Open-Domain Fine-Grained Class Extraction from Web Search Queries

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

Abstract

This paper introduces a method for extracting fine-grained class labels (*"countries with double taxation agreements with india"*) from Web search queries. The class labels are more numerous and more diverse than those produced by current extraction methods. Also extracted are representative sets of instances (*singapore*, *united kingdom*) for the class labels.

1 Introduction

Motivation: As more semantic constraints are added, concepts like *companies* become more specific, e.g., *companies* that are in the *software* business, and have been *started in a garage*. The sets of instances associated with the classes become smaller; the class labels used to concisely describe the meaning of more specific concepts tend to become longer. In fact, fine-grained class labels such as *"software companies started in a garage"* are often complex noun phrases, since they must somehow summarize multiple semantic constraints. Although Web users are interested in both coarse (e.g., *"companies"*) and fine-grained (e.g., *"software companies started in a garage"*) class labels, virtually all class labels acquired from text by previous extraction methods (Etzioni et al., 2005; Van Durme and Paşca, 2008; Kozareva and Hovy, 2010; Snow et al., 2006) exhibit little syntactic diversity. Indeed, instances and class labels that are relatively complex nouns are known to be difficult to detect and pick out precisely from surrounding text (Downey et al., 2007). This and other challenges associated with large-scale extraction from Web text (Etzioni et al., 2011) cause the extracted class labels to usually follow a rigid modifiers-plus-nouns format. The format covers nouns (*"companies"*) possibly preceded by one or many modifiers (*"software companies"*, *"computer security software companies"*). Examples of actual extractions include *"european cities"* (Etzioni et al., 2005), *"strong acids"* (Pantel and Pennacchiotti, 2006), *"prestigious private schools"* (Van Durme and Paşca, 2008), *"aquatic birds"* (Kozareva and Hovy, 2010).

As an alternative to extracting class labels from text, some methods simply import them from human-curated resources, for example from the set of categories encoded in Wikipedia (Remy, 2002). As a result, class labels potentially exhibit higher syntactic diversity. The modifiers-plus-nouns format (*"computer security software companies"*) is usually still the norm. But other formats are possible: *"software companies based in london"*, *"software companies of the united kingdom"*. Vocabulary coverage gaps remain a problem, with many relevant class labels (*"software companies of texas"* *"software companies started in a garage"*, *"software companies that give sap training"*) still missing. There is a need for methods that more aggressively identify fine-grained class labels, beyond those extracted by previous methods or encoded in existing, manually-created resources. Such class labels increase coverage, for example in scenarios that enrich Web search results with instances available for the class labels specified in the queries.

Contributions: The contributions of this paper are twofold. First, it proposes a weakly-supervised

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 403–414,
Seattle, Washington, USA, 18-21 October 2013.

method to assemble a large vocabulary of class labels from queries. The class labels include fine-grained class labels (*"countries with double taxation agreements with india"*, *"no front license plate states"*) that are difficult to extract from text by previous methods for open-domain information extraction. Second, the method acquires representative instances (*singapore*, *united kingdom*; *arizona*, *new mexico*) that belong to fine-grained class labels (*"countries with double taxation agreements with india"*, *"no front license plate states"*). Both class labels and their instances are extracted from Web search queries.

2 Extraction from Queries

2.1 Extraction of Class Labels

Overview: Given a set of arbitrary Web search queries as input, our method produces a vocabulary of fine-grained class labels. For this purpose, it: a) selects an initial vocabulary of class labels, as a subset of input queries that are likely to correspond to search requests for classes; b) expands the vocabulary, by generating a large, noisy set of other possible class labels, through replacements of ngrams within initial class labels with their similar phrases; c) restricts the generated class labels to those that match the syntactic structure of class labels within the initial vocabulary; and d) further restricts the generated class labels to those that appear within the larger set of arbitrary Web search queries.

Initial Vocabulary of Class Labels: Out of a set of arbitrary search queries available as input, the queries in the format *"list of .."* are selected as the initial vocabulary of class labels. The prefix *"list of"* is discarded from each query. Thus, the query *"list of software companies that use linux"* gives the class label *"software companies that use linux"*.

Generation via Phrase Similarities: As a prerequisite to generating class labels, distributionally similar phrases (Lin and Pantel, 2002; Lin and Wu, 2009; Pantel et al., 2009) and their scores are collected in advance. A phrase is represented as a vector of its contextual features. A feature is a word, collected from windows of three words centered around the occurrences of the phrase in sentences across Web documents (Lin and Wu, 2009). In the contextual vector of a phrase, the weight of a feature is the pointwise-mutual information (Lin and Wu, 2009) between the phrase P and the feature F. The distributional similarity score between two phrases is the cosine similarity between the contextual vectors of the two phrases. The lists of most distributionally similar phrases of a phrase P are thus compiled offline, by ranking the similar phrases of P in decreasing order of their similarity score relative to P.

Each class label from the initial vocabulary is expanded into a set of generated, candidate class labels. To this effect, every ngram P within a given class label is replaced with each of the distributionally similar phrases, if any, available for the ngram. As shown later in the experimental section, the expansion can increase the vocabulary by a factor of 100.

Approximate Syntactic Filtering: The set of generated class labels is noisy. The set is filtered, by retaining only class labels whose syntactic structure matches the syntactic structure of some class label(s) from the initial vocabulary. The syntactic structure is loosely approximated at surface rather than syntactic level. A generated class label is retained, if its sequence of part of speech tags matches the sequence of part of speech tags of one of the class labels from the initial vocabulary. As an additional constraint, the sequence must contain one tag corresponding to a common noun in plural form, i.e., NNS. Otherwise, the class label is discarded.

Query Filtering: Generated class labels that pass previous filters are further restricted. They are intersected with the set of arbitrary Web search queries available as input. Generated class labels that are not full queries are discarded.

2.2 Extraction of Instances

Overview: Our method mines instances of fine-grained class labels from queries. In a nutshell, it identifies queries containing two types of information simultaneously. First, the queries contain an instance (*marvin gaye*) of the more general class labels (*"musicians"*) from which the fine-grained class labels (*"musicians who have been shot"*) can be obtained. Second, the queries contain the constraints added by the fine-grained class labels (*"... shot"*) on top of the more general class labels.

Instances of General Class Labels: Following (Ponzetto and Strube, 2007), the Wikipedia category network is refined into a hierarchy that discards

non-IsA (thematic) edges, and retains only IsA (subsumption) edges from the network (Ponzetto and Strube, 2007). Instances, i.e., titles of Wikipedia articles, are propagated upwards to all their ancestor categories. The class label *"musicians"* would be mapped into *madonna, marvin gaye, jon bon jovi* etc. The mappings from each ancestor category, to all its descendant instances in the Wikipedia hierarchy, represent our mappings from more general class labels to instances.

Decomposition of Fine-Grained Class Labels: A fine-grained class label (e.g., *"musicians who have been shot"*) is effectively decomposed into pairs of two pieces of information. The first piece is a more general class label (*"musicians"*), if any occurs in it. The second piece is a bag of words, collected from the remainder of the fine-grained class label after discarding stop words. Note that the standard set of stop words is augmented with auxiliary verbs (e.g., *does*, *has*, *is*, *would*), determiners, conjunctions, prepositions, and question wh-words (Radev et al., 2005) (e.g., *where*, *how*). In the first piece of each pair, the general class label is then replaced with each of its instances. This produces multiple pairs of a candidate instance and a bag of words, for each fine-grained class label. As an illustration, the class labels *"musicians who have been shot"* and *"automobiles with remote start"* are decomposed into pairs like <*madonna*, {*shot*}>, <*marvin gaye*, {*shot*}>; and <*buick lacrosse*, {*remote*, *start*}>, <*nissan versa*, {*remote*, *start*}>, respectively.

Matching of Candidate Instances: A decomposed class label is retained, if there are matching queries that contain the candidate instance, the bag of words, and optionally stop words. Otherwise, the decomposed class label is discarded. The word matching is performed after word stemming (Porter, 1980). The aggregated frequency of the matching queries is assigned as the score of the candidate instance for the fine-grained class label:

$$Score(I, C) = \sum_{Q} (Freq(Q) | Match(Q, < I, C >)) \quad (1)$$

For example, the score of the candidate instance *marvin gaye* for the class label *"musicians who have been shot"*, is the sum of the frequencies of the matching queries *"marvin gaye is shot"*, *"when was marvin gaye shot"*, *"why marvin gaye was shot"* etc. Similarly, the score of *buick lacrosse* for *"automobiles with remote start"* is given by the aggregated frequencies of the queries *"buick lacrosse remote start"*, *"how to remote start buick lacrosse"*, *"remote start for buick lacrosse"*. Candidate instances of a class label are ranked in decreasing order of their scores.

3 Experimental Setting

Web Textual Data: The experiments rely on a sample of 1 billion queries in English submitted by users of a Web search engine. Each query is accompanied by its frequency of occurrence. Also available is a sample of around 200 million Web documents in English.

Phrase Similarities: Web documents are used in the experiments only to construct a phrase similarity repository following (Lin and Wu, 2009; Pantel et al., 2009). The repository contains ranked lists of the top 1000 phrases, computed to be the most distributionally similar to each of around 16 million phrases.

Text Pre-Processing: The TnT tagger (Brants, 2000) assigns part of speech tags to words in class labels.

Instances: To collect mappings from Wikipedia categories (as more general class labels) to titles of descendant Wikipedia articles (as instances), a snapshot of Wikipedia articles was intersected with the Wikipedia category hierarchy from (Ponzetto and Strube, 2007). The mappings connect a total of 1,535,083 instances to a total of 108,756 class labels.

4 Evaluation of Class Labels

4.1 Evaluation Procedure

Experimental Runs: Human-compiled information available within Wikipedia serves as the source of data for two baseline runs. The set of all categories, listed in Wikipedia for any of its articles, corresponds to the set of class labels "acquired" in run R_{wc}. Categories used for internal Wikipedia bookkeeping (Ponzetto and Strube, 2007) are discarded. Their names contain one of the words *article(s)*, *category(ies)*, *indices*, *pages*, *redirects*, *stubs*, or *templates*. Similarly, the titles of Wikipedia articles with the prefix *"List of ..."* (e.g., *"List of automobile manufacturers of Germany"*) form the set of class labels

"acquired" in run R_{wl}. The prefix *"List of"* is discarded.

For completeness, a third baseline run, R_{dc}, corresponds to class labels extracted from Web documents. The class labels are noun phrases *C* that fill extraction patterns equivalent to *"C such as I"*. The patterns are matched to document sentences. The boundaries of the class labels *C* are approximated from part of speech tags of sentence words (Van Durme and Paşca, 2008). The patterns were proposed in (Hearst, 1992). They were employed widely in subsequent methods (Etzioni et al., 2005; Kozareva et al., 2008; Wu et al., 2012), which extract class labels precisely from the set of class labels *C* produced by the extraction patterns. Even methods using queries as a textual data source still extract class labels from documents using the same extraction patterns (Paşca, 2010). Therefore, from the point of view of evaluating class labels, run R_{dc} is a valid representative of previous extraction methods, including (Etzioni et al., 2005; Kozareva et al., 2008; Van Durme and Paşca, 2008; Paşca, 2010; Wu et al., 2012).

Besides the baseline runs, three experimental runs are considered. In run R_{ql}, the queries starting with the prefix *"list of"* form the set of class labels. The prefix *"list of"* is discarded from each query. In run R_{qg}, the class labels are generated via phrase similarities, starting from R_{ql} as an initial set of class labels. Run R_{qa} represents an ablation experiment. It is created from R_{qg}, by limiting the expansion of a given class label via distributional similarities to only one, rather than multiple, phrases within the class label. Note that, by design, none of the class labels that appear in R_{ql} also appear in runs R_{qa} or R_{qg}. Therefore, the intersection between R_{ql}, on one hand, and R_{qa} and R_{qg}, on the other hand, is the empty set.

All data, including the class labels extracted in all experimental runs, is converted to lower case.

4.2 Relative Coverage of Class Labels

Coverage Over Entire Sets: Table 1 illustrates the overall coverage of the various experimental runs. The table takes all class labels into account, relative to the Wikipedia-based runs as reference sets: R_{wc} (Wikipedia categories), in the upper part of the table; and R_{wl} (Wikipedia List-Of categories), in the lower part of the table. Note that the number of class labels extracted by the individual run shown in the second column (B) is shown in the fourth column (|B|). In particular, there are around 1.6 million unique *"list of .."* queries, from which class labels are collected in run R_{ql}.

Counts					Cvg
A	B	|A|	|B|	|A∩B|	$\frac{\|A \cap B\|}{\|A\|}$
vs. Wikipedia categories:					
R_{wc}	R_{dc}	295,587	2,884,390	15,011	0.051
	R_{ql}	295,587	1,649,261	21,979	0.074
	R_{qa}	295,587	33,073,741	33,502	0.113
	R_{qg}	295,587	134,235,151	43,935	0.148
	$R_{ql} \cup R_{qg}$	295,587	135,884,412	65,914	0.222
vs. Wikipedia categories that are queries:					
$R_{wc} \cap Q$	R_{dc}	126,318	2,884,390	14,840	0.117
	R_{ql}	126,318	1,649,261	21,979	0.173
	R_{qa}	126,318	33,073,741	33,502	0.265
	R_{qg}	126,318	134,235,151	43,935	0.347
	$R_{ql} \cup R_{qg}$	126,318	135,884,412	65,914	0.521
vs. Wikipedia List-Of categories:					
R_{wl}	R_{dc}	134,840	2,884,390	8,099	0.060
	R_{ql}	134,840	1,649,261	26,446	0.196
	R_{qa}	134,840	33,073,741	16,204	0.120
	R_{qg}	134,840	134,235,151	20,021	0.148
	$R_{ql} \cup R_{qg}$	134,840	135,884,412	46,467	0.344
vs. Wikipedia List-Of categories that are queries:					
$R_{wl} \cap Q$	R_{dc}	47,442	2,884,390	7,985	0.168
	R_{ql}	47,442	1,649,261	24,821	0.523
	R_{qa}	47,442	33,073,741	16,204	0.341
	R_{qg}	47,442	134,235,151	20,021	0.422
	$R_{ql} \cup R_{qg}$	47,442	135,884,412	44,842	0.945

Table 1: Coverage of class labels extracted by various experimental runs, relative to class labels available in Wikipedia before and after intersecting them with a large set of arbitrary queries (A = reference set, relative to which coverage is computed; B = measured set, for which coverage is computed relative to the reference set; |A| = size of set A; Q = set of input queries)

During the computation of coverage, the reference set, and the set for which coverage is being computed, are intersected. Intersection relies on strict string matching. All words, including punctuation, must match exactly in order for a class label to be part of the intersection. The reference sets are intersected with the set of all Web search queries Q used in the experiments. Coverage is computed both before and after intersection. Less than half (126,318 of 295,587) of the class labels, for

the reference set R_{wc}; and about a third (47,442 of 134,840) for R_{wl}; appear in the set Q of all queries.

Three conclusions can be drawn from the results. First, query-based runs vastly outperform Wikipedia-based runs in terms of absolute coverage. Run R_{ql} contains around 5 and 12 times more class labels, than R_{wc} and R_{wl} respectively. On top of that, generating class labels via phrase similarities further increases the class label count by about 20 times for R_{qa}, and 80 times for R_{qg}. Second, query-based runs R_{qa} and R_{qg} surpass the document-based run R_{dc}. Third, higher class label counts translate into higher relative coverage. In the upper part of the table, run R_{wl} contains 3.9% (relative to R_{wc}) and 7.1% (relative to $R_{wc} \cap Q$) of the reference set. But the relative coverage doubles for R_{ql} at 7.4% (relative to R_{wc}) and 17.3% (relative to $R_{wc} \cap Q$). Coverage again doubles for R_{qg} at 14.8% (relative to R_{wc}) and 34.7% (relative to $R_{wc} \cap Q$). The union of query-based initial and generated class labels is $R_{ql} \cup R_{qg}$. The union contains about a quarter (i.e., 22.2%) or half (52.1%) of the reference set R_{wc}, depending on whether the reference set is intersected with the set of all queries or not. In the lower part of the table, more than 90% of the queries in the reference set R_{wl} that are also queries are found among the class labels collectively extracted in the query-based runs. Note that, since R_{ql} is disjoint from R_{qa} and R_{qg}, none of the class labels already in R_{ql} can be "re-discovered" (generated) again in R_{qa} or R_{qg}. Therefore, by experimental design, relative coverage scores of R_{ql} may be relatively difficult to surpass by R_{qa} or R_{qg} taken individually.

Diversity: Class labels restricted to those that have the format *".. that/which/who .."* are relatively more specific, e.g., *"grocery stores that double coupons in omaha"*, *"airlines which fly from santa barbara"*, *"writers who were doctors"*. The most frequent head phrases of such restricted class labels offer an idea about how diverse the class labels are. The counts of class labels for the most frequent head phrases are in the order of 10's in the case of R_{wl} vs. 10,000's for R_{qg}. In comparison, none of the class labels of run R_{dc} have this format. The lack of such class labels in run R_{dc}, and their smaller proportion in run R_{wl} vs. R_{qg}, suggest that class labels extracted by the proposed method exhibit higher lexical and syntactic diversity than previous methods do.

Tag (Value): Examples of Class Labels
correct (1.0): angioplasty specialists in kolkata, good things pancho villa did, eating disorders inpatient units in the uk nhs specialist services
questionable (0.5): picture framers adelaide cbd, side effects bicalutamide, different eating disorders, private hospitals treat kidney stones uk
incorrect (0.0): al hirschfield theatre hours, value of berkshire hathaway shares, remove spaces in cobol, dogs with loss of appetite, 1999 majorca open

Table 2: Correctness tags manually assigned to class labels containing one of the (underlined) target phrases, extracted by various runs

4.3 Precision of Class Labels

Evaluation Metric: Class labels being evaluated are manually assigned a correctness tag. A class label is deemed *correct*, if it is grammatically well-formed and describes a relevant concept that embodies some (unspecified) set of instances that share similar properties; *questionable*, if it is relevant but not well-formed; or *incorrect*. A *questionable* class label is not well-formed because it lacks necessary linking particles (e.g., the prepositions *of* or *for* in *"side effects bicalutamide"*), or contains undesirable modifiers (*"different eating disorders"*). Examples of *correct* and *incorrect* class labels are *"angioplasty specialists in kolkata"* and *"al hirschfield theatre hours"* respectively.

To compute the precision score, the correctness tags are converted to numeric values, as shown in Table 2: *correct* to 1; *questionable* to 0.5; and *incorrect* to 0. Precision over a list of class labels is measured as the sum of the correctness values of the class labels in the list, divided by the size of the list.

Precision Relative to Target Phrases: The precision of the class labels in each run is determined similarly to how relative coverage was computed earlier. More precisely, the precision is computed over the class labels whose names contain each phrase from the set of 75 target phrases from (Alfonseca et al., 2010). For each phrase, and for each run, a random sample of at most 50 of the class labels that match the phrase is selected for evaluation. The samples taken for each run, corresponding to the same phrase, are combined into a merged list. This produces one merged list for each phrase, for a total of 75 merged lists. The precision score over a target

phrase is the precision score over its sample of class labels.

The last two columns of Table 3 capture the precision scores for the class labels. The scores are computed in two ways: averaged over the (variable) subsets of target phrases for which some matching class label(s) exist, in the last but one column, e.g., over 19 of the 75 target phrases for R_{wc}; and averaged over the entire set of 75 target phrases, in the last column. The former does not penalize a run for not being able to extract any class labels containing a particular target phrase, whereas the latter does penalize. Naturally, precision scores over the entire set of target phrases decrease when coverage is lower, for runs R_{wc}, R_{wl} and, to a lesser extent, R_{dc} and R_{ql}. But even after ignoring target phrases with no matching class labels, precision scores in the last but one column in Table 3 reveal important properties of the experimental runs. First, between the two Wikipedia-based runs, R_{wl} has perfect class labels, whereas as many as 1 in 4 class labels of run R_{wc} are marked as incorrect during the evaluation. Second, the class labels collected from *"list of .."* queries in run R_{ql} correspond to relevant, well-formed concepts in 80% of the cases. Third, the generation of class labels via phrase similarities (R_{qg}) greatly increases coverage as shown earlier. The increase comes at the expense of lowering precision from 80% to 72%. However, the phrases from initial queries that are expanded via distributional similarities can be limited from multiple to only one, by switching from R_{qg} to R_{qa}. This gives higher precision for R_{qa} than for R_{qg}.

As a complement to Table 3, the graphs in Figure 1 offer a more detailed view into the precision of class labels. The figure covers a Wikipedia-based run (R_{wc}) and two query-based runs (R_{ql}, R_{qg}). The graphs show the precision scores, over each of the 75 target phrases. Among target phrases for which some matching class labels exist in the respective run, the target phrases with the lowest precision scores are *robotics* (score of 0.15) and *karlsruhe* (0.33), for R_{wc}; *carotid arteries* and *kidney stones*, both with a score of 0.00 because their matching class labels are all incorrect, for R_{dc}; *african population* and *chester arthur*, both with a score of 0.00 because their matching class labels are all incorrect, for R_{ql}; and *arlene martel* (0.00) and *right to vote* (0.25), for R_{qg}.

Run	Target Phrases			Precision of Class Labels Over Target Phrases	
	All	Matched	Cvg	Over Matched	Over All
R_{wc}	75	19	0.253	0.756	0.191
R_{wl}	75	15	0.200	1.000	0.200
R_{dc}	75	35	0.467	0.834	0.389
R_{ql}	75	48	0.640	0.800	0.512
R_{qa}	75	70	0.933	0.868	0.810
R_{qg}	75	73	0.973	0.724	0.705

Table 3: Precision of class labels that match (i.e., whose names contain) each target phrase, computed as an average over (variable) subsets of target phrases for which some matching class label(s) exist, and as an average over the entire set of 75 target phrases

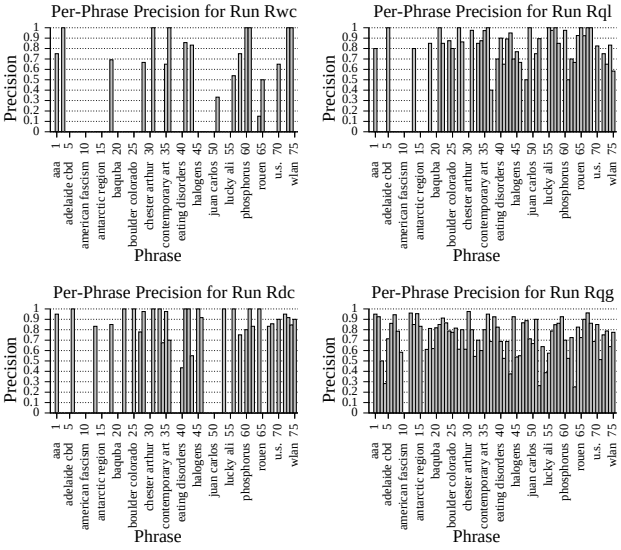

Figure 1: Precision scores for runs R_{wc}, R_{ql}, R_{dc} and R_{qg}, over class labels that match (i.e., contain) each of the 75 target phrases

Precision over Samples of Class Labels: The precision is separately computed over a random sample of 400 class labels per experimental run. The samples are selected from the set of all class labels extracted by the respective run. The precision scores are: 0.759 for R_{wc}; 1.000 for R_{wl}; 0.806 for R_{dc}; 0.811 for R_{ql}; 0.856 for R_{qa}; and 0.711 for R_{qg}. The scores are in line with scores computed earlier over the target phrases, in the fourth column of Table 3.

Discussion: As noted in (Ponzetto and Strube, 2007), Wikipedia organizes its articles and categories into a category network that mixes IsA (subsumption) edges with non-IsA (thematic) edges. Whenever an edge in Wikipedia is not IsA, the par-

Longest Class Labels
R_{wl}: [japanese army and navy members in military or politic services in proper japan korea manchuria occupied china and nearest areas in previous times and pacific war epoch(1930-40s), mental disorders as defined by the diagnostic and statistical manual of mental disorders and the international statistical classification of diseases and related health problems,..]
R_{qg}: [differences between transformational leadership and transactional leadership, things to do in llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch, philosophical differences between thomas jefferson and alexander hamilton, musculoskeletal manifestations of human immunodeficiency virus infection,..]

Table 4: Longest class labels extracted by runs R_{wl} and R_{qg}

ent category may not be a relevant concept that describes some set of instances that share similar properties. Such categories are not good class labels, and therefore are marked as incorrect. Examples include the class labels *"austrian contemporary art"*, *"1999 majorca open"* and *"u.s. route 30"*, listed in Wikipedia as categories of the instances *vienna biennale*, *1999 majorca open* and *squirrel hill tunnel* respectively. This affects the precision scores for R_{wc} in Table 3. It also affects the coverage values relative to R_{wc} in Table 1. Ideally, high-precision experimental runs would not extract any incorrect class labels that happen to appear in R_{wc}, for example *"austrian contemporary art"*. But the coverage relative to R_{wc} would artificially penalize such runs, for not extracting the incorrect class labels from R_{wc}.

As a proxy for estimating class label complexity, Table 4 shows the longest class labels derived from Wikipedia (R_{wl}) vs. generated from queries (R_{qg}).

Class labels derived from Web search queries may be semantically overlapping. Examples are *"writers who killed themselves"* vs. *"writers who committed suicide"*. The overlap is desirable, since different Web users may request the same information via different queries. The same phenomenon has been observed in other information extraction tasks. It also affects manually-created resources like Wikipedia. The continuous manual refinements to Wikipedia content still cannot prevent the occurrence of duplicate class labels among Wikipedia List-Of categories. The duplicates are present in run R_{wl}. Examples are *"formula one drivers that never qualified for a race"* vs. *"formula one drivers who never qualified for a race"*; or *"goaltenders who have scored a goal in a nhl game"* vs. *"goaltenders who have scored a goal in an nhl game"*. Some of the lexical differences among class labels are due to undesirable misspellings. Again, similar problems occasionally affect existing Wikipedia categories: *"nobel laureates who endorse barack obama"* vs. *"nobel laureates who endorse barrack obama"*.

Target Class Labels
007 movie actors, .308 weapons, actors with obsessive compulsive disorder, antibiotics for multiple sclerosis, astronauts in space station, automobiles with remote start, beatles songs of love, beetles that bite, companies with sustainable competitive advantage, countries with double taxation agreements with india, criminals who have been executed, daft punk live albums, dallas medical companies, direct democracy states, electronic companies in electronic city bangalore, expensive brands of shoes, eye diseases in cats, f1 car companies, fwd sports cars, garden landscaping magazines, heliskiing resorts, hell in a cell wrestlers, holidays celebrated in sydney, ibf weight classes, ibiza 2011 djs, immunology scientists, jewelry manufacturing companies, kanye west songs on youtube, kingston upon thames supermarkets, latin military ranks, ludhiana newspapers, maastricht treaty countries, musicians who have been shot, no front license plate states, non-profit organizations in nashville tennessee, organic chocolate companies, plants which are used in homeopathy, programming languages for server side programming, qatar chemical companies, qld private schools, real estate companies in virginia beach virginia, respiratory infection antibiotics, serial killers with antisocial personality disorder, singers with curly hair, telecommunications companies in the philippines, trains from la to san diego, visual basic database management systems, warmblood colors, washington university basketball players, world heritage sites in northern ireland

Table 5: Set of 50 class labels, used in the evaluation of extracted instances

5 Evaluation of Instances

5.1 Evaluation Procedure

Target Set of Class Labels: The target set for evaluation is shown in Table 5. Initially, a random sample of 100 class labels is selected from all class labels in

Tag (Value): Examples of Instances
correct (1.0): countries with double taxation agreements with india: thailand; hell in a cell wrestlers: brock lesnar; ibiza 2011 djs: dimitri from paris; heliskiing resorts: valle nevado
questionable (0.5): 007 movie actors: david niven; kanye west songs on youtube: the good life; holidays celebrated in sydney: waitangi day
incorrect (0.0): electronic companies in electronic city bangalore: bank of baroda; garden landscaping magazines: marquis; immunology scientists: rosalind franklin

Table 6: Correctness tags manually assigned to instances extracted from queries for various class labels

run R_{qg}. Class labels deemed incorrect, as well as class labels for which no instances are extracted, are manually removed from the sample. Out of the remaining class labels, a smaller random sample of 50 of the remaining class labels is retained, for the purpose of evaluating the quality of instances extracted for various class labels.

Evaluation Metric: The evaluation computes the precision of the ranked list of instances extracted for each target class label. To remove any undesirable bias towards higher-ranked instances, the ranked list is sorted alphabetically, then each instance is assigned one of the correctness tags from Table 6. Instances are deemed questionable, if they would be correct for a rather obscure interpretation of the class label. For example, *david niven* is an actor in one of the spoofs rather than main releases of the *007 movie*. Instances that would be correct if a few words were dropped or added are also deemed questionable: *the good life* is not one of the *"kanye west songs on youtube"* but *good life* is.

To compute the precision score over a ranked list of instances, the correctness tags are converted to numeric values. Precision at some rank N in the list is measured as the sum of the correctness values of the instances extracted up to rank N, divided by the number of instances extracted up to rank N.

5.2 Precision of Instances

Precision: Precision scores in Table 7 vary across target class labels. For some class labels, the extracted instances are noisy enough that scores are below 0.50 at ranks 10 and higher. This is the case for *"electronic companies in electronic city bangalore"* and *"daft punk live albums"*, and especially for *"garden landscaping magazines"* which has the worst precision. On the other hand, instances extracted for *"companies with sustainable competitive advantage"* or *"criminals who have been executed"* have high precision across all ranks. As an average over all target class labels, precision is 0.76 at rank 10, and 0.71 at rank 50. Although there is room for improvement, we find these accuracy levels to be encouragingly good, especially at rank 50. As a reminder, instances are extracted from noisy queries, and for class labels as fine-grained as those acquired and used in our experiments. Some of the extracted ranked lists of instances are shown in Table 8.

Target Class Label	Precision of Instances			
	@1	@5	@10	@50
007 movie actors	1.00	1.00	0.85	0.85
actors with obsessive compulsive disorder	0.00	0.60	0.70	0.70
antibiotics for multiple sclerosis	0.50	0.60	0.55	0.58
astronauts in space station	1.00	0.70	0.85	0.83
automobiles with remote start	1.00	1.00	0.75	0.75
beatles songs of love	0.00	0.50	0.65	0.52
beetles that bite	1.00	0.80	0.50	0.56
companies with sustainable competitive advantage	1.00	1.00	0.80	0.88
countries with double taxation agreements with india	1.00	1.00	1.00	0.90
criminals who have been executed	1.00	1.00	0.90	0.82
daft punk live albums	0.50	0.40	0.35	0.35
dallas medical companies	0.00	0.70	0.65	0.54
direct democracy states	1.00	1.00	0.90	0.86
electronic companies in electronic city bangalore	1.00	0.40	0.40	0.42
expensive brands of shoes	1.00	1.00	0.90	0.92
eye diseases in cats	0.50	0.50	0.35	0.35
f1 car companies	1.00	1.00	0.80	0.30
fwd sports cars	1.00	1.00	1.00	1.00
garden landscaping magazines	0.00	0.10	0.15	0.06
heliskiing resorts	1.00	1.00	1.00	1.00
hell in a cell wrestlers	1.00	1.00	1.00	0.92
holidays celebrated in sydney	1.00	0.70	0.75	0.75
...	...	...	...	...
Average over 50 class labels	0.80	0.80	0.76	0.71

Table 7: Precision at various ranks in the ranked lists of instances extracted from queries, for various target class labels and as an average over the entire set of 50 target class labels

Target Class Label	Extracted Instances
countries with double taxation agreements with india	[singapore, malaysia, mauritius, kenya, australia, united kingdom, cyprus, turkey, thailand, germany,..]
direct democracy states	[california, oregon, nevada, wisconsin, louisiana, arizona, vermont, alaska, illinois, michigan,..]
fwd sports cars	[scion tc, ford probe, honda prelude, nissan 200sx, lotus elan, mitsubishi fto, dodge srt-4, mitsubishi gto, volvo c30, toyota celica,..]
garden landscaping magazines	[front, contemporary, gallery, edge, view, chelsea, wallpaper, expo, wizard, sunset,..]
holidays celebrated in sydney	[halloween, australia day, anzac day, independence day, waitangi day, melbourne cup, hogmanay, rotuma day, solstice, yule,..]

Table 8: Ranked lists of instances extracted for a sample of class labels

In additional experiments, the same evaluation procedure is applied to output from two previous extraction methods. The first method starts by internally generating a small set of seed instances for a class label given as input (Wang and Cohen, 2009). A set expansion module then expands the seed set into a longer, ranked list of instances. The instances are extracted from unstructured and semi-structured text within Web documents. The documents are accessed via the search interface of a general-purpose Web search engine (cf. (Wang and Cohen, 2009) for more details). The second method extracts instances of class labels using the extraction patterns proposed in (Hearst, 1992). As such, it is similar to (Kozareva et al., 2008; Van Durme and Paşca, 2008; Wu et al., 2012). The method corresponds to the run R_{dc} described earlier, where the relative ranking of instances and class labels uses the co-occurrence of instances and class labels within queries (Paşca, 2010). For the purpose of the evaluation, when no instances are available for a target class label, the class label is generalized into iteratively shorter phrases containing fewer modifiers, until some instances are available for the shorter phrase. For example, target class labels like *actors with obsessive compulsive disorder*, *beatles songs of love*, *garden landscaping magazines* do not have any instances extracted by the second method. Therefore, the instances evaluated for the second method for these target class labels are collected from the instances of the more general *actors*, *beatles songs*, *landscaping magazines*. Without the generalization, the target class label would receive no credit during the evaluation, and the two previous methods would have lower precision scores. Over the 50 target class labels, the precision of the two methods is 0.11 and 0.27 at rank 5; 0.06 and 0.25 at rank 10; 0.05 and 0.22 at rank 20; and 0.05 and 0.20 at rank 50. The results confirm that, as explained earlier, previous methods for open-domain information extraction have limited ability to extract instances of fine-grained class labels.

Discussion: Earlier errors in the acquisition of the class label affect the usefulness of any instances that may be subsequently extracted for them. The experiments require candidate instances to appear in Wikipedia. This may improve precision, at the expense of not extracting instances that are not yet in Wikipedia (Lin et al., 2012).

6 Related Work

Previous methods for extracting classes of instances from text acquire sets of instances that are each either unlabeled (Pennacchiotti and Pantel, 2009; Jain and Pennacchiotti, 2010; Shi et al., 2010), or associated with a class label (Banko et al., 2007; Wang and Cohen, 2009). The sets of instances and/or class labels may be organized as flat sets or hierarchically, relative to inferred hierarchies (Kozareva and Hovy, 2010) or existing hierarchies such as WordNet (Snow et al., 2006; Davidov and Rappoport, 2009) or the category network within Wikipedia (Wu and Weld, 2008; Ponzetto and Navigli, 2009). Semi-structured text from Web documents is a complementary resource to unstructured text, for the purpose of extracting relations in general (Cafarella et al., 2008), and classes and instances in particular (Talukdar et al., 2008; Dalvi et al., 2012).

With previous methods, the vocabulary of class labels potentially produced for any instance is confined to a closed set provided manually as input (Wang and Cohen, 2009; Carlson et al., 2010). The closed set is often derived from resources like Wikipedia (Talukdar and Pereira, 2010; Lin et al.,

2012; Hoffart et al., 2013) or Freebase (Pantel et al., 2012). Alternatively, the vocabulary is not a closed set, but instead is acquired along with the instances (Pantel and Pennacchiotti, 2006; Snow et al., 2006; Banko et al., 2007; Van Durme and Paşca, 2008; Kozareva and Hovy, 2010). In the latter case, the extracted class labels take the form of head nouns preceded by modifiers. Examples are *"cities"*, *"european cities"* (Etzioni et al., 2005); *"artists"*, *"strong acids"* (Pantel and Pennacchiotti, 2006); *"outdoor activities"*, *"prestigious private schools"* (Van Durme and Paşca, 2008); *"methaterians"*, *"aquatic birds"* (Kozareva and Hovy, 2010). In contrast, the class labels extracted in our method exhibit greater syntactic diversity and are finer-grained. In addition, they are not constrained to a particular set of categories available in resources like Wikipedia.

Fine-grained class labels roughly correspond to queries submitted in typed search (Demartini et al., 2009) or entity search (Balog et al., 2010) or list-seeking questions (*"name the circuit judges in the cayman islands that are british"*). But our focus is on generating, rather than answering such queries or, more generally, attempting to deeply understand their semantics (Li, 2010). Phrase similarities can be derived with any methods, using documents (Lin and Wu, 2009) or search queries (Jain and Pennacchiotti, 2010).

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), ranking of class labels already extracted from text (Paşca, 2010), extracting attributes of instances (Alfonseca et al., 2010) and identifying the occurrences in queries of instances of several types, where the types are defined in a manually-created resource (Pantel et al., 2012). Comparatively, we show that queries are useful in identifying possible class labels, not only re-ranking them; and even in populating the class labels with relevant, albeit small, sets of corresponding instances.

As automatically-extracted class labels become finer-grained, they more clearly illustrate a phenomenon that received little attention. Namely, class labels of an instance, on one hand, and relations linking the instance with other instances and classes, on the other hand, are not mutually exclusive pieces of knowledge. Their extraction does not necessarily require different, dedicated techniques. Quite the opposite, class labels serve in text as nothing more than convenient lexical representations, or lexical shorthands, of relations linking instances with other instances. The class labels *"no front license plate states"* and *"states with no front license plate requirement"* are applicable to *arizona*. If so, it is because *arizona* is a *state*, and states require the installation of *license plates* on vehicles, and the requirement does not apply to the *front* of vehicles in the case of *arizona*. The connection between class labels and relations has been judiciously exploited in (Nastase and Strube, 2008). In that study, relations encoded implicitly within Wikipedia categories are transformed into explicit relations. As an example, the explicit relation that *deconstructing harry* is *directed* by *woody allen* is obtained from the fact that *deconstructing harry* is listed under *"movies directed by woody allen"* in Wikipedia. Ours is the first approach to examine the potential for extracting relations from search queries, where relations are compactly and loosely folded into the respective class labels. A variety of methods address the more general task of acquisition of open-domain relations from documents, e.g., (Zhu et al., 2009; Carlson et al., 2010; Fader et al., 2011; Lao et al., 2011).

7 Conclusion

The approach introduced in this paper exploits knowledge loosely encoded within Web search queries. It acquires a vocabulary of class labels that are finer grained than in previous literature. The class labels have precision comparable to that of class labels derived from human-created knowledge repositories. Furthermore, representative instances are extracted from queries for the fine-grained class labels, at encouraging levels of accuracy. Current work explores the use of noisy syntactic features to increase the accuracy of extracted class labels; the extraction of instances from evidence in multiple, rather than single queries; the expansion of extracted instances into larger sets; and the conversion of fine-grained class labels into relations among classes.

References

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.

K. Balog, M. Bron, and M. de Rijke. 2010. Category-based query modeling for entity search. In *Proceedings of the 32nd European Conference on Information Retrieval (ECIR-10)*, pages 319–331, Milton Keynes, United Kingdom.

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.

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

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.

B. Dalvi, W. Cohen, and J. Callan. 2012. Websets: Extracting sets of entities from the Web using unsupervised information extraction. In *Proceedings of the 5th ACM Conference on Web Search and Data Mining (WSDM-12)*, pages 243–252, Seattle, Washington.

D. Davidov and A. Rappoport. 2009. Enhancement of lexical concepts using cross-lingual Web mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 852–861, Singapore.

G. Demartini, T. Iofciu, and A. de Vries. 2009. Overview of the INEX 2009 Entity Ranking track. In *INitiative for the Evaluation of XML Retrieval Workshop*, pages 254–264, Brisbane, 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.

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*, 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.

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.

X. Li. 2010. Understanding the semantic structure of noun phrase queries. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 1337–1345, Uppsala, Sweden.

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

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.

T. Lin, Mausam, and O. Etzioni. 2012. No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-12)*, pages 893–903, Jeju Island, Korea.

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. 2010. The role of queries in ranking labeled instances extracted from text. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 955–962, Beijing, China.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 113–120, Sydney, Australia.

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.

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. Ponzetto and R. Navigli. 2009. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 2083–2088, Pasadena, California.

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

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

D. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. 2005. Probabilistic question answering on the Web. *Journal of the American Society for Information Science and Technology*, 56(3):571–583.

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

S. Shi, H. Zhang, X. Yuan, and J. Wen. 2010. Corpus-based semantic class mining: Distributional vs. pattern-based approaches. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 993–1001, Beijing, China.

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

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.

P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pages 582–590, Honolulu, Hawaii.

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.

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.

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

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.

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.