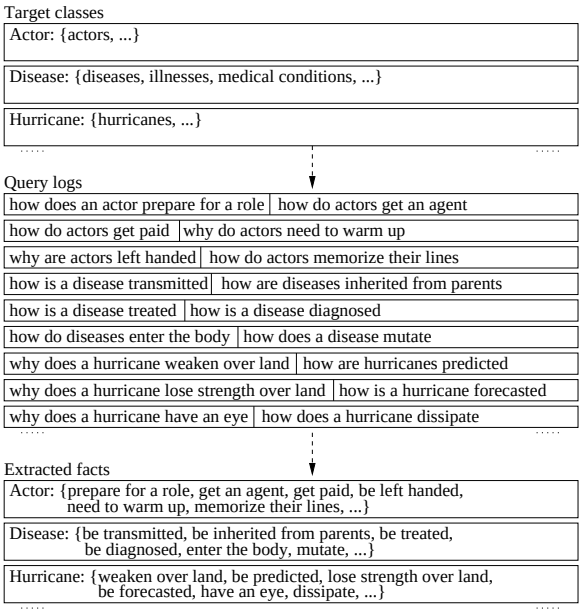

Figure 1: Overview of extraction of knowledge from Web search queries

of keyword-based queries, extracting facts from such queries would be difficult. But if queries are restricted to fact-seeking questions, the expected format of the questions makes it easier to identify the likely boundaries of the class and the fact mentioned in the queries. Queries such as *"why does a (company)$_C$ (pay dividends)$_F$"* and *"how do (companies)$_C$ (get audited)$_F$"*, follow the linguistic structure, even if minimal, imposed by formulating the query as a question. This allows one to approximate the location of the class C, possibly towards the beginning of the query; the start of the fact F, possibly as the verb immediately following the class; and the end of the fact, which possibly coincides with the end of the query.

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 class descriptors, i.e., phrases that describe the class. It also has access to a set of anonymized queries. As illustrated in Figure 1, the method selects queries that contain a class descriptor and what is deemed to be likely a fact. It outputs ranked lists of facts for each class. The extraction consists in several stages: 1) the selection of a subset of queries that refer to a class in a form that suggests the queries inquire about a fact of the class; 2) the extraction of facts, from query fragments that describe the property of interest to users submitting the queries; and 3) the aggregation and

ranking of facts of a class.

Extraction Patterns: In order to determine whether a query contains a fact for a class, the query is matched against the extraction patterns from Table 1.

The use of targeted patterns in relation extraction has been suggested before (Hearst, 1992; Fader et al., 2011; Mesquita et al., 2013). Specifically, in (Tokunaga et al., 2005), the patterns *"A of D"* or *"what is the A of D"* extract noun-phrase A attributes from queries and documents, for phrase descriptors D of the class. In our case, the patterns are constructed such that they match questions that likely inquire about the reason why, or manner in which, a relevant fact F may hold for a class C. For example, the first pattern from Table 1 matches the queries *"why does a company pay dividends"* and *"why do video games come out on tuesdays"*. These queries seek explanations for why certain properties may hold for *companies* and *video games* respectively.

A class C can be mentioned in queries through lexicalized, phrase descriptors D that capture its meaning. The descriptors D of the class C may be available as non-disambiguated items, i.e., as strings (*companies*, *firms*, *businesses*, *video games*); or as disambiguated items, that is, as pointers to knowledge base entries with a disambiguated meaning (*Company*, *Video Game*). In the first case, the matching of a query fragment, on one hand, to the portion of an extraction pattern corresponding to the class C, on the other hand, consists in simple string matching with one of the descriptors D specified for C. 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 C from the pattern. The subset of queries matching any of the extraction patterns, for any descriptor D of a class C, are the queries that contribute to extracting facts of the class C.

If a pattern from Table 1 employs a form of the auxiliary verb *"be"*, the extracted facts are modified by having the verb *"be"* inserted at their beginning. For example, the fact *"be stored sideways"* is extracted from the query *"why is wine stored sideways"*. In all patterns, the candidate fact is required to start with a verb that acts as the predicate of the query.

Ranking of Facts: Facts of a class C are aggregated from facts of individual class descriptors D.

Extraction Pattern → Examples of Matched Queries
why [does|did|do] [a|an|the|<nothing>] D F → why does a (company)$_D$ (pay dividends)$_F$ → why do (planes)$_D$ (take longer to fly west than east)$_F$ → why do (video games)$_D$ (come out on tuesdays)$_F$
why [is|was|were] [a|an|the|<nothing>] D F → why are (cars)$_D$ (made of steel)$_F$ → why is a (newspaper)$_D$ (written in columns)$_F$ → why is (wine)$_D$ (stored sideways)$_F$
how [does|did|do] [a|an|the|<nothing>] D F → how does a (company)$_D$ (use financial statements)$_F$ → how does (food)$_D$ (get absorbed)$_F$ → how do (stadiums)$_D$ (get cleaned)$_F$
how [is|was|were] [a|an|the|<nothing>] D F → how are (hurricanes)$_D$ (predicted)$_F$ → how is a (treaty)$_D$ (ratified)$_F$ → how is a (cell phone)$_D$ (unlocked)$_F$

Table 1: The extraction patterns match queries likely to inquire about facts of a class (D=a phrase acting as a class descriptor; F=a sequence of tokens whose first token is the head verb of the query)

A fact F is deemed more relevant for C if the fact is extracted for more of the descriptors D of the class C, and for fewer descriptors D that do not belong to the class C. Concretely, the score of a fact for a class is the lower bound of the Wilson score interval (Brown et al., 2001):

$$Score(F, C) = LowBound(Wilson(N_+, N_-))$$

where:

- the number of positive observations N_+ is the number of queries for which the fact A is extracted for some descriptor D of the class C, $|\{Query(D, A)\}_{D \in C}|$; and
- the number of negative observations N_- is the number of queries for which the fact F is extracted for some descriptors D outside of the class C, $|\{Query(D, A)\}_{D \notin C}|$.

The scores are internally computed at 95% confidence. Facts of each class are ranked in decreasing order of their scores. In case of ties, facts are ranked in decreasing order of the frequency sum of the source queries from which the facts are extracted.

3 Experimental Setting

Textual Data Sources: The experiments rely on a random sample of around 1 billion fully-anonymized Web search queries in English. The sample is drawn from queries 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

Target Class (class descriptors to be looked up in queries)	
Actor (actors)	Mountain (mountains)
Aircraft (planes)	Movie (movies)
Award (awards)	NationalPark (national parks)
Battle (battles)	NbaTeam (nba teams)
Car (cars)	Newspaper (newspapers)
CartoonChar (cartoon characters)	Painter (painters)
CellPhone (cell phones)	ProgLanguage (programming languages)
ChemicalElem (elements)	Religion (religions)
City (cities)	River (rivers)
Company (companies)	SearchEngine (search engines)
Country (countries)	SkyBody (celestial bodies)
Currency (currencies)	Skyscraper (skyscrapers)
DigitalCamera (digital cameras)	SoccerClub (soccer teams)
Disease (diseases)	SportEvent (sport events)
Drug (drugs)	Stadium (stadiums)
Empire (empires)	TerroristGroup (terrorist groups)
Flower (flowers)	Treaty (treaties)
Food (foods)	University (universities)
Holiday (holidays)	VideoGame (video games)
Hurricane (hurricanes)	Wine (wines)

Table 2: Set of 40 target classes used in the evaluation of extracted facts

the query logs.

Target Classes: Table 2 shows the set of 40 target classes for evaluating the extracted facts. Similar evaluation strategies were followed in previous work (Paşca, 2007). As illustrated earlier in Figure 1, a target class consists in a small set of phrase descriptors. The phrase descriptors are selected such that they best approximate the meaning of the class. In general, the descriptors can be selected and expanded with any strategy from any source. One such possible source might be synonym sets from WordNet (Fellbaum, 1998). Following a stricter strategy, the sets of descriptors in our experiments contain only one phrase each, manually selected to match the target class. Examples are the sets of phrase descriptors {*actors*} for the class *Actor* and {*nba teams*} for *NbaTeam*. The occurrence of a descriptor (*nba teams*) in a query (*"how do nba teams make money"*) is deemed equivalent to a mention of the corresponding class (*NbaTeam*) in that query. Each set of descriptors of a class is then expanded (not shown in Table 2), to also include the singular forms of the descriptors (e.g., *nba team* for *nba teams*). Further inclusion of additional descriptors would increase the coverage of the extracted facts.

Experimental Runs: The baseline run R_D is the extraction method introduced in (Fader et al., 2011). The method produces triples of an instance or a class, a text fragment capturing a fact, and another instance or class. In these experiments, the second and third elements of each triple are concatenated together, giving pairs of an instance or a class, and a fact applying to it. The baseline run is applied to around 500 million Web documents in English. [1] In addition to the baseline run, the method introduced in this paper constitutes the second experimental run R_Q. Facts extracted by the two experimental runs are directly comparable: both are text snippets extracted from the respective sources of text - documents in the case of R_D, or queries in the case of R_Q.

Parameter Settings: Queries that match any of the extraction patterns from Table 1 are syntactically parsed (Petrov et al., 2010), in order to verify that the first token of an extracted fact is the head verb of the query. Extracted facts that do not satisfy the constraint are discarded. A positive side effect of doing so is to avoid extraction from some of the particularly subjective queries. For example, facts extracted from the queries *"why is (A) evil"* or *"why is (B) ugly"*, where *(A)* and *(B)* are the name of a company and actress respectively, are discarded.

4 Evaluation Results

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

Following evaluation methodology from prior work (Paşca, 2007), the top 50 facts, from a ranked lists extracted for each target class, are manually assigned correctness labels. A fact is marked as *vital*, if it must be present among representative facts of the class; *okay*, if it provides useful but non-essential information; and *wrong*, if it is incorrect (Paşca, 2007). For example, the facts *"run on kerosene"*, *"be delayed"* and *"fly wiki"* are annotated as *vital*, *okay* and *wrong* respectively for the class *Aircraft*. To compute the precision score

[1] At the time when the experiments were conducted, the facts were extracted by the baseline run from English documents in the ClueWeb collection, and were accessible at http://reverb.cs.washington.edu.

Target Class: Sample of Extracted Facts (with Source Queries)	Target Class: Sample of Extracted Facts (with Source Queries)
Actor (may): prepare for a role (how does an actor prepare for a role), get an agent (how do actors get an agent), do love scenes (how do actors do love scenes), get paid (how do actors get paid), be left handed (why are actors left handed), need to warm up (why do actors need to warm up)	Car (may): backfire (why does a car backfire), burn oil (why do cars burn oil), pull to the right (why do cars pull to the right), pull to the left (why does a car pull to the left), catch on fire (how does a car catch on fire), run hot (why do cars run hot), get repossessed (why do cars get repossessed)
Company (may): buy back stock (how does a company buy back stock), go public (why does a company go public), buy back shares (why do companies buy back shares), incorporate in delaware (why do companies incorporate in delaware), pay dividends (why does a company pay dividends), merge (how do companies merge)	Disease (may): be transmitted (how is a disease transmitted), be inherited from parents (how are diseases inherited from parents), affect natural selection (how do diseases affect natural selection), be treated (how is a disease treated), affect the conquest of the americas (how did diseases affect the conquest of the americas), be diagnosed (how is a disease diagnosed)
Hurricane (may): weaken over land (why does a hurricane weaken over land), be predicted (how are hurricanes predicted), lose strength over land (why does a hurricane lose strength over land), have an eye (why does a hurricane have an eye), be forecasted (how is a hurricane forecasted), dissipate (how does a hurricane dissipate), lose strength (how do hurricanes lose strength)	NbaTeam (may): make money (how does an nba team make money), communicate to win (how does an nba team communicate to win), want expiring contracts (why do nba teams want expiring contracts), make the playoffs (how do nba teams make the playoffs), get their names (how do nba teams get their names), do sign and trades (why do nba teams do sign and trades), lose money (how do nba teams lose money)

Table 3: Examples of facts extracted for various classes by run R_Q

Class	Precision						Class	Precision					
	@10		@20		@50			@10		@20		@50	
	R_D	R_Q	R_D	R_Q	R_D	R_Q		R_D	R_Q	R_D	R_Q	R_D	R_Q
Actor	0.60	0.85	0.57	0.85	0.60	0.83	Mountain	0.20	0.75	0.10	0.72	0.05	0.55
Aircraft	0.50	0.95	0.42	0.87	0.47	0.81	Movie	0.40	0.20	0.37	0.20	0.40	0.32
Award	0.50	0.25	0.45	0.25	0.52	0.23	NationalPark	0.40	0.70	0.32	0.72	0.30	0.69
Battle	0.25	0.45	0.42	0.46	0.38	0.44	NbaTeam	0.60	0.75	0.42	0.80	0.20	0.77
Car	0.55	0.80	0.62	0.82	0.52	0.75	Newspaper	0.25	0.80	0.32	0.55	0.44	0.59
CartoonChar	0.25	0.60	0.22	0.57	0.18	0.55	Painter	0.30	0.75	0.40	0.65	0.42	0.61
CellPhone	0.75	0.90	0.75	0.82	0.55	0.82	ProgLanguage	0.20	0.75	0.25	0.72	0.25	0.70
ChemicalElem	0.45	0.90	0.45	0.72	0.54	0.72	Religion	0.10	0.80	0.30	0.70	0.13	0.69
City	0.30	0.80	0.27	0.67	0.27	0.63	River	0.65	0.95	0.70	0.87	0.54	0.57
Company	0.60	0.95	0.57	0.95	0.53	0.91	SearchEngine	0.40	0.70	0.37	0.65	0.38	0.64
Country	0.30	0.85	0.25	0.90	0.20	0.83	SkyBody	0.55	0.00	0.32	0.00	0.28	0.00
Currency	0.40	0.90	0.25	0.85	0.22	0.73	Skyscraper	0.45	0.85	0.37	0.77	0.24	0.78
DigitalCamera	0.30	0.90	0.35	0.85	0.42	0.77	SoccerClub	0.35	0.15	0.37	0.33	0.41	0.31
Disease	0.55	0.90	0.60	0.70	0.64	0.60	SportEvent	0.30	0.00	0.27	0.00	0.32	0.00
Drug	0.20	0.95	0.30	0.87	0.40	0.78	Stadium	0.50	0.85	0.50	0.77	0.47	0.75
Empire	0.15	0.45	0.12	0.52	0.23	0.49	TerroristGroup	0.90	0.55	0.70	0.55	0.55	0.53
Flower	0.60	0.90	0.50	0.80	0.48	0.78	Treaty	1.00	0.75	0.90	0.75	0.77	0.59
Food	0.65	0.80	0.55	0.85	0.43	0.85	University	0.10	0.95	0.05	0.92	0.10	0.70
Holiday	0.30	0.25	0.17	0.22	0.19	0.14	VideoGame	0.20	0.90	0.25	0.85	0.28	0.77
Hurricane	0.40	0.80	0.37	0.77	0.32	0.73	Wine	0.70	1.00	0.60	0.87	0.56	0.70
							Average-Class	0.43	0.71	0.40	0.67	0.38	0.63

Table 4: Relative accuracy of facts extracted from documents in run R_D, vs. facts extracted from queries in run R_Q

over a set of facts, 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 facts, divided by the number of facts. Table 3 shows a sample of facts extracted from queries by run R_Q, which are judged to be *vital* or *okay*.

Table 4 provides a comparison of precision at ranks 10, 20 and 50, for each of the 40 target classes and as an average over all target classes. The scores vary from one class to another and between the two runs, for example 0.22 (R_D) and 0.73 (R_Q) for the class *Currency* at rank 50, but 0.77 (R_D) and 0.59 (R_Q) for *Treaty*. Run R_Q fails to extract any facts for two of the target classes, *SkyBody* and *SportEvent*. Therefore, it receives no credit for those classes during the computation of precision.

Over all target classes, run R_Q is superior to run R_D, with relative precision boosts of 65% (0.71 vs. 0.43) at rank 10, 67% at rank 20, and 65% at rank 50. The results show that facts extracted from

Run: [Ranked Facts Extracted from Text for a Sample of Classes]
Class: Actor (may):
R_D: [do a great job, get the part, play their roles, play their parts, play their characters, be on a theatre, die aged 81, be all great, deliver their lines, portray their characters, take on a role, be best known for his role, play the role of god, be people, give great performances, bring the characters to life, wear a mask, be the one, have chemistry, turn director, read the script, ..]
R_Q: [prepare for a role, get an agent, do love scenes, get paid, be left handed, need to warm up, get started, get paid so much, memorize their lines, get ripped so fast, remember their lines, make themselves cry, learn their lines, jump out of a window in times square, lose weight so fast, play dead, be paid, kiss, remember lines, memorize lines, get discovered, get paid for movies, go uncredited, say break a leg, get their start, have perfect skin, become actors, ..]
Class: Car (may):
R_D: [get a tax write-off, can be more competitive than airline rates, be in good condition, be first for second hand cars, be in the shop, relocate to a usa firm, be in motion, come to a stop, hire companies, be in great shape, be for sale, hire service from spain, ride home, be on fire, use the autos.com, come to a halt, catch fire, be on road, be on display, go on sale, hit a tree, be available for delivery, stop in front, be a necessity, go off the road, pull out in front, hire services, run out of gas, ..]
R_Q: [backfire, burn oil, save ostriches from extinction, pull to the right, pull to the left, catch on fire, run hot, sputter, get repossessed, have a top speed, be called a car, have gears, get impounded, be called cars, go to auction, called whip, made of steel, get hot in the sun, shake at high speed, changed america, totaled, cut out, cut off while driving, fail emissions, protect from lightning, run rich, lose oil, become electrically charged, cut off, flip over, know tire pressure, have a maximum speed, require premium gas, shake at high speeds, stall out, cause acid rain, fog up, get stuck in park, need an oil change, ..]
Class: Company (may):
R_D: [say in a statement, specialize in local moves, be in the process, go out of business, have been in business, be in business, do business, file for bankruptcy, make money, be on track, say in a press release, be a place, have cut back on health insurance, state in a press release, be on the verge, save money, be in talks, have helped thousands of consumers, reduce costs, go bust, be in the midst, say in a release, be founded in 1999, be in trouble, be founded in 2000, be losing money, ..]
R_Q: [buy back stock, go public, buy back shares, incorporate in delaware, pay dividends, merge, go global, go international, use financial statements, verify education, expand internationally, go green, verify employment, need a website, choose to form as a corporation, do market research, go private, diversify, go into administration, get on angies list, pay dividend, struck off, buy back their shares, get audited, need a mission statement, repurchase common stock, spin off, get listed on the nyse, create value, distribute dividends, need a strategic plan, ..]
Class: Mountain (may):
R_D: [spot fever, meet the sea, be covered with snow, be covered in snow, be the place, come into view, be on fire, be fun, fly fishing, be volcano, be moved out of their places, enjoy the exhilaration, meet the ocean, be available for hire, keep their secrets, win the mwc in 2010, ..]
R_Q: [affect rainfall, affect the climate of an area, affect climate, be measured, be formed, be created, be made, grow, affect weather, have snow on top, affect solar radiation, affect temperature, be formed ks2, affect the weather, be built, affect people, look blue, tops cold, affect neighboring climates, be formed video, help shape the development of greek civilization, be made for kids, occur, affect the climate, be formed, be formed wikipedia, have roots, affect precipitation, exist, affect life on earth, be formed kids, float in avatar, erode, have snow on the top, affect the political character of greece, help rain form, ..]

Table 5: Comparative top facts extracted for a sample of classes from documents (R_D) or queries (R_Q)

queries have higher levels of accuracy.

Facts from Documents vs. Queries: Table 5 compares the top facts extracted by the two experimental runs for a sample of target classes. Most commonly, erroneous facts are extracted by run R_D due to the extraction of relatively uninteresting properties (a *Company* may *"say in a statement"* or *"be in the process"*). Other errors in R_D are caused by wrong boundary detection of facts within documents (a *Company* may *"be in the midst"*), or by the association of a fact with the wrong instance or class (a *Car* may *"hire companies"* or *"hire services"*).

As for facts extracted by run R_Q, they are sometimes too informal, due to the more conversational nature of queries when compared to documents. Queries may suggest that a *Car* may *"know tire pressure"*. Occasionally, similarly to facts from documents, they have wrong boundaries (a *Mountain* may *"be made for kids"* or *"be formed wikipedia"*); and they may correspond to less interesting, or too specific, properties (a *Company* may *"incorporate in delaware"*). Lastly, queries may appear to be questions, but occasionally they really are not. An example is the query *"why did the actor jump out of the window in times square"*, which may refer to a joke. When such queries match one of the extraction patterns, they produce wrong facts. Overall, Table 5 corroborates the scores from Table 4. It suggests that a) facts extracted by either R_D or R_Q still need refinement, before they can capture essential characteristics of the respective classes and nothing else; and b) facts extracted in run R_Q have higher quality than facts extracted in run R_D. Indeed, because fact-seeking queries inquire about the value (or reason, or manner) of some relations of an instance, the facts themselves tend to be more relevant than facts extracted from arbitrary document sentences.

An issue related to facts extracted from text

is their ability to capture the kind of "obvious" commonsense knowledge (Zang et al., 2013) that would be essential for machine-driven reasoning. If it is obvious that *"teachers give lectures"*, how likely is it for such information to be explicitly stated in documents or, even more interestingly, inquired about in queries? Anecdotal evidence gathered during experimentation suggests that queries do produce many commonsense facts, perhaps even surprisingly so given that a) queries tend to be shorter and grammatically simpler than document sentences; and b) the patterns in Table 1 are relatively more restrictive than the patterns used in (Fader et al., 2011). Indeed, the patterns in Table 1, when applied to queries like *"why do teachers give homework"*, *"why do teachers give grades"*, actually produce commonsense knowledge that *teachers* give *homework*, *grades* (to their students). In fact, the quality of equivalent facts extracted from documents in (Fader et al., 2011) may be lower. Concretely, facts extracted in (Fader et al., 2011) state that what *teachers* give is *students*, *class*, *homework* and *feedback*, in this order. The first two of these extractions are errors, likely caused by the incorrect detection of complex entities and their inter-dependencies in document sentences (Downey et al., 2007).

A necessary condition for the usefulness of extracted facts is that the source text contain consistent, true information. But both documents and queries may contain contradictory or false information, whether due to unsupported conjectures, unintended errors or systematic campaigns that fall under the scope of adversarial information retrieval (Castillo and Davison, 2011). The phenomena potentially affect prior work on Web-based open-domain extraction, and potentially affect the quality of facts extracted from queries in this paper. For example, facts extracted from queries like *"why do companies like obamacare"* and *"why do companies hate obamacare"* would be inconsistent, if not incorrect.

Occasionally, facts extracted from the two text sources refer to the same properties. For example, a *VideoGame* may *"be good for the hand-eye coordination"*, according to documents; and may *"improve hand eye coordination"*, according to queries. Nevertheless, facts derived from queries likely serve as a complement, rather than replacement, of facts from documents. In particular, facts extracted from queries make no attempt to isolate the value of the respective properties, whereas facts extracted from documents usually do.

Stricter Comparison of Data Sources: In the experiments described so far, distinct sets of patterns are applied in the experimental runs to documents vs. queries. More precisely, run R_D applies the patterns introduced in (Fader et al., 2011) to document sentences, whereas run R_Q the patterns shown in Table 1 to queries. To more accurately gauge the role of queries vs. documents in extracting facts from unstructured text, additional experiments isolate the effect of extracting facts from different types of data sources. For this purpose, the same set of patterns from Table 1 is matched against the sentences from around 500 million Web documents. The patterns are applied to document sentences converted to lowercase, similarly to how they are applied to queries. This corresponds to a new experimental run R_{DS}, which employs the same patterns as the earlier run R_Q but runs over document sentences instead of queries.

As an average over the target classes, the precision of facts extracted by run R_{DS} is 0.50, 0.47 and 0.44 at ranks 10, 20 and 50 respectively. Two conclusions can be drawn from comparing these scores with the average scores from the earlier Table 4. First, the average precision of run R_{DS} is higher than for run R_D. In other words, when extracting from document sentences in R_{DS} and R_D, the patterns proposed in our method give fewer and more accurate facts than the patterns from (Fader et al., 2011). Second, although R_{DS} is more accurate than R_D, it is less accurate than run R_Q. Note that, among the top 50 facts extracted for each target class by runs R_{DS} and R_Q, an average of 13% of the facts are extracted by both runs. There are several phenomena contributing to the difference in precision. While inherently noisy, queries tend to be more compact, and therefore more focused. In comparison, document sentences matching the patterns are often more convoluted (e.g., *"who do cities keep building stadiums despite study after study showing they do not make money"*, or *"how does a company go from low associate satisfaction to #15 on the fortune 100 best list in the midst of a crippling recession"*). Furthermore, both queries and sentences may not be useful questions from which relevant facts can be extracted, even when they match the extraction patterns. However, anecdotal evidence suggests

that this happens more frequently with document sentences than with queries. Examples include document sentences extracted from sites aggregating jokes (*"why did the cell phone ask to see the psychologist"*). The results confirm that queries represent an intriguing resource for fact extraction, providing a useful complement to document sentences for the purpose of extracting facts.

Quantitative Results: From the set of queries used as input in run R_Q, 3.8% of all queries start with *why* or *how*. In turn, 13.6% of them match one of the extraction patterns from Table 1, and therefore produce a candidate fact in R_Q. In the case of run R_{DS}, 18.7% of the document sentences that start with *why* or *how* match one of the patterns from Table 1.

Choice of Extraction Patterns: The sets of patterns sometimes employed in relation extraction from documents (Hearst, 1992) occasionally benefit from the addition of new patterns, or the refinement into more specific patterns (Kozareva et al., 2008). Similarly, the set of patterns proposed in Table 1, which targets the extraction of facts from queries, is neither exhaustive nor final. Other patterns beyond *why* and *how* may prove useful, whether they rely on relatively less frequent *when* and *where* queries, or extract relations containing underspecified arguments from *who* or *what* queries.

When applied to queries in run R_Q, the *how* patterns from Table 1 match 3.3 times more queries than the *why* patterns.

In separate experiments, *why* vs. *how* patterns from Table 1 are temporarily disabled. The ratio of facts extracted on average per target class in run R_Q diminishes from 100% (with both patterns) to 30% (with *why* only) or 70% (with *how* only). Overall, no difference in accuracy is observed over facts extracted by *why* vs. *how* patterns.

Choice of Phrase Descriptors: A separate experiment investigates the impact of expanding the sets of phrase descriptors associated with each target class. Among many possible strategies, each set of phrase descriptors associated with a target class is expanded automatically, using WordNet and distributional similarities. For this purpose, for each target class, the set of synonyms and hyponyms of all senses, if any, available in WordNet for each phrase descriptor is intersected with the set of the 50 most distributionally similar phrases, if any, available for each phrase descriptor. The original set of phrase descriptors of each target class is then expanded, to include the phrases from the intersected set, if any.

A repository of distributionally similar phrases is collected in advance following (Lin and Wu, 2009; Pantel et al., 2009), from a sample of around 200 million Web documents. Their intersection with phrases collected from WordNet aims at reducing the noise associated with expansion solely from either source. For example, for the class *Actor*, the set of phrases {*player*, *worker*, *heavy*, *plant*, *actress*, *comedian*, *film star*, ..} is collected from WordNet for the descriptor *actors*. The set is intersected with the set of phrases {*film stars*, *performers*, *comedians*, *actresses*, ..} most distributionally similar to *actors*. Examples of sets of phrase descriptors after expansion are {*actors*, *actresses*, *comedians*, *players*, *film stars*, ..}, for the class *Actor*; and {*battles*, *naval battles*, *fights*, *skirmishes*, *struggles*, ..}, for *Battle*.

On average, the sets of phrase descriptors associated with each target class contains 2 vs. 11 phrases, before vs. after expansion. Some of the sets of phrase descriptors, such as for the target classes *CartoonChar* and *DigitalCamera*, remain unchanged after expansion. As expected, expansion may introduce noisy phrase descriptors, such as *players* for *Actor*, or *diets* for *Food*. The presence of noisy phrase descriptors lowers the precision of the extracted facts. After expansion, the precision scores of R_Q, as an average over all target classes, become smaller by 6% (0.71 vs. 0.67), at rank 10; 6% (0.67 vs. 0.63), at rank 20; and 7% (0.63 vs. 0.59), at rank 50. Expansion also affects relative coverage, increasing the average number of facts extracted by R_Q per target class by more than twice (i.e., by a factor of 2.6).

Redundant Facts: Due to lexical variation in the source text fragments, some of the extracted facts may be near-duplicates of one another. In general, the phenomenon affects facts extracted from text by previous methods (Van Durme and Paşca, 2008; Etzioni et al., 2011; Fader et al., 2011). In particular, it affects facts extracted from both documents or queries in our experiments. For example, the facts extracted from documents for *Actor* include *"play their roles"*, *"play their parts"*, *"play their characters"* and *"portrayed their characters"*. Separately, the facts *"memorize their lines"*, *"remember their lines"* and *"learn their lines"* are extracted from queries for the class

Actor. The automatic detection of equivalent facts would increase the usefulness of facts extracted from text in general, and of facts extracted by the method presented here in particular.

5 Related Work

A variety of methods address the more general task of acquisition of open-domain relations from text, e.g., (Banko et al., 2007; Carlson et al., 2010; Wu and Weld, 2010; Fader et al., 2011; Lao et al., 2011; Mausam et al., 2012; Lopez de Lacalle and Lapata, 2013). In general, relations extracted from document sentences (e.g., *"Claude Monet was born in Paris"*) are tuples of an argument (*claude monet*), a text fragment acting as the lexicalized relation (*was born in*), and another argument (*paris*) (cf. (Banko et al., 2007; Fader et al., 2011; Mausam et al., 2012)). For convenience, the relation and second argument may be concatenated into a fact applying to the first argument, as in *"was born in paris"* for *claude monet*. Relatively shallow tools like part of speech taggers, or more complex tools like semantic taggers (Van Durme et al., 2008; Van Durme et al., 2009) can be employed in order to extract relations from document sentences. The former choice scales better to Web documents of arbitrary quality, whereas the latter could be more accurate over high-quality documents such as news articles (Mesquita et al., 2013). In both cases, document sentences mentioning an instance or a class may refer to properties of the instance that people other than the author of the document are less likely to inquire about. Consequently, even top-ranked extracted relations occasionally include less informative ones, such as *"come into view"* for *mount rainier*, *"be on the table"* for *madeira wine*, or *"allow for features"* for *javascript* (Fader et al., 2011).

Data available within Web documents, from which relations are extracted in previous work, includes unstructured (Banko et al., 2007; Fader et al., 2011), structured (Raju et al., 2008) and semi-structured text (Yoshinaga and Torisawa, 2007; Pasupat and Liang, 2014), layout formatting tags (Wong et al., 2008), itemized lists or tables (Cafarella et al., 2008). Another source is human-compiled resources (Wu and Weld, 2010) including infoboxes and category labels (Nastase and Strube, 2008; Hoffart et al., 2013; Wang et al., 2013; Flati et al., 2014) in Wikipedia, or topics and relations in Freebase (Weston et al., 2013; Yao and Van Durme, 2014).

Whether Web search queries are a useful textual data source for open-domain information extraction has been investigated in several tasks. Examples are collecting unlabeled sets of similar instances (Jain and Pennacchiotti, 2010), extracting attributes of instances (Alfonseca et al., 2010; Paşca, 2014), identifying mentions in queries of instances defined in a manually-created resource (Pantel et al., 2012), and extracting the most salient of the instances mentioned within Web documents (Gamon et al., 2013).

Other previous work shares the intuition that the submission of Web search queries is influenced by, and indicative of, various relations. Relations are loosely defined, either by approximating them via distributional similarities (Alfonseca et al., 2009), or by exploring the acquisition of untyped, similarity-based relations from query logs (Baeza-Yates and Tiberi, 2007). In both cases, the computed relations hold among full-length queries. Untyped relations can also be identified among query terms for the purpose of query reformulation (Wang and Zhai, 2008). More generally, the choice of query substitutions may reveal various relations among full queries or query terms (Jones et al., 2006), but requires individual queries to be connected to one another via query sessions or via search-result click-through data.

6 Conclusion

Anonymized search queries submitted by Web users represent requests for knowledge. Collectively, they can also be seen as informal, lexicalized knowledge assertions. By asking about a property of some class, fact-seeking queries implicitly assert the relevance of the property for the class.

Since Web search queries refer to properties that Web users are collectively interested in, factual knowledge extracted from queries tends to be more relevant than facts extracted from arbitrary documents using previous methods. Current work explores the extraction of facts from implicit rather than explicit fact-seeking questions, that is, from queries that do not start with a question prefix; and the combination of queries as a source of more accurate facts, and documents as a source of more numerous facts.

References

E. Alfonseca, K. Hall, and S. Hartmann. 2009. Large-scale computation of distributional similarities for queries. In *Proceedings of the 2009 Conference of the North American Association for Computational Linguistics (NAACL-HLT-09), Short Papers*, pages 29–32, Boulder, Colorado.

E. Alfonseca, M. Paşca, and E. Robledo-Arnuncio. 2010. Acquisition of instance attributes via labeled and related instances. In *Proceedings of the 33rd International Conference on Research and Development in Information Retrieval (SIGIR-10)*, pages 58–65, Geneva, Switzerland.

R. Baeza-Yates and A. Tiberi. 2007. 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.

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

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.

C. Castillo and B. Davison. 2011. Adversarial web search. *Journal of Foundations and Trends in Information Retrieval*, 4(5):377–486.

J. Dalton, L. Dietz, and J. Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International Conference on Research and Development in Information Retrieval (SIGIR-14)*, pages 365–374, Gold Coast, Queensland, Australia.

D. Downey, M. Broadhead, and O. Etzioni. 2007. 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.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the Web: an experimental study. *Artificial Intelligence*, 165(1):91–134.

O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 3–10, Barcelona, Spain.

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.

T. Flati, D. Vannella, T. Pasini, and R. Navigli. 2014. Two is bigger (and better) than one: the Wikipedia Bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-14)*, pages 945–955, Baltimore, Maryland.

M. Gamon, T. Yano, X. Song, J. Apacible, and P. Pantel. 2013. Identifying salient entities in web pages. In *Proceedings of the 22nd International Conference on Information and Knowledge Management (CIKM-13)*, pages 2375–2380, Burlingame, California.

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

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

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.

R. Jones, B. Rey, O. Madani, and W. Greiner. 2006. Generating query substitutions. In *Proceedings of the 15h World Wide Web Conference (WWW-06)*, pages 387–396, Edinburgh, Scotland.

Z. Kozareva and E. Hovy. 2010. 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.

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic class learning from the Web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 1048–1056, Columbus, Ohio.

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.

D. Lin and X. Wu. 2009. 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.

O. Lopez de Lacalle and M. Lapata. 2013. Unsupervised relation extraction with general domain knowledge. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP-13)*, pages 415–425, Seattle, Washington.

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.

F. Mesquita, J. Schmidek, and D. Barbosa. 2013. 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.

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. 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. 2014. Acquisition of noncontiguous class attributes from Web search queries. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL-14)*, pages 386–394, Gothenburg, Sweden.

P. Pantel and A. Fuxman. 2011. Jigs and lures: Associating web queries with structured entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-11)*, pages 83–92, Portland, Oregon.

P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas. 2009. 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.

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.

P. Pasupat and P. Liang. 2014. Zero-shot entity extraction from Web pages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-14)*, pages 391–401, Baltimore, Maryland.

M. Pennacchiotti and P. Pantel. 2009. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 238–247, Singapore.

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.

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.

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

S. Sekine and H. Suzuki. 2007. Acquiring ontological knowledge from query logs. In *Proceedings of the 16th World Wide Web Conference (WWW-07), Posters*, pages 1223–1224, Banff, Canada.

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 and M. Paşca. 2008. 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.

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.

B. Van Durme, P. Michalak, and L. Schubert. 2009. Deriving generalized knowledge from corpora using Wordnet abstraction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, pages 808–816, Athens, Greece.

R. Wang and W. Cohen. 2009. 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.

X. Wang and C. Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th International Conference on Information and Knowledge Management (CIKM-08)*, pages 479–488, Napa Valley, California.

Z. Wang, Z. Li, J. Li, J. Tang, and J. Pan. 2013. Transfer learning based cross-lingual knowledge extraction for Wikipedia. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-13)*, pages 641–650, Sofia, Bulgaria.

J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP-13)*, pages 1366–1371, Seattle, Washington.

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.

W. Wu, , H. Li, H. Wang, and K. Zhu. 2012. 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.

X. Yao and B. Van Durme. 2014. Information extraction over structured data: Question Answering with Freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-14)*, pages 956–966, Baltimore, Maryland.

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.

L. Zang, C. Cao, Y. Cao, Y. Wu, and C. Cao. 2013. A survey of commonsense knowledge acquisition. *Journal of Computer Science and Technology*, 28(4):689–719.