# Bootstrapped Extraction of Class Attributes

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

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

## ABSTRACT

As an alternative to previous studies on extracting class attributes from unstructured text, which consider either Web documents or query logs as the source of textual data, A bootstrapped method extracts class attributes simultaneously from both sources, using a small set of seed attributes. The method improves extraction precision and also improves attribute relevance across 40 test classes.

## Categories and Subject Descriptors

I.2.7 [**Computing Methodologies**]: Natural Language Processing

## General Terms

Algorithms, Experimentation

## 1. EXTRACTION OF ATTRIBUTES

**Motivation:** Class attributes capture quantifiable properties (e.g., *hiking trails*, *entrance fee* and *elevation*), of given classes of instances (e.g., *NationalPark*), and thus potentially serve as a skeleton towards constructing large-scale knowledge bases automatically. Previous work on extracting class attributes from unstructured text consider either Web documents [5] or query logs [2] as the extraction source. In this poster, we develop *Bootstrapped Web Search* (BWS), a method for combining Web documents and query logs as textual data sources that may contain class attributes. Web documents have textual content of higher semantic quality, convey information directly in natural language rather than through sets of keywords, and contain more raw textual data. In contrast, search queries are usually ambiguous, short, keyword-based approximations of often-underspecified user information needs. Previous work has shown, however, that extraction from query logs yields significantly higher precision than extraction from Web documents [2].

BWS is a generic method for multiple-source class attribute extraction that allows for corpora with varying levels of extraction precision to be combined favorably. It requires no supervision other than a small set of seed attributes for each semantic class. We test this method by combining query logs and Web documents, leveraging their strengths in order to improve coverage and precision.

**Combining Multiple Data Sources:** Significant previous work has been done on attribute extraction across a wide variety of data sources, e.g. news reports, query logs and Web documents. If extraction from such domains yields high precision results, intuitively it should be possible to obtain even more accurate attributes while lowering bias by using a combination of data sources.

*Contributions made during an internship at Google.


*WWW 2009,* April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

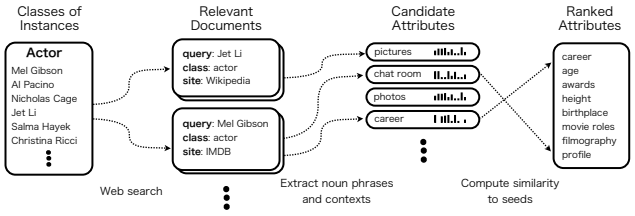


**Figure 1: Overview of the Web-search extraction procedure**

We consider the general problem of extracting and ranking a set of candidate attributes $\mathcal{A}$ using a ranked list of noisy seed attributes $\mathcal{S}$ extracted from a different source. The only assumption placed on $\mathcal{S}$ is that attribute precision correlates with rank on average.

In order to leverage noisy supervision, we introduce a simple one-parameter smoothing procedure that builds on the assumption that a seed's rank is proportional to its precision. Candidate attributes are ranked higher when they are more similar to a large number of the most precise seed attributes. Intuitively, a candidate attribute $a \in \mathcal{A}$ should be ranked highly overall if it has a high average similarity to the most precise seed attributes.

Given $\mathcal{A}$, the unordered set of extracted candidate attributes, and $\mathcal{S}$, the set of (noisy) ranked seed attributes extracted from a different source, each attribute $a \in \mathcal{A}$ is assigned a class-specific *extraction profile*, $\rho_a(s) \stackrel{\text{def}}{=} s^{-1} \sum_{i=0}^{s} \text{Sim}(a, \mathcal{S}_i)$, equal to its average similarity function Sim over the top $s < |\mathcal{S}|$ total seeds. Given a set of candidate attributes and their associated extraction profiles, a ranked list $\mathcal{A}_{\text{ranked}}$ is constructed incrementally. At each step $s < |\mathcal{S}|$, the top $\alpha$ attributes ranked by their similarity to the top $s$ seeds, $\rho_a(s)$, are added to $\mathcal{A}_{\text{ranked}}$, and removed from $\mathcal{A}$.

Each evaluation of $\rho_a(s)$ requires $O(s)$ similarity calculations, and hence $O(|\mathcal{A}||\mathcal{S}|^2)$ operations would be required to rank all candidate attributes. For efficiency we calculate $\rho_a$ at a discrete set of *seed levels*, $\delta = [5, 10, 20, 40, 60, 100]$.$\mathcal{A}_{\text{ranked}}$ is then constructed recursively, and at each step the top $(\delta_s - \delta_{s-1})/\alpha$ attributes ranked by $\rho(\delta_s)$ are added to $\mathcal{A}_{\text{ranked}}$ (skipping duplicates). The parameter $\alpha$ controls how strongly attribute ranking should prefer to emphasize the number of seeds over the list rank; as $\alpha \to \infty$, more seeds are used on average to score each attribute. Intuitively, this method works because each seed level has a different peak where the best results are obtained. By adjusting $\alpha$, we can construct a list of attributes with high precision across all ranks.

**Extraction from Relevant Web Documents:** As a specific application of combining multiple textual data sources for attribute extraction, we use attributes extracted from query logs in [3] as noisy supervision for finding attributes in *relevant* Web documents. Web documents relevant to a particular class are found by performing search queries for each instance of that class and collecting the top

documents returned. This approach is novel and is hypothesized to yield better attributes than untargeted documents precisely because of the increase in relevancy. There are three main phases: 1) document acquisition and noun-phrase extraction, 2) context vector generation from extracted usage patterns and 3) attribute ranking using distributional similarity. Figure 1 summarizes the approach taken, using the class *Actor* as an example.

## 2. METHODS

**WD:** (Web Document based extraction) – This method uses a fixed set of linguistically motivated surface patterns (e.g. "the $A$ of $I$" or "$I$'s $A$" for instance $I$ and attribute $A$) combined with a pre-specified set of instance classes to extract attributes from a Web-based textual corpus [3].

**QP:** (Query logs using Patterns) – This method uses the same procedure as WD, but is applied to query logs instead of Web documents, yielding higher precision in practice [2].

**QL:** (Query Logs using seeds) – A set of 5 manually specified *seed* attributes for each class are used to automatically extract patterns (syntactic contexts) that contain a seed and an instance from the same class. These patterns are then used to find other candidate attributes, i.e. non-seed noun-phrases that match the extracted patterns. These candidate attributes are then ranked based on their similarity across all contextual patterns. This method produces significantly more precise attributes than QP [3].

**WS**: (Web Search extraction) – The first novel method proposed in this poster; attributes are extracted from relevant documents returned from search queries.

**BWS:** (Bootstrapped Web Search extraction) – The second method proposed in this poster; it uses the top 100 high-precision attributes extracted with QL as additional supervision for WS.

## 3. EVALUATION

**Classes of Instances:** Extraction specificity is controlled via a set of *instance clusters* (corresponding to semantic classes) for which we wish to obtain attributes. Obtaining such collections has been studied extensively in previous work [1, 4]. In this poster we use 40 classes chosen manually to have broad coverage.

**Data Sources:** Our Web documents corpus is procured by retrieving the top 200 search results for each instance using Google and removing all non-html documents. The total (compressed) size of the corpus is over 16GB. The query log data contains 50M anonymized English queries submitted to Google.

**Evaluation Methodology:** During evaluation, each candidate attribute extracted for a class is hand-labeled as one of three categories: vital (1.0), okay (0.5) and wrong (0.0; cf. [6]). *Vital* attributes should appear in any complete list of attributes for the target class; *okay* attributes are useful but non-essential; *wrong* attributes are incorrect.

**Results:** Our main results are two-fold. First, extraction from top Web search results yields higher attribute precision than fixed-pattern Web extraction, but has lower precision than extraction from queries, confirming similar results using untargeted Web corpora [3]. Second, the noisy attributes extracted from query logs can be used as additional seed targets for Web search extraction, yielding better precision than either method individually.

## 4. CONCLUSION

Extracting attributes from documents on the Web is difficult due to the presence of noise, however such sources are significantly more content rich than other, more high-precision attribute sources such as query logs. In this poster we develop a conceptually simple seed-based method for leveraging high-precision low-coverage attribute sources (e.g. query logs) in order to improve extraction from high-coverage low-precision sources such as Web documents. This approach yields significantly higher precision than previous methods (both Web and query-log based) and improves coverage by mitigating the "search bias" inherent in query-log based extraction.

| | Precision (%) | | | | Relative Recall (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | 5 | 10 | 20 | 50 | 5 | 10 | 20 | 50 |
| QP | 76.3 | 72.1 | 64.3 | 53.1 | 5.3 | 11.2 | 19.2 | 31.3 |
| QL | 96.5 | 90.9 | 85.6 | 76.5 | 11.6 | 23.5 | 44.3 | 100 |
| WD | 56.5 | 53.5 | 50.4 | 41.8 | 2.1 | 3.5 | 6.7 | 11.9 |
| WS | 96.5 | 78.3 | 62.5 | 43.6 | 9.3 | 11.6 | 13.7 | 17.8 |
| BWS | **97.8** | **94.8** | **88.3** | **76.5** | 9.3 | 21.0 | 41.8 | 76.1 |

**Table 1: Comparative precision of attributes, as well as recall of vital attributes relative to QL, measured at ranks 5, 10, 20 and 50 in the ranked lists of attributes extracted by various methods**

| Class | Attributes |
|---|---|
| AircraftModel | weight, *length,* fuel capacity, wing span, *history, specifications, photographs,* fuel consumption, cost, price |
| CarModel | transmission, *acceleration,* top speed, gearbox, gas mileage, owners manual, transmission problems, engine type, mpg, reliability |
| ChemicalElem | *symbol,* atomic number, mass, *classification,* atomic structure, freezing point, discovery date, number, physical properties, atomic weight |
| Company | *headquarters, chairman,* location, *ceo,* stock price, company profile, corporate office, president, parent company, stock quote |
| Country | *president, area,* population, *flag, economy, religion, climate, geography,* culture, currency |
| Drug | side effects, *dosage, price,* color, withdrawal symptoms, mechanism of action, mechanism, *dangers,* overdose, dose |
| Empire | ruler, *size,* collapse, founding, *location, definition, chronology, downfall,* kings, end |
| Movie | *director,* cast, producer, genre, *crew, synopsis,* official site, release date, script, *actors* |
| Painter | paintings, *biography,* birthplace, works, *artwork, bibliography, autobiography quotations, quotes, biographies* |
| SearchEngine | quality, speed, market share, number of users, reliability, number, mission statement, phone book, algorithms, video search |
| Wine | vintage, *style, color, taste,* wine reviews, *cost, style of wine*, wine ratings, fermentation, aging |
| WorldWarBattle | date, result, *location,* combatants, *images, importance, summary, timeline,* casualties, survivors |

**Table 2: Top 10 attributes extracted by BWS for a few classes; shown in italics if also found among top 10 in QL**

## 5. REFERENCES


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

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

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

[4] K. Shinzato and K. Torisawa. Acquiring hyponymy relations from Web documents. In *Proc. of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 73–80, 2004.

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

[6] E. Voorhees. Evaluating answers to definition questions. In *Proceedings of the 2003 Human Language Technology Conference (HLT-NAACL-03)*, pages 109–111, Edmonton, Canada, 2003.