The Role of Queries in Ranking Labeled Instances Extracted from Text

Marius Paşca
Google Inc.
mars@google.com

Abstract

A weakly supervised method uses anonymized search queries to induce a ranking among class labels extracted from unstructured text for various instances. The accuracy of the extracted class labels exceeds that of previous methods, over evaluation sets of instances associated with Web search queries.

1 Introduction

Classes pertaining to unrestricted domains (e.g., *west african countries*, *science fiction films*, *slr cameras*) and their instances (*cape verde*, *avatar*, *canon eos 7d*) play a disproportionately important role in Web search. They occur prominently in Web documents and among search queries submitted most frequently by Web users (Jansen et al., 2000). They also serve as building blocks in formal representation of human knowledge, and are useful in a variety of text processing tasks.

Recent work on offline acquisition of fine-grained, labeled classes of instances applies manually-created (Banko et al., 2007; Talukdar et al., 2008) or automatically-learned (Snow et al., 2006) extraction patterns to large document collections. Although various methods exploit additional textual resources to increase accuracy (Van Durme and Paşca, 2008) and coverage (Talukdar et al., 2008), some of the extracted class labels are inevitably less useful (*works*) or spurious (*car makers*) for an associated instance (*avatar*). In Web search, the relative ranking of documents returned for a query directly affects the outcome of the search. Similarly, the relative ranking among class labels extracted for a given instance influences any applications using the labels.

Our paper proposes the use of features other than those computed over the underlying document collection, such as the frequency of co-occurrence or diversity of extraction patterns producing a given pair (Etzioni et al., 2005), to determine the relative ranking of various class labels, given a class instance. Concretely, the method takes advantage of the co-occurrence of a class label and an instance within search queries from anonymized query logs. It re-ranks lists of class labels produced for an instance by standard extraction patterns, to promote class labels that co-occur with the instance. This corresponds to a soft ranking approach, focusing on the ranking of candidate extractions such as the less relevant ones are ranked lower, as opposed to removed when deemed unreliable based on various clues.

By using queries in ranking, the ranked lists of class labels available for various instances are instrumental in determining the classes to which given sets of instances belong. The accuracy of the class labels exceeds that of previous work, over evaluation sets of instances associated with Web search queries. The results confirm the usefulness of the extracted IsA repository, which remains general-purpose and is not tailored to any particular task.

2 Instance Class Ranking

2.1 Extraction of Instances and Classes

The initial extraction of labeled instances relies on hand-written patterns from (Hearst, 1992), widely used in work on extracting hierarchies from text (Snow et al., 2006; Ponzetto and Strube,

Coling 2008: Poster Volume, pages 955–962,
Beijing, August 2010

2007):

$\langle$[..] $\mathcal{C}$ [such as|including] $\mathcal{I}$ [and|,|.]$\rangle$,
where $\mathcal{I}$ is a potential instance (e.g., *diderot*) and $\mathcal{C}$ is a potential class label (e.g., *writers*).

Following (Van Durme and Paşca, 2008), the boundaries of potential class labels $\mathcal{C}$ are approximated from the part-of-speech tags of the sentence words, whereas the boundaries of instances $\mathcal{I}$ are identified by checking that $\mathcal{I}$ occurs as an entire query in query logs. Since users type many queries in lower case, the collected data is converted to lower case.

When applied to inherently-noisy Web documents, the extraction patterns may produce irrelevant extractions (Kozareva et al., 2008). Causes of errors include incorrect detection of possible enumerations, as in *companies such as Procter and Gamble* (Downey et al., 2007); incorrect estimation of the boundaries of class labels, due to incorrect attachment as in *years* from *on a limited number of vehicles over the past few years, including the Chevrolet Corvette*; subjective (*famous actors*) (Hovy et al., 2009), relational (*competitors, nearby landmarks*) and otherwise less useful (*others*, *topics*) class labels; or questionable source sentences, as in *Large mammals such as deer and wild turkeys can be [..]* (Van Durme and Paşca, 2008).

As a solution, recent work uses additional evidence, as a means to filter the pairs extracted by patterns, thus trading off coverage for higher precision. The repository extracted from a similarly-sized Web document collection using the same initial extraction patterns as here, after a weighted intersection of pairs extracted with patterns and clusters of distributionally similar phrases, contains a total of 9,080 class labels associated with 263,000 instances in (Van Durme and Paşca, 2008). Subsequent extensions of the repository, using data derived from tables within Web documents, increase instance coverage and induce a ranking among class labels of each instance, but do not increase the number of class labels (Talukdar et al., 2008). Due to aggressive filtering, the resulting number of class labels is higher than the often-small sets of entity types studied previously, but may still be insufficient given the diversity of Web search queries.

2.2 Ranking of Classes per Instance

As an alternative, the soft ranking approach proposed here attempts to rank better class labels higher, without necessarily removing class labels deemed incorrect according to various criteria. For each instance $\mathcal{I}$, the associated class labels are ranked in the following stages:

1) Apply the scoring formula below, resulting in a ranked list of class labels $\mathrm{L}_1(\mathcal{I})$:

$$Score(\mathcal{I},\mathcal{C}) = Size(\{Pattern(\mathcal{I},\mathcal{C})\})^2 \times Freq(\mathcal{I},\mathcal{C})$$

Thus, a class label $\mathcal{C}$ is deemed more relevant for an instance $\mathcal{I}$ if $\mathcal{C}$ is extracted by multiple extraction patterns and its original frequency-based score is higher.

2) For each term within any class label from $\mathrm{L}_1(\mathcal{I})$, compute a score equal to the frequency sum of the term within anonymized queries containing the instance $\mathcal{I}$ as a prefix, and the term anywhere else in the queries. Each class label is assigned the geometric mean of the scores of its terms, after ignoring stop words. The class labels are ranked according to the means, resulting in a ranked list $\mathrm{L}_2(\mathcal{I})$. In case of ties, $\mathrm{L}_2(\mathcal{I})$ preserves the relative ranking from $\mathrm{L}_1(\mathcal{I})$. Thus, a class label is deemed more relevant if its individual terms occur in popular queries containing the instance.

3) Compute a merged ranked list of class labels out of the ranked lists $\mathrm{L}_1(\mathcal{I})$ and $\mathrm{L}_2(\mathcal{I})$, by sorting the class labels in decreasing order of the inverse of the average rank, computed with the following formula:

$$MergedScore(\mathcal{C}) = \frac{2}{Rank(\mathcal{C}, L_1) + Rank(\mathcal{C}, L_2)}$$

where 2 is the number of input lists of class labels, and Rank($\mathcal{C}$, L_i) is the rank of $\mathcal{C}$ in the list L_i of class labels computed for the corresponding input instance. The rank is set to 1000, if $\mathcal{C}$ is not present in the list L_i. By using only the relative ranks of the class labels within the input lists, and not on their scores, the outcome of the merging is less sensitive to how class labels of a given instance are scored within the IsA repository. In case of ties, the scores of the class labels from $\mathrm{L}_1(\mathcal{I})$ serve as a secondary ranking criterion.

Note that the third stage is introduced because relying on query logs to estimate the relevance of class labels exposes the ranking method to significant noise. On one hand, arguably useful class labels (e.g., *authors*) may not occur in queries along with the respective instances (*diderot*). On the other hand, for each query containing an instance and (part of) useful class labels, there are many other queries containing, e.g., attributes (*diderot biography* or *diderot beliefs*) or the name of a book in the query *diderot the nun*. Therefore, the ranked lists $\mathrm{L}_2(\mathcal{I})$ may be too noisy to be used directly as rankings of the class labels for $\mathcal{I}$.

3 Experimental Setting

3.1 Textual Data Sources

The acquisition of the IsA repository relies on unstructured text available within Web documents and search queries. The collection of queries is a sample of 50 million unique, fully-anonymized queries in English submitted by Web users in 2009. Each query is accompanied by its frequency of occurrence in the logs. The document collection consists of a sample of 100 million documents in English. The textual portion of the documents is cleaned of HTML, tokenized, split into sentences and part-of-speech tagged using the TnT tagger (Brants, 2000).

3.2 Experimental Runs

The experimental runs correspond to different methods for extracting and ranking pairs of an instance and a class:

- as available in the repository from (Talukdar et al., 2008), which is collected from a document collection similar in size to the one used here plus a collection of Web tables, in a run denoted R_g;
- from the repository extracted here, with class labels of an instance ranked based on the frequency and the number of extraction patterns (see $Score(\mathcal{I}, \mathcal{C})$ in Section 2), in run R_s;
- from the repository extracted here, with class labels of an instance ranked based on the $MergedScore$ from Section 2, in run R_u.

3.3 Evaluation Procedure

The manual evaluation of open-domain information extraction output is time consuming (Banko et al., 2007). Fortunately, it is possible to implement an automatic evaluation procedure for ranked lists of class labels, based on existing resources and systems. Assume that a gold standard is available, containing gold class labels that are each associated with a gold set of their instances. The creation of such gold standards is discussed later. Based on the gold standard, the ranked lists of class labels available within an IsA repository can be automatically evaluated as follows. First, for each gold label, the ranked lists of class labels of individual gold instances are retrieved from the IsA repository. Second, the individual retrieved lists are merged into a ranked list of class labels, associated with the gold label. The merged list is computed using an extension of the $MergedScore$ formula described earlier in Section 2. Third, the merged list is compared against the gold label, to estimate the accuracy of the merged list. Intuitively, a ranked list of class labels is a better approximation of a gold label, if class labels situated at better ranks in the list are closer in meaning to the gold label.

3.4 Evaluation Metric

Given a gold label and a list of class labels, if any, derived from the IsA repository, the rank of the highest class label that matches the gold label determines the score assigned to the gold label, in the form of the reciprocal rank, $\max(1/\mathrm{rank}_{match})$. Thus, if the gold label matches a class label at rank 1, 2, 3, 4 or 5 in the computed list, the gold label receives a score of 1, 0.5, 0.33, 0.25 or 0.2 respectively. The score is 0 if the gold label does not match any of the top 20 class labels. The overall score over the entire set of gold labels is the mean reciprocal rank (MRR) score over all gold labels from the set. Two types of MRR scores are automatically computed:

- MRR_f considers a gold label and a class label to match if they are identical;
- MRR_p considers a gold label and a class label to match if one or more of their tokens that are not stop words are identical.

During matching, all string comparisons are case-insensitive, and all tokens are first converted to their singular form (e.g., *european countries* to *european country*) when available, by using WordNet's morphological routines. Thus, *insurance carriers* and *insurance companies* are considered to not match in MRR$_f$ scores, but match in MRR$_p$ scores, whereas *insurance companies* and *insurance company* match in both MRR$_f$ and MRR$_p$ scores. Note that both MRR$_f$ and MRR$_p$ scores fail to give any credit to arguably valid and useful class labels, such as *insurers* for the gold label *insurance carriers*, or *asian nations* for the gold label *asia countries*. On the other hand, MRR$_p$ scores may give credit to less relevant class labels, such as *insurance policies* for the gold label *insurance carriers*. Therefore, MRR$_p$ is an approximate, and MRR$_f$ is a conservative, lower-bound estimate of the actual usefulness of the computed ranked lists of class labels as approximations of the semantics of the gold labels.

4 Evaluation Results

4.1 Evaluation Sets of Queries

A random sample of anonymized, class-seeking queries (e.g., *video game characters* or *smartphone*) submitted by Web users to Google Squared [1] over a 30-day interval is filtered, to remove queries for which Google Squared returns fewer than 10 instances at the time of the evaluation. The resulting evaluation set of queries, denoted Q$_e$, contains 807 queries, each associated with a ranked list of between 10 and 100 instances automatically extracted by Google Squared.

Since the instances available as input for each query as part of Q$_e$ are automatically extracted, they may (e.g., *acorn a7000*) or may not (e.g., *konrad zuse*) be true instances of the respective queries (e.g., *computers*). A second evaluation set Q$_m$ is assembled as a subset of 40 queries from Q$_e$, such that the instances available for each query in Q$_m$ are correct. For this purpose, each instance returned by Google Squared for the 40 queries from Q$_m$ is reviewed by at least three human annotators. Instances deemed highly relevant (out of 5 possible grades) with high inter-annotator agreement are retained. As a result, the 40 queries from Q$_m$ are associated with between 8 and 33 human-validated instances.

Query Set: Sample of Queries
Q$_e$ (807 queries): 2009 movies, amino acids, asian countries, bank, board games, buildings, capitals, chemical functional groups, clothes, computer language, dairy farms near modesto ca, disease, egyptian pharaohs, eu countries, french presidents, german islands, hawaiian islands, illegal drugs, irc clients, lakes, macintosh models, mobile operator india, nba players, nobel prize winners, orchids, photo editors, programming languages, renaissance artists, roller costers, science fiction tv series, slr cameras, soul singers, states of india, taliban members, thomas edison inventions, u.s. presidents, us president, water slides
Q$_m$ (40 queries): actors, airlines, birds, cars, celebrities, computer languages, digital camera, dog breeds, drugs, endangered animals, european countries, fruits, greek gods, horror movies, ipods, names, netbooks, operating systems, park slope restaurants, presidents, ps3 games, religions, renaissance artists, rock bands, universities, university, vitamins

Table 1: Size and composition of evaluation sets of queries associated with non-filtered (Q$_e$) or manually-filtered (Q$_m$) instances

Table 1 shows a sample of the queries from Q$_e$ and queries from Q$_m$. A small number of queries are slight lexical variations of one another, such as *u.s. presidents* and *us presidents* in Q$_e$, or *universities* and *university* in Q$_m$. In general, however, the sets cover a wide range of domains of interest, including entertainment for *2009 movies* and *rock bands*; biology for *endangered animals* and *amino acids*; geography for *asian countries* and *hawaiian islands*; food for *fruits*; history for *egyptian pharaohs* and *greek gods*; health for *drugs* and *vitamins*; and technology for *photo editors* and *ipods*. Some of the queries from Table 1 are specific enough that computing them exactly,

[1] Google Squared (http://www.google.com/squared) is a Web search tool taking as input class-seeking queries (e.g., *insurance companies*) and returning lists of instances (e.g., *allstate*, *state farm insurance*), along with attributes (e.g., *industry*, *headquarters*) and values for each instance.

	Accuracy											
I_Q	3			5			10			15		
C_I	5	10	20	5	10	20	5	10	20	5	10	20
MRR_f computed over Q_e:												
R_g	0.106	0.112	0.112	0.121	0.122	0.123	0.131	0.135	0.127	0.134	0.132	0.127
R_s	0.186	0.195	0.198	0.198	0.207	0.210	0.204	0.214	0.218	0.206	0.216	0.221
R_u	0.202	0.211	0.216	0.232	0.238	0.244	0.245	0.255	0.257	0.245	0.252	0.254
MRR_p computed over Q_e:												
R_g	0.390	0.399	0.394	0.420	0.420	0.413	0.443	0.443	0.435	0.439	0.431	0.425
R_s	0.489	0.495	0.495	0.517	0.528	0.529	0.541	0.553	0.557	0.551	0.557	0.557
R_u	0.520	0.531	0.533	0.564	0.573	0.578	0.590	0.601	0.602	0.598	0.603	0.601
MRR_f computed over Q_m:												
R_g	0.284	0.289	0.295	0.305	0.327	0.322	0.320	0.335	0.335	0.334	0.328	0.337
R_s	0.406	0.436	0.442	0.431	0.447	0.466	0.467	0.470	0.501	0.484	0.501	0.554
R_u	0.423	0.426	0.429	0.436	0.483	0.508	0.500	0.526	0.530	0.520	0.540	0.524
MRR_p computed over Q_m:												
R_g	0.507	0.517	0.531	0.495	0.509	0.518	0.555	0.553	0.550	0.563	0.561	0.572
R_s	0.667	0.662	0.660	0.675	0.677	0.699	0.702	0.695	0.716	0.756	0.765	0.787
R_u	0.711	0.703	0.680	0.734	0.731	0.748	0.733	0.797	0.782	0.799	0.834	0.819

Table 2: Accuracy of instance set labeling, as full-match (MRR_f) or partial-match (MRR_p) scores over the evaluation sets of queries associated with non-filtered instances (Q_e) or manually-filtered instances (Q_m), for various experimental runs (I_Q=number of instances available in the input evaluation sets that are used for retrieving class labels; C_I=number of class labels retrieved from IsA repository per input instance)

even from a comprehensive, perfect list of extracted instance, would be very difficult whether done automatically or manually. Examples of such queries are *dairy farms near modesto ca* and *science fiction tv series*, but also *mobile operator india* (phrase expressed as keywords) in Q_e, or *park slope restaurants* (specific location) in Q_m.

Access to a system such as Google Squared is useful, but not necessary to conduct the evaluation. Given other sets of queries, it is straightforward, albeit time consuming, to create evaluation sets similar to Q_m, by manually compiling correct instances, for each selected query or concept.

Following the general evaluation procedure, each query from the sets Q_e and Q_m acts as a gold class label associated with its set of instances. Given a query and its instances $\mathcal{I}$ from the evaluation sets Q_e or Q_m, we compute merged, ranked lists of class labels, by merging the ranked lists of class labels available in the underlying IsA repository for each instance $\mathcal{I}$. The evaluation compares the merged lists of class labels, on one hand, and the corresponding queries from Q_e or Q_m, on the other hand.

4.2 Accuracy of Class Labels

Table 2 summarizes results from comparative experiments, quantifying a) horizontally, the impact of alternative parameter settings on the computed lists of class labels; and b) vertically, the comparative accuracy of the experimental runs over the query sets. The experimental parameters are the number of input instances from the evaluation sets that are used for retrieving class labels, I_Q, set to 3, 5, 10 and 15; and the number of class labels retrieved per input instance, C_I, set to 5, 10 and 20.

The scores over Q_m are higher than those over Q_e, confirming the intuition that the higher-quality input set of instances available in Q_m relative to Q_e should lead to higher-quality class labels for the corresponding queries. When I_Q is fixed, increasing C_I leads to small, if any, score improvements. Conversely, when C_I is fixed, even small values of I_Q, such as 3 or 5 (that is, very small sets of instances provided as input) produce scores that are competitive with those obtained with a higher value like. This suggests that useful class labels can be generated even in extreme scenarios, where the number of instances available as input is as small as 3 or 5.

For most combinations of parameter settings and on both query sets, run R_u produces the highest scores. In particular, when I_Q is set to 10 and C_I to 20, run R_u identifies the original query as an exact match among the top four class labels returned; and as a partial match among the top two class labels returned, as an average over the Q_e set. In this case, the original query is identified at ranks 1, 2, 3, 4 and 5 for 16.8%, 8.7%, 6.1%, 3.7% and 1.7% of the queries, as an exact match; and for 48.8%, 14.2%, 6.1%, 3.6% and 1.9% respectively, as a partial match. The corresponding MRR_f score of 0.257 over the Q_e set obtained with run R_u is higher than with run R_s, and much higher than with run R_g. In all experiments, the higher scores of R_u can be attributed to higher coverage of class labels, relative to R_g; and higher-quality lists of class labels, relative to R_s but also to R_g, despite the fact that R_g combines high-precision seed data with using both unstructured and structured text as sources of class labels (cf. (Talukdar et al., 2008)). Among combinations of parameter settings described in Table 2, values around 15 for I_Q and 20 for C_I give the highest scores over both Q_e and Q_m.

5 Related Work

5.1 Extraction of IsA Repositories

Knowledge including instances and classes can be manually compiled by experts (Fellbaum, 1998) or collaboratively by non-experts (Singh et al., 2002). Alternatively, classes of instances acquired automatically from text are potentially less expensive to acquire, maintain and grow, and their coverage and scope are theoretically bound only by the size of the underlying data source. Existing methods for extracting classes of instances acquire sets of instances that are each either unlabeled (Wang and Cohen, 2008; Pennacchiotti and Pantel, 2009; Lin and Wu, 2009), or associated with a class label (Pantel and Pennacchiotti, 2006; Banko et al., 2007; Wang and Cohen, 2009). When associated with a class label, the sets of instances may be organized as flat sets or hierarchically, relative to existing hierarchies such as WordNet (Snow et al., 2006) or the category network within Wikipedia (Wu and Weld, 2008; Ponzetto and Navigli, 2009). Semi-structured text was shown to be a complementary resource to unstructured text, for the purpose of extracting relations from Web documents (Cafarella et al., 2008).

The role of anonymized query logs in Web-based information extraction has been explored in the tasks of class attribute extraction (Paşca and Van Durme, 2007) and instance set expansion (Pennacchiotti and Pantel, 2009). Our method illustrates the usefulness of queries considered in isolation from one another, in ranking class labels in extracted IsA repositories.

5.2 Labeling of Instance Sets

Previous work on generating relevant labels, given sets or clusters of items, focuses on scenarios where the items within the clusters are descriptions of, or full-length documents within document collections. The documents are available as a flat set (Cutting et al., 1993; Carmel et al., 2009) or are hierarchically organized (Treeratpituk and Callan, 2006). Relying on semi-structured content assembled manually as part of the structure of Wikipedia articles, such as article titles or categories, the method introduced in (Carmel et al., 2009) derives labels for clusters containing 100 full-length documents each. In contrast, our method relies on IsA relations automatically extracted from unstructured text within arbitrary Web documents, and computes labels given textual input that is orders of magnitude smaller, i.e., around 10 phrases (instances). The experiments described in (Carmel et al., 2009) assign labels to one of 20 sets of newsgroup documents from a standard benchmark. Each set of documents is associated with a higher-level, coarse-grained label used as a gold label against which the generated labels are compared. In comparison, our experiments compute text-derived class labels for finer-grained, often highly-specific gold labels.

Reducing the granularity of the items to be labeled from full documents to condensed document descriptions, (Geraci et al., 2006) submits arbitrary search queries to external Web search engines. It organizes the top 200 returned Web documents into clusters, by analyzing the text snippets associated with each document in the output from the search engines. Any words and phrases from the snippets may be selected as labels for the clusters, which in general leads to labels that are not intended to capture any classes that may be associated to the query. For example, labels of clusters generated in (Geraci et al., 2006) include *armstrong ceilings*, *italia*, *armstrong sul sito* and *louis jazz* for the query *armstrong*; and *madonnaweb*, *music*, *madonna online* and *madonna* itself for the query *madonna*. The amount of text available as input for the purpose of labeling is at least two orders of magnitude larger than in our method, and the task of selecting any phrases as labels, as opposed to selecting only labels that correspond to classes, is more relaxed and likely easier.

Another approach specifically addresses the problem of generating labels for sets of instances, where the labels are extracted from unstructured text. In (Pantel and Ravichandran, 2004), given a collection of news articles that is both cleaner and smaller than Web document collections, a syntactic parser is applied to document sentences in order to identify and exploit syntactic dependencies for the purpose of selecting candidate class labels. Such methods are comparatively less applicable to Web document collections, due to scalability issues associated with parsing a large set of Web documents of variable quality. Moreover, the class labels generated in (Pantel and Ravichandran, 2004) tend to be rather coarse-grained. For example, the top labels generated for a set of Chinese universities (*qinghua university*, *fudan university*, *beijing university*) are *university*, *institution*, *stock-holder*, *college* and *school*.

6 Conclusion

The method presented in this paper produces an IsA repository whose class labels have higher coverage and accuracy than with recent methods operating on document collections. This is done by injecting useful ranking signals from inherently-noisy queries, rather than making binary, coverage-reducing quality decisions on the extracted data. Current work investigates the usefulness of the extracted class labels in the generation of flat or hierarchical query refinements for class-seeking queries.

Acknowledgments

The author thanks Randolph Brown for assistance in assembling the evaluation sets of class-seeking queries.

References

Banko, M., 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.

Brants, T. 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.

Cafarella, M., 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.

Carmel, D., H. Roitman, and N. Zwerding. 2009. Enhancing cluster labeling using Wikipedia. In *Proceedings of the 32nd ACM Conference on Research and Development in Information Retrieval (SIGIR-09)*, pages 139–146, Boston, Massachusetts.

Cutting, D., D. Karger, and J. Pedersen. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In *Proceedings of the 16th ACM Conference on Research and Development in Information Retrieval (SIGIR-93)*, pages 126–134, Pittsburgh, Pennsylvania.

Downey, D., 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.

Etzioni, O., 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.

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

Geraci, F., M. Pellegrini, M. Maggini, and F. Sebastiani. 2006. Cluster generation and cluster labelling for Web snippets: A fast and accurate hierarchical solution. In *Proceedings of the 13th Conference on String Processing and Information Retrieval (SPIRE-06)*, pages 25–36, Glasgow, Scotland.

Hearst, M. 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.

Hovy, E., Z. Kozareva, and E. Riloff. 2009. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 948–957, Singapore.

Jansen, B., A. Spink, and T. Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the Web. *Information Processing and Management*, 36(2):207–227.

Kozareva, Z., 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.

Lin, D. 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.

Paşca, M. and B. Van Durme. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837, Hyderabad, India.

Pantel, P. 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.

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

Pennacchiotti, M. 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.

Ponzetto, S. 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.

Ponzetto, S. 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.

Singh, P., T. Lin, E. Mueller, G. Lim, T. Perkins, and W. Zhu. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. In *Proceedings of the ODBASE Conference (ODBASE-02)*, pages 1223–1237.

Snow, R., 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.

Talukdar, P., 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.

Treeratpituk, P. and J. Callan. 2006. Automatically labeling hierarchical clusters. In *Proceedings of the 7th Annual Conference on Digital Government Research (DGO-06)*, pages 167–176, San Diego, California.

Van Durme, B. 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.

Wang, R. and W. Cohen. 2008. Iterative set expansion of named entities using the web. In *Proceedings of the International Conference on Data Mining (ICDM-08)*, pages 1091–1096, Pisa, Italy.

Wang, R. 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.

Wu, F. 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.