Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks

Partha Pratim Talukdar*
University of Pennsylvania
Philadelphia, PA 19104
partha@cis.upenn.edu

Joseph Reisinger*
University of Texas at Austin
Austin, TX 78712
joeraii@cs.utexas.edu

Marius Paşca
Google Inc.
Mountain View, CA 94043
mars@google.com

Deepak Ravichandran
Google Inc.
Mountain View, CA 94043
deepakr@google.com

Rahul Bhagat*
USC Information Sciences Institute
Marina Del Rey, CA 90292
rahul@isi.edu

Fernando Pereira
Google Inc.
Mountain View, CA 94043
pereira@google.com

Abstract

We present a graph-based semi-supervised label propagation algorithm for acquiring open-domain labeled classes and their instances from a combination of unstructured and structured text sources. This acquisition method significantly improves coverage compared to a previous set of labeled classes and instances derived from free text, while achieving comparable precision.

1 Introduction

1.1 Motivation

Users of large document collections can readily acquire information about the instances, classes, and relationships described in the documents. Such relations play an important role in both natural language understanding and Web search, as illustrated by their prominence in both Web documents and among the search queries submitted most frequently by Web users (Jansen et al., 2000). These observations motivate our work on algorithms to extract instance-class information from Web documents.

While work on named-entity recognition traditionally focuses on the acquisition and identification of instances within a small set of coarse-grained classes, the distribution of instances within query logs indicates that Web search users are interested in a wider range of more fine-grained classes. Depending on prior knowledge, personal interests and immediate needs, users submit for example medical queries about the symptoms of *leptospirosis* or the treatment of *monkeypox*, both of which are instances of *zoonotic diseases*, or the risks and benefits of *surgical procedures* such as *PRK* and *angioplasty*. Other users may be more interested in *African countries* such as *Uganda* and *Angola*, or *active volcanoes* like *Etna* and *Kilauea*. Note that *zoonotic diseases*, *surgical procedures*, *African countries* and *active volcanoes* serve as useful class labels that capture the semantics of the associated sets of class instances. Such interest in a wide variety of specific domains highlights the utility of constructing large collections of fine-grained classes.

Comprehensive and accurate class-instance information is useful not only in search but also in a variety of other text processing tasks including co-reference resolution (McCarthy and Lehnert, 1995), named entity recognition (Stevenson and Gaizauskas, 2000) and seed-based information extraction (Riloff and Jones, 1999).

1.2 Contributions

We study the acquisition of open-domain, labeled classes and their instances from both structured and unstructured textual data sources by combining and ranking individual extractions in a principled way with the Adsorption label-propagation algorithm (Baluja et al., 2008), reviewed in Section 3 below.

A collection of labeled classes acquired from text (Van Durme and Paşca, 2008) is extended in two ways:

1. Class label coverage is increased by identifying additional class labels (such as *public agencies* and *governmental agencies*) for existing

*Contributions made during internships at Google.

instances such as *Office of War Information*),

2. The overall instance coverage is increased by extracting additional instances (such as *Addison Wesley* and *Zebra Books*) for existing class labels (*book publishers*).

The WebTables database constructed by Cafarella et al. (2008) is used as the source of additional instances. Evaluations on gold-standard labeled classes and instances from existing linguistic resources (Fellbaum, 1998) indicate coverage improvements relative to that of Van Durme and Paşca (2008), while retaining similar precision levels.

2 First Phase Extractors

To show Adsorption's ability to uniformly combine extractions from multiple sources and methods, we apply it to: 1) high-precision open-domain extractions from free Web text (Van Durme and Paşca, 2008), and 2) high-recall extractions from WebTables, a large database of HTML tables mined from the Web (Cafarella et al., 2008). These two methods were chosen to be representative of two broad classes of extraction sources: free text and structured Web documents.

2.1 Extraction from Free Text

Van Durme and Paşca (2008) produce an open-domain set of instance clusters $C \in \mathcal{C}$ that partitions a given set of instances $\mathcal{I}$ using distributional similarity (Lin and Pantel, 2002), and labels using *is-a* patterns (Hearst, 1992). By filtering the class labels using distributional similarity, a large number of high-precision labeled clusters are extracted. The algorithm proceeds iteratively: at each step, all clusters are tested for label *coherence* and all coherent labels are tested for high cluster *specificity*. Label L is coherent if it is shared by at least $J\%$ of the instances in cluster C, and it is specific if the total number of other clusters $C' \in \mathcal{C}, C' \neq C$ containing instances with label L is less than K. When a cluster is found to match these criteria, it is removed from $\mathcal{C}$ and added to an output set. The procedure terminates when no new clusters can be removed from $\mathcal{C}$. Table 1 shows a few randomly chosen classes and representative instances obtained by this procedure.

2.2 Extraction from Structured Text

To expand the instance sets extracted from free text, we use a *table-based extraction* method that mines structured Web data in the form of HTML tables. A significant fraction of the HTML tables in Web pages is assumed to contain coherent lists of instances suitable for extraction. Identifying such tables from scratch is hard, but seed instance lists can be used to identify potentially coherent table columns. In this paper we use the WebTables database of around 154 million tables as our structured data source (Cafarella et al., 2008).

We employ a simple ranking scheme for candidate instances in the WebTables corpus $\mathcal{T}$. Each table $\mathbf{T} \in \mathcal{T}$ consists of one or more columns. Each column $g \in \mathbf{T}$ consists of a set of candidate instances $i \in g$ corresponding to row elements. We define the set of unique *seed matches* in g relative to semantic class $C \in \mathcal{C}$ as

$$M_C(g) \stackrel{\text{def}}{=} \{i \in I(C) : i \in g\}$$

where $I(C)$ denotes the set of instances in seed class C. For each column g, we define its α*-unique class coverage*, that is, the set of classes that have at least α unique seeds in g,

$$Q(g;\alpha) \stackrel{\text{def}}{=} \{C \in \mathcal{C} : |M_C(g)| \geq \alpha\}.$$

Using M and Q we define a method for scoring columns relative to each class. Intuitively, such a score should take into account not only the number of matches from class C, but also the total number of classes that contribute to Q and their relative overlap. Towards this end, we introduce the scoring function

$$score(C,g;\alpha) \stackrel{\text{def}}{=} \underbrace{|M_C(g)|}_{\text{seed matches}} \cdot \overbrace{\frac{|M_C(g)|}{|\bigcup_{C' \in Q(g;\alpha)} I(C')|}}^{\text{class coherence}}$$

which is the simplest scoring function combining the number of seed matches with the coherence of the table column. Coherence is a critical notion in WebTables extraction, as some tables contain instances across many diverse seed classes, contributing to extraction noise. The class coherence introduced here also takes into account class overlap; that

Class	Size	Examples of Instances
Book Publishers	70	crown publishing, kluwer academic, prentice hall, puffin
Federal Agencies	161	catsa, dhs, dod, ex-im bank, fsis, iema, mema, nipc, nmfs, tdh, usdot
Mammals	956	armadillo, elephant shrews, long-tailed weasel, river otter, weddell seals, wild goat
NFL Players	180	aikman, deion sanders, fred taylor, jamal lewis, raghib ismail, troy vincent
Scientific Journals	265	biometrika, european economic review, nature genetics, neuroscience
Social Issues	210	gender inequality, lack of education, substandard housing, welfare dependency
Writers	5089	bronte sisters, hemingway, kipling, proust, torquato tasso, ungaretti, yeats

Table 1: A sample of the open-domain classes and associated instances from (Van Durme and Paşca, 2008).

is, a column containing many semantically similar classes is penalized less than one containing diverse classes.[1] Finally, an extracted instance i is assigned a score relative to class C equal to the sum of all its column scores,

$$score(i, C; \alpha) \stackrel{\text{def}}{=} \frac{1}{Z_C} \sum_{g \in \mathbf{T}, \mathbf{T} \in \mathcal{T}} score(C, g; \alpha)$$

where Z_C is a normalizing constant set to the maximum score of any instance in class C. This scoring function assigns high rank to instances that occur frequently in columns with many seed matches and high class specificity.

The ranked list of extracted instances is post-filtered by removing all instances that occur in less than d unique Internet domains.

3 Graph-Based Extraction

To combine the extractions from both free and structured text, we need a representation capable of encoding efficiently all the available information. We chose a graph representation for the following reasons:

- Graphs can represent complicated relationships between classes and instances. For example, an ambiguous instance such as *Michael Jordan* could belong to the class of both *Professors* and *NBA players*. Similarly, an instance may belong to multiple nodes in the hierarchy of classes. For example, *Blue Whales* could belong to both classes *Vertebrates* and *Mammals*, because *Mammals* are a subset of *Vertebrates*.
- Extractions from multiple sources, such as Web queries, Web tables, and text patterns can be represented in a single graph.
- Graphs make explicit the potential paths of information propagation that are implicit in the more common local heuristics used for weakly-supervised information extraction. For example, if we know that the instance *Bill Clinton* belongs to both classes *President* and *Politician* then this should be treated as evidence that the class of *President* and *Politician* are related.

Each instance-class pair (i, C) extracted in the first phase (Section 2) is represented as a weighted edge in a graph $G = (V, E, W)$, where V is the set of nodes, E is the set of edges and $W : E \rightarrow \mathbb{R}^+$ is the weight function which assigns positive weight to each edge. In particular, for each (i, C, w) triple from the set of base extractions, i and C are added to V and (i, C) is added to E, [2] with $W(i, C) = w$. The weight w represents the total score of all extractions with that instance and class. Figure 1 illustrates a portion of a sample graph. This simple graph representation could be refined with additional types of nodes and edges, as we discuss in Section 7.

In what follows, all nodes are treated in the same way, regardless of whether they represent instances or classes. In particular, all nodes can be assigned class labels. For an instance node, that means that the instance is hypothesized to belong to the class; for a class node, that means that the node's class is hypothesized to be semantically similar to the label's class (Section 5).

We now formulate the task of assigning labels to nodes as graph label propagation. We are given a

[1] Note that this scoring function does not take into account class containment: if all seeds are both *wind Instruments* and *instruments*, then the column should assign higher score to the more specific class.

[2] In practice, we use two directed edges, from i to C and from C to i, both with weight w.

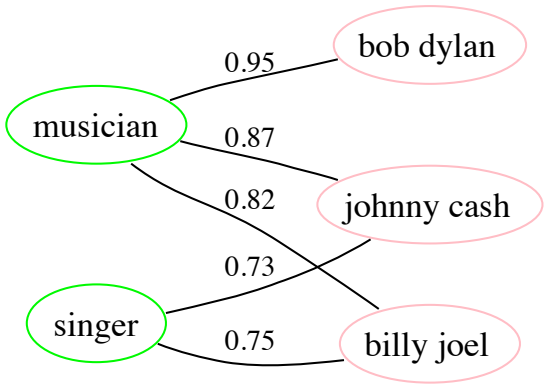

Figure 1: Section of a graph used as input into Adsorption. Though the nodes do not have any type associated with them, for readability, instance nodes are marked in pink while class nodes are shown in green.

set of instances $\mathcal{I}$ and a set of classes $\mathcal{C}$ represented as nodes in the graph, with connecting edges as described above. We annotate a few instance nodes with labels drawn from $\mathcal{C}$. That is, classes are used both as nodes in the graph and as labels for nodes. There is no necessary alignment between a class node and any of the (class) labels, as the final labels will be assigned by the Adsorption algorithm.

The Adsorption label propagation algorithm (Baluja et al., 2008) is now applied to the given graph. Adsorption is a general framework for label propagation, consisting of a few nodes annotated with labels and a rich graph structure containing the universe of all labeled and unlabeled nodes. Adsorption proceeds to label all nodes based on the graph structure, ultimately producing a probability distribution over labels for each node.

More specifically, Adsorption works on a graph $G = (V, E, W)$ and computes for each node v a *label distribution* L_v that represents which labels are more or less appropriate for that node. Several interpretations of Adsorption-type algorithms have appeared in various fields (Azran, 2007; Zhu et al., 2003; Szummer and Jaakkola, 2002; Indyk and Matousek, 2004). For details, the reader is referred to (Baluja et al., 2008). We use two interpretations here:

Adsorption through Random Walks: Let $G_r = (V, E_r, W_r)$ be the edge-reversed version of the original graph $G = (V, E, W)$ where $(a, b) \in E_r$ iff $(b, a) \in E$; and $W_r(a, b) = W(b, a)$. Now, choose a node of interest $q \in V$. To estimate L_q for q, we perform a random walk on G_r starting from q to generate values for a random label variable L. After reaching a node v during the walk, we have three choices:

1. With probability p_v^{cont}, continue the random walk to a neighbor of v.
2. With probability p_v^{abnd}, abandon the random walk. This abandonment probability makes the random walk stay relatively close to its source when the graph has high-degree nodes. When the random walk passes through such a node, it is likely that further transitions will be into regions of the graph unrelated to the source. The abandonment probability mitigates that effect.
3. With probability p_v^{inj}, stop the random walk and emit a label L from I_v.

L_q is set to the expectation of all labels L emitted from random walks initiated from node q.

Adsorption through Averaging: For this interpretation we make some changes to the original graph structure and label set. We extend the label distributions L_v to assign a probability not only to each label in $\mathcal{C}$ but also to the *dummy* label $\perp$, which represents lack of information about the actual label(s). We represent the initial knowledge we have about some node labels in an *augmented* graph $G' = (V', E', W')$ as follows. For each $v \in V$, we define an *initial* distribution $I_v = L^{\perp}$, where $L^{\perp}$ is the *dummy* distribution with $L^{\perp}(\perp) = 1$, representing lack of label information for v. In addition, let $V_s \subseteq V$ be the set of nodes for which we have some actual label knowledge, and let $V' = V \cup \{\bar{v} : v \in V_s\}, E' = E \cup \{(\bar{v}, v) : v \in V_s\}$, and $W'(\bar{v}, v) = 1$ for $v \in V_s$, $W'(u, v) = W(u, v)$ for $u, v \in V$. Finally, let $I_{\bar{v}}$ (seed labels) specify the knowledge about possible labels for $v \in V_s$. Less formally, the $\bar{v}$ nodes in G' serve to *inject* into the graph the prior label distributions for each $v \in V_s$.

The algorithm proceeds as follows: For each node use a fixed-point computation to find label

distributions that are weighted averages of the label distributions for all their neighbors. This causes the non-dummy initial distribution of V_s nodes to be propagated across the graph.

Baluja et al. (2008) show that those two views are equivalent. Algorithm 1 combines the two views: instead of a random walk, for each node v, it iteratively computes the weighted average of label distributions from neighboring nodes, and then uses the random walk probabilities to estimate a new label distribution for v.

For the experiments reported in Section 4, we used the following heuristics from Baluja et al. (2008) to set the random walk probabilities:

- Let $c_v = \frac{\log \beta}{\log(\beta + \exp H(v))}$ where $H(v) = -\sum_u p_{uv} \times \log(p_{uv})$ with $p_{uv} = \frac{W(u,v)}{\sum_{u'} W(u',v)}$. $H(v)$ can be interpreted as the entropy of v's neighborhood. Thus, c_v is lower if v has many neighbors. We set $\beta = 2$.

- $j_v = (1 - c_v) \times \sqrt{H(v)}$ if $I_v \neq L^\top$ and 0 otherwise.

- Then let

$$\begin{aligned} z_v &= \max(c_v + j_v, 1) \\ p_v^{cont} &= c_v / z_v \\ p_v^{inj} &= j_v / z_v \\ p_v^{abnd} &= 1 - p_v^{cont} - p_v^{abnd} \end{aligned}$$

Thus, abandonment occurs only when the continuation and injection probabilities are low enough.

The algorithm is run until convergence which is achieved when the label distribution on each node ceases to change within some tolerance value. Alternatively, the algorithm can be run for a fixed number of iterations which is what we used in practice[3].

Finally, since Adsorption is memoryless, it easily scales to tens of millions of nodes with dense edges and can be easily parallelized, as described by Baluja et al. (2008).

[3]The number of iterations was set to 10 in the experiments reported in this paper.

Algorithm 1 Adsorption Algorithm.
Input: $G' = (V', E', W')$, I_v $(\forall v \in V')$.
Output: Distributions $\{L_v : v \in V\}$.

1: $L_v = I_v \; \forall v \in V'$
2:
3: **repeat**
4: $\quad N_v = \sum_u W(u, v)$
5: $\quad D_v = \frac{1}{N_v} \sum_u W(u, v) L_u \;\; \forall v \in V'$
6: $\quad$ **for all** $v \in V'$ **do**
7: $\quad\quad L_v = p_v^{cont} \times D_v + p_v^{inj} \times I_v + p_v^{abnd} \times L^\top$
8: $\quad$ **end for**
9: **until** convergence

4 Experiments

4.1 Data

As mentioned in Section 3, one of the benefits of using Adsorption is that we can combine extractions by different methods from diverse sources into a single framework. To demonstrate this capability, we combine extractions from free-text patterns and from Web tables. To the best of our knowledge, this is one of the first attempts in the area of minimally-supervised extraction algorithms where unstructured and structured text are used in a principled way within a single system.

Open-domain (instance, class) pairs were extracted by applying the method described by Van Durme and Paşca (2008) on a corpus of over 100M English web documents. A total of 924K (instance, class) pairs were extracted, containing 263K unique instances in 9081 classes. We refer to this dataset as A8.

Using A8, an additional 74M unique (instance,class) pairs are extracted from a random 10% of the WebTables data, using the method outlined in Section 2.2. For maximum coverage we set $\alpha = 2$ and $d = 2$, resulting in a large, but somewhat noisy collection. We refer to this data set as WT.

4.2 Graph Creation

We applied the graph construction scheme described in Section 3 on the A8 and WT data combined, resulting in a graph with 1.4M nodes and 75M edges. Since extractions in A8 are not scored, weight of all

Seed Class	Seed Instances
Book Publishers	millbrook press, academic press, springer verlag, chronicle books, shambhala publications
Federal Agencies	dod, nsf, office of war information, tsa, fema
Mammals	african wild dog, hyaena, hippopotamus, sperm whale, tiger
NFL Players	ike hilliard, isaac bruce, torry holt, jon kitna, jamal lewis
Scientific Journals	american journal of roentgenology, pnas, journal of bacteriology, american economic review, ibm systems journal

Table 2: Classes and seeds used to initialize Adsorption.

edges originating from A8 were set at 1[4]. This graph is used in all subsequent experiments.

5 Evaluation

We evaluated the Adsorption algorithm under two experimental settings. First, we evaluate Adsorption's extraction precision on (instance, class) pairs obtained by Adsorption but not present in A8 (Section 5.1). This measures whether Adsorption can add to the A8 extractions at fairly high precision. Second, we measured Adsorption's ability to assign labels to a fixed set of gold instances drawn from various classes (Section 5.2).

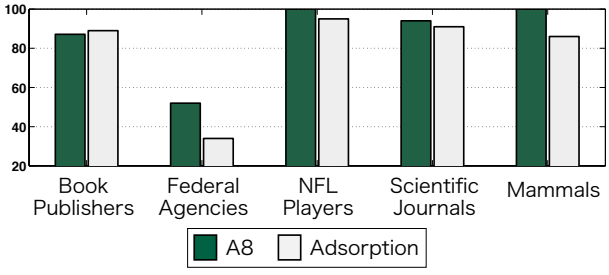

Figure 2: Precision at 100 comparisons for A8 and Adsorption.

5.1 Instance Precision

First we manually evaluated precision across five randomly selected classes from A8: Book Publishers, Federal Agencies, NFL Players, Scientific Journals and Mammals. For each class, 5 seed instances were chosen manually to initialize Adsorption. These classes and seeds are shown in Table 2. Adsorption was run for each class separately and the resulting ranked extractions were manually evaluated.

Since the A8 system does not produce ranked lists of instances, we chose 100 random instances from the A8 results to compare to the top 100 instances produced by Adsorption. Each of the resulting 500 instance-class pairs (i, C) was presented to two human evaluators, who were asked to evaluate whether the relation "i is a C" was correct or incorrect. The user was also presented with Web search link to verify the results against actual documents. Results from these experiments are presented in Figure 2 and Table 4. The results in Figure 2 show that the A8 system has higher precision than the Adsorption system. This is not surprising since the A8 system is tuned for high precision. When considering individual evaluation classes, changes in precision scores between the A8 system and the Adsorption system vary from a small increase from 87% to 89% for the class Book Publishers, to a significant decrease from 52% to 34% for the class Federal Agencies, with a decrease of 10% as an average over the 5 evaluation classes.

Class	Precision at 100 (non-A8 extractions)
Book Publishers	87.36
Federal Agencies	29.89
NFL Players	94.95
Scientific Journals	90.82
Mammal Species	84.27

Table 4: Precision of top 100 Adsorption extractions (for five classes) which were **not** present in A8.

Table 4 shows the precision of the Adsorption system for instances not extracted by the A8 system.

[4] A8 extractions are assumed to be high-precision and hence we assign them the highest possible weight.

Seed Class	Non-Seed Class Labels Discovered by Adsorption
Book Publishers	small presses, journal publishers, educational publishers, academic publishers, commercial publishers
Federal Agencies	public agencies, governmental agencies, modulation schemes, private sources, technical societies
NFL Players	sports figures, football greats, football players, backs, quarterbacks
Scientific Journals	prestigious journals, peer-reviewed journals, refereed journals, scholarly journals, academic journals
Mammal Species	marine mammal species, whale species, larger mammals, common animals, sea mammals

Table 3: Top class labels ranked by their similarity to a given seed class in Adsorption.

Seed Class	Sample of Top Ranked Instances Discovered by Adsorption
Book Publishers	small night shade books, house of anansi press, highwater books, distributed art publishers, copper canyon press
NFL Players	tony gonzales, thabiti davis, taylor stubblefield, ron dixon, rodney hannah
Scientific Journals	journal of physics, nature structural and molecular biology, sciences sociales et santé, kidney and blood pressure research, american journal of physiology–cell physiology

Table 5: Random examples of top ranked extractions (for three classes) found by Adsorption which were not present in A8.

Such an evaluation is important as one of the main motivations of the current work is to increase coverage (recall) of existing high-precision extractors without significantly affecting precision. Results in Table 4 show that Adsorption is indeed able to extraction with high precision (in 4 out of 5 cases) new instance-class pairs which were not extracted by the original high-precision extraction set (in this case A8). Examples of a few such pairs are shown in Table 5. This is promising as almost all state-of-the-art extraction methods are high-precision and low-recall. The proposed method shows a way to overcome that limitation.

As noted in Section 3, Adsorption ignores node type and hence the final ranked extraction may also contain classes along with instances. Thus, in addition to finding new instances for classes, it also finds additional class labels similar to the seed class labels with which Adsorption was run, at no extra cost. Some of the top ranked class labels extracted by Adsorption for the corresponding seed class labels are shown in Table 3. To the best of our knowledge, there are no other systems which perform both tasks simultaneously.

5.2 Class Label Recall

Next we evaluated each extraction method on its relative ability to assign labels to class instances. For each test instance, the five most probably class labels are collected using each method and the Mean Reciprocal Rank (MRR) is computed relative to a gold standard target set. This target set, WN-gold, consists of the 38 classes in Wordnet containing 100 or more instances.

In order to extract meaningful output from Adsorption, it is provided with a number of labeled seed instances (1, 5, 10 or 25) from each of the 38 test classes. Regardless of the actual number of seeds used as input, all 25 seed instances from each class are removed from the output set from all methods, in order to ensure fair comparison.

The results from this evaluation are summarized in Table 6; AD x refers to the adsorption run with x seed instances. Overall, Adsorption exhibits higher MRR than either of the baseline methods, with MRR increasing as the amount of supervision is increased. Due to its high coverage, WT assigns labels to a larger number of the instance in WN-gold than any other method. However, the average rank of the correct class assignment is lower, resulting is

Method	MRR (full)	MRR (found only)	# found
A8	0.16	0.47	2718
WT	0.15	0.21	**5747**
AD 1	0.26	0.45	4687
AD 5	0.29	0.48	4687
AD 10	0.30	0.51	4687
AD 25	**0.32**	**0.55**	4687

Table 6: Mean-Reciprocal Rank scores of instance class labels over 38 Wordnet classes (WN-gold). MRR (full) refers to evaluation across the entire gold instance set. MRR (found only) computes MRR only on recalled instances.

lower MRR scores compared to Adsorption. This result highlights Adsorption's ability to effectively combine high-precision, low-recall (A8) extractions with low-precision, high-recall extractions (WT) in a manner that improves *both* precision and coverage.

6 Related Work

Graph based algorithms for minimally supervised information extraction methods have recently been proposed. For example, Wang and Cohen (2007) use a random walk on a graph built from entities and relations extracted from semi-structured text. Our work differs both conceptually, in terms of its focus on open-domain extraction, as well as methodologically, as we incorporate both unstructured and structured text. The re-ranking algorithm of Bellare et al. (2007) also constructs a graph whose nodes are instances and attributes, as opposed to instances and classes here. Adsorption can be seen as a generalization of the method proposed in that paper.

7 Conclusion

The field of open-domain information extraction has been driven by the growth of Web-accessible data. We have staggering amounts of data from various structured and unstructured sources such as general Web text, online encyclopedias, query logs, web tables, or link anchor texts. Any proposed algorithm to extract information needs to harness several data sources and do it in a robust and scalable manner. Our work in this paper represents a first step towards that goal. In doing so, we achieved the following:

1. Improved coverage relative to a high accuracy instance-class extraction system while maintaining adequate precision.

2. Combined information from two different sources: free text and web tables.

3. Demonstrated a graph-based label propagation algorithm that given as little as five seeds per class achieved good results on a graph with more than a million nodes and 70 million edges.

In this paper, we started off with a simple graph. For future work, we plan to proceed along the following lines:

1. Encode richer relationships between nodes, for example instance-instance associations and other types of nodes.

2. Combine information from more data sources to answer the question of whether more data or diverse sources are more effective in increasing precision and coverage.

3. Apply similar ideas to other information extraction tasks such as relation extraction.

Acknowledgments

We would like to thank D. Sivakumar for useful discussions and the anonymous reviewers for helpful comments.

References

A. Azran. 2007. The rendezvous algorithm: multiclass semi-supervised learning with markov random walks. *Proceedings of the 24th international conference on Machine learning*, pages 49–56.

S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph.

K. Bellare, P. Talukdar, G. Kumaran, F. Pereira, M. Liberman, A. McCallum, and M. Dredze. 2007. Lightly-Supervised Attribute Extraction. *NIPS 2007 Workshop on Machine Learning for Web Search.*

M. Cafarella, A. Halevy, D. Wang, E. Wu, and Y. Zhang. 2008. Webtables: Exploring the power of tables on the web. *VLDB*.

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

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.

P. Indyk and J. Matousek. 2004. Low-distortion embeddings of finite metric spaces. *Handbook of Discrete and Computational Geometry*.

B. Jansen, 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.

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.

K. McCarthy and W. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1050–1055, Montreal, Quebec.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)*, pages 474–479, Orlando, Florida.

M. Stevenson and R. Gaizauskas. 2000. Using corpus-derived name lists for named entity recognition. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, Seattle, Washington.

M. Szummer and T. Jaakkola. 2002. Partially labeled classification with markov random walks. *Advances in Neural Information Processing Systems 14: Proceedings of the 2002 NIPS Conference*.

B. Van Durme and M. Paşca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. *Twenty-Third AAAI Conference on Artificial Intelligence*.

R. Wang and W. Cohen. 2007. Language-Independent Set Expansion of Named Entities Using the Web. *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 342–350.

X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. *ICML-03, 20th International Conference on Machine Learning*.