**Automatic Set Instance Extraction using the Web**

# Richard C. Wang

Language Technologies Institute Carnegie Mellon University [rcwang@cs.cmu.edu](mailto:rcwang@cs.cmu.edu)

**William W. Cohen**

Machine Learning Department Carnegie Mellon University [wcohen@cs.cmu.edu](mailto:wcohen@cs.cmu.edu)

**Abstract**

An important and well-studied problem is the production of semantic lexicons from a large corpus. In this paper, we present a system named ASIA (Automatic Set In- stance Acquirer), which takes in the name of a semantic class as input (e.g., “car makers”) and automatically outputs its in- stances (e.g., “ford”, “nissan”, “toyota”). ASIA is based on recent advances in web- based *set expansion* - the problem of find- ing all instances of a set given a small number of “seed” instances. This ap- proach effectively exploits web resources and can be easily adapted to different languages. In brief, we use language- dependent hyponym patterns to find a noisy set of initial seeds, and then use a state-of-the-art language-independent set expansion system to expand these seeds. The proposed approach matches or outper- forms prior systems on several English- language benchmarks. It also shows ex- cellent performance on three dozen addi- tional benchmark problems from English, Chinese and Japanese, thus demonstrating language-independence.

# Introduction

An important and well-studied problem is the pro- duction of semantic lexicons for classes of in- terest; that is, the generation of all instances of a set (e.g., “apple”, “orange”, “banana”) given a name of that set (e.g., “fruits”). This task is often addressed by linguistically analyzing very large collections of text (Hearst, 1992; Kozareva et al., 2008; Etzioni et al., 2005; Pantel and Ravichandran, 2004; Pasca, 2004), often using hand-constructed or machine-learned shallow lin-

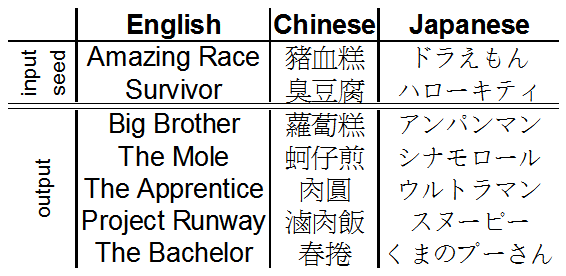


Figure 1: Examples of SEAL’s input and output. English entities are reality TV shows, Chinese en- tities are popular Taiwanese foods, and Japanese entities are famous cartoon characters.

is included within that of another word. For exam- ple, *x* is a hyponym of *y* if *x* is a (kind of) *y*. The opposite of hyponym is *hypernym*.

In this paper, we evaluate a novel approach to this problem, embodied in a system called ASIA1 (Automatic Set Instance Acquirer). ASIA takes a semantic class name as input (e.g., “car makers”) and automatically outputs instances (e.g., “ford”, “nissan”, “toyota”). Unlike prior methods, ASIA makes heavy use of tools for web-based *set ex- pansion*. Set expansion is the task of finding all instances of a set given a small number of exam- ple (seed) instances. ASIA uses SEAL (Wang and Cohen, 2007), a language-independent web-based system that performed extremely well on a large number of benchmark sets – given three correct seeds, SEAL obtained average MAP scores in the high 90’s for 36 benchmark problems, including a dozen test problems each for English, Chinese and Japanese. SEAL works well in part because it can efficiently find and process many *semi-structured* web documents containing instances of the set be- ing expanded. Figure 1 shows some examples of SEAL’s input and output.

SEAL has been recently extended to be robust to errors in its initial set of seeds (Wang et al.,

guistic patterns to detect *hyponym* instances. A hy-

ponym is a word or phrase whose semantic range

1<http://rcwang.com/asia>

2008), and to use bootstrapping to iteratively im- prove its performance (Wang and Cohen, 2008). These extensions allow ASIA to extract instances of sets from the Web, as follows. First, given a semantic class name (e.g., “fruits”), ASIA uses a small set of language-dependent *hyponym patterns* (e.g., “fruits such as ”) to find a large but noisy set of seed instances. Second, ASIA uses the ex- tended version of SEAL to expand the noisy set of seeds.

ASIA’s approach is motivated by the conjecture that for many natural classes, the amount of infor- mation available in semi-structured documents on the Web is much larger than the amount of infor- mation available in free-text documents; hence, it is natural to attempt to augment search for set in- stances in free-text with semi-structured document analysis. We show that ASIA performs extremely well experimentally. On the 36 benchmarks used in (Wang and Cohen, 2007), which are relatively small closed sets (e.g., countries, constellations, NBA teams), ASIA has excellent performance for both recall and precision. On four additional English-language benchmark problems (US states, countries, singers, and common fish), we com- pare to recent work by Kozareva, Riloff, and Hovy (Kozareva et al., 2008), and show comparable or better performance on each of these benchmarks; this is notable because ASIA requires less infor- mation than the work of Kozareva *et al* (their sys- tem requires a concept name and a seed). We also compare ASIA on twelve additional benchmarks to the extended Wordnet 2.1 produced by Snow *et al* (Snow et al., 2006), and show that for these twelve sets, ASIA produces more than five times as many set instances with much higher precision (98% versus 70%).

Another advantage of ASIA’s approach is that it is nearly language-independent: since the underly- ing set-expansion tools are language-independent, all that is needed to support a new target language is a new set of hyponym patterns for that lan- guage. In this paper, we present experimental re- sults for Chinese and Japanese, as well as English, to demonstrate this language-independence.

We present related work in Section 2, and ex- plain our proposed approach for ASIA in Sec- tion 3. Section 4 presents the details of our ex- periments, as well as the experimental results. A comparison of results are illustrated in Section 5, and the paper concludes in Section 6.

# Related Work

There has been a significant amount of research done in the area of semantic class learning (aka lexical acquisition, lexicon induction, hyponym extraction, or open-domain information extrac- tion). However, to the best of our knowledge, there is not a system that can perform set instance ex- traction in multiple languages given only the *name* of the set.

Hearst (Hearst, 1992) presented an approach that utilizes hyponym patterns for extracting can- didate instances given the name of a semantic set. The approach presented in Section 3.1 is based on this work, except that we extended it to two other languages: Chinese and Japanese.

Pantel *et al* (Pantel and Ravichandran, 2004) presented an algorithm for automatically inducing names for semantic classes and for finding their instances by using “concept signatures” (statistics on co-occuring instances). Pasca (Pasca, 2004) presented a method for acquiring named entities in arbitrary categories using lexico-syntactic extrac- tion patterns. Etzioni *et al* (Etzioni et al., 2005) presented the KnowItAll system that also utilizes hyponym patterns to extract class instances from the Web. All the systems mentioned rely on either a English part-of-speech tagger, a parser, or both, and hence are language-dependent.

Kozareva *et al* (Kozareva et al., 2008) illustrated an approach that uses a single hyponym pattern combined with graph structures to learn semantic class from the Web. Section 5.1 shows that our approach is competitive experimentally; however, their system requires more information, as it uses the name of the semantic set and a seed instance.

Pasca (Pas¸ca, 2007b; Pas¸ca, 2007a) illustrated a set expansion approach that extracts instances from Web search queries given a set of input seed instances. This approach is similar in flavor to SEAL but, addresses a different task from that ad- dressed here: for ASIA the user provides no seeds, but instead provides the *name* of the set being ex- panded. We compare to Pasca’s system in Sec- tion 5.2.

Snow *et al* (Snow et al., 2006) use known hyper- nym/hyponym pairs to generate training data for a machine-learning system, which then learns many lexico-syntactic patterns. The patterns learned are based on English-language dependency parsing. We compare to Snow *et al*’s results in Section 5.3.

# Proposed Approach

ASIA is composed of three main components: the Noisy Instance Provider, the Noisy Instance Ex- pander, and the Bootstrapper. Given a semantic class name, the Provider extracts a initial set of noisy candidate instances using hand-coded pat- terns, and ranks the instances by using a sim- ple ranking model. The Expander expands and ranks the instances using evidence from semi- structured web documents, such that irrelevant ones are ranked lower in the list. The Bootstrap- per enhances the quality and completeness of the ranked list by using an unsupervised iterative tech- nique. Note that the Expander and Bootstrap- per rely on SEAL to accomplish their goals. In this section, we first describe the Noisy Instance Provider, then we briefly introduce SEAL, fol-

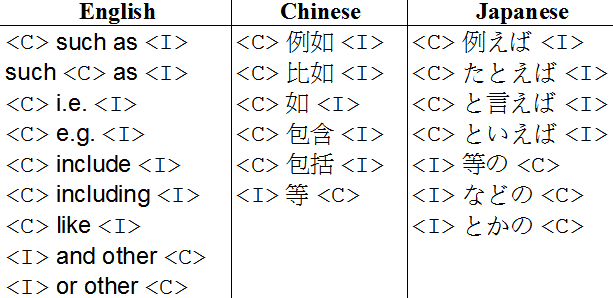


Figure 2: Hyponym patterns in English, Chinese, and Japanese. In each pattern, <C> is a place- holder for the semantic class name and <I> is a placeholder for its instances.

Lastly, the Provider ranks each candidate instance *x* based on its weight assigned by the simple rank- ing model presented below:

lowed by the Noisy Instance Expander, and finally, the Bootstrapper.

*weight* (*x*) =

*sf* (*x,* **S**)

*|***S***|*

*ef* (*x,* **E**)

*× |***E***|*

*wcf* (*x,* **E**)

*× |***C***|*

## Noisy Instance Provider

Noisy Instance Provider extracts candidate in- stances from free text (i.e., web snippets) us- ing the methods presented in Hearst’s early work (Hearst, 1992). Hearst exploited several patterns for identifying hyponymy relation (e.g., such au- thor as Shakespeare) that many current state-of- the-art systems (Kozareva et al., 2008; Pantel and Ravichandran, 2004; Etzioni et al., 2005; Pasca,

where **S** is the set of snippets, **E** is the set of ex- cerpts, and **C** is the set of chunks. *sf* (*x,* **S**) is the snippet frequency of *x* (i.e., the number of snippets containing *x*) and *ef* (*x,* **E**) is the excerpt frequency of *x*. Furthermore, *wcf* (*x,* **E**) is the weighted chunk frequency of *x*, which is defined as follows:

*wcf* (*x,* **E**) = 「 「

1

*dist* (*x, e*) + 1

2004) are using. However, unlike all of those sys- tems, ASIA does not use any NLP tool (e.g., parts- of-speech tagger, parser) or rely on capitalization for extracting candidates (since we wanted ASIA to be as language-independent as possible). This leads to sets of instances that are noisy; however, we will show that set expansion and re-ranking can improve the initial sets dramatically. Below, we will refer to the initial set of noisy instances ex- tracted by the Provider as the *initial* set.

In more detail, the Provider first constructs a few queries of hyponym phrase by using a se- mantic class name and a set of pre-defined hy- ponym patterns. For every query, the Provider re- trieves a hundred snippets from Yahoo!, and splits each snippet into multiple excerpts (a snippet of- ten contains multiple continuous excerpts from its web page). For each excerpt, the Provider extracts all chunks of characters that would then be used as candidate instances. Here, we define a *chunk* as a sequence of characters bounded by punctua- tion marks or the beginning and end of an excerpt.

*e∈***E** *x∈e*

where *dist* (*x, e*) is the number of characters be- tween *x* and the hyponym phrase in excerpt *e*. This model weights every occurrence of *x* based on the assumption that chunks closer to a hyponym phrase are usually more important than those fur- ther away. It also heavily rewards frequency, as our assumption is that the most common instances will be more useful as seeds for SEAL.

Figure 2 shows the hyponym patterns we use for English, Chinese, and Japanese. There are two types of hyponym patterns: The first type are the ones that require the class name *C* to precede its instance *I* (e.g., *C* such as *I*), and the second type are the opposite ones (e.g., *I* and other *C*). In order to reduce irrelevant chunks, when excerpts were extracted, the Provider drops all characters preceding the hyponym phrase in excerpts that contain the first type, and also drops all charac- ters following the hyponym phrase in excerpts that contain the second type. For some semantic class names (e.g., “cmu buildings”), there are no web

documents containing any of the hyponym-phrase queries that were constructed using the name. In this case, the Provider turns to a *back-off* strategy which simply treats the semantic class name as the hyponym phrase and extracts/ranks all chunks co- occurring with the class name in the excerpts.

## Set Expander - SEAL

In this paper, we rely on a set expansion system named SEAL (Wang and Cohen, 2007), which stands for Set Expander for Any Language. The system accepts as input a few *seeds* of some target set *S* (e.g., “fruits”) and automatically finds other probable instances (e.g., “apple”, “banana”) of *S* in web documents. As its name implies, SEAL is independent of document languages: both the written (e.g., English) and the markup language (e.g., HTML). SEAL is a research system that has shown good performance in published results (Wang and Cohen, 2007; Wang et al., 2008; Wang and Cohen, 2008). Figure 1 shows some examples of SEAL’s input and output.

In more detail, SEAL contains three major com- ponents: the Fetcher, Extractor, and Ranker. The *Fetcher* is responsible for fetching web docu- ments, and the URLs of the documents come from top results retrieved from the search engine us- ing the concatenation of all seeds as the query. This ensures that every fetched web page contains all seeds. The *Extractor* automatically constructs “wrappers” (i.e. page-specific extraction rules) for each page that contains the seeds. Every wrap- per comprises two character strings that specify the left and right contexts necessary for extract- ing candidate instances. These contextual strings are maximally-long contexts that bracket at least one occurrence of every seed string on a page. All other candidate instances bracketed by these con- textual strings derived from a particular page are extracted from the same page.

After the candidates are extracted, the *Ranker* constructs a graph that models all the relations between documents, wrappers, and candidate in- stances. Figure 3 shows an example graph where

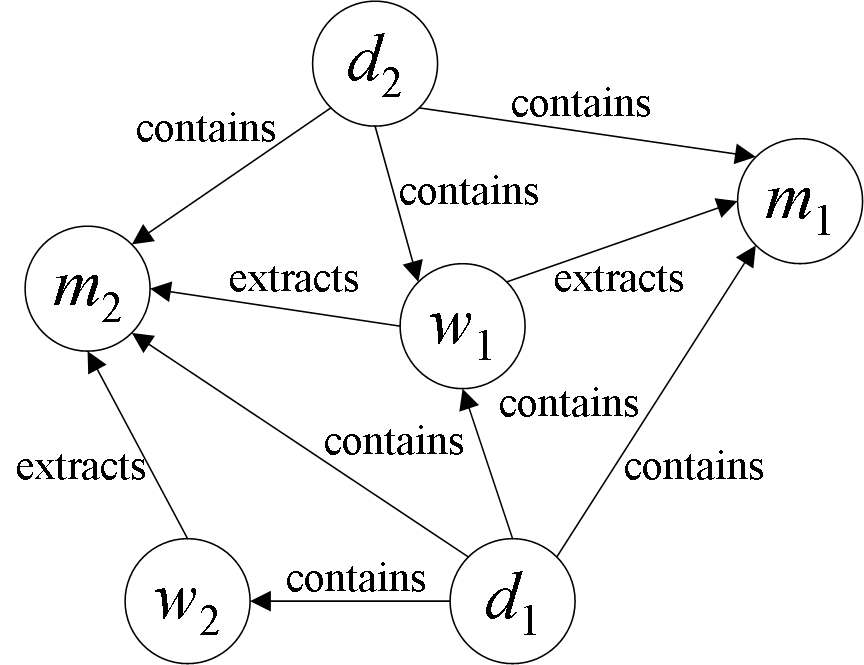


Figure 3: An example graph constructed by SEAL. Every edge from node *x* to *y* actually has an inverse relation edge from node *y* to *x* that is not shown here (e.g., *m*1 *is extracted by w*1).

seed nodes by many short, low fan-out paths. The final expanded set contains all candidate instance nodes, ranked by their weights in the graph.

## Noisy Instance Expander

Wang (Wang et al., 2008) illustrated that it is feasi- ble to perform set expansion on noisy input seeds. The paper showed that the noisy output of any Question Answering system for list questions can be improved by using a noise-resistant version of SEAL (An example of a list question is “Who were the husbands of Heddy Lamar?”). Since the initial set of candidate instances obtained using Hearst’s method are noisy, the Expander expands them by performing multiple iterations of set ex- pansion using the noise-resistant SEAL.

For every iteration, the Expander performs set expansion on a static collection of web pages. This collection is pre-fetched by querying Google and Yahoo! using the input class name and words such as “list”, “names”, “famous”, and “common” for discovering web pages that might contain lists of the input class. In the first iteration, the Expander expands instances with scores of at least *k* in the initial set. In every upcoming iteration, it expands instances obtained in the last iteration that have scores of at least *k* and that also exist in the ini- tial set. We have determined *k* to be 0.4 based on our development set2. This process repeats until

*th*

each node *di* represents a document, *wi* a wrapper,

the set of seeds for *i*

*th*

iteration is identical to that

and *mi* a candidate instance. The Ranker performs

of (*i −* 1)

iteration.

Random Walk with Restart (Tong et al., 2006) on this graph (where the initial “restart” set is the set of seeds) until all node weights converge, and then ranks nodes by their final score; thus nodes are weighted higher if they are connected to many

There are several differences between the origi-

nal SEAL and the noise-resistant SEAL. The most important difference is the Extractor. In the origi-

2A collection of closed-set lists such as planets, Nobel prizes, and continents in English, Chinese and Japanese

nal SEAL, the Extractor requires the longest com- mon contexts to bracket at least one instance of *ev- ery* seed per web page. However, when seeds are noisy, such common contexts usually do not ex- ist. The Extractor in noise-resistant SEAL solves this problem by requiring the contexts to bracket at least one instance of *a minimum of two* seeds, rather than *every* seed. This is implemented using a trie-based method described briefly in the origi- nal SEAL paper (Wang and Cohen, 2007). In this paper, the Expander utilizes a slightly-modified version of the Extractor, which requires the con- texts to bracket as many seed instances as possible. This idea is based on the assumption that irrelevant instances usually do not have common contexts; whereas relevant ones do.

## Bootstrapper

*Bootstrapping* (Etzioni et al., 2005; Kozareva, 2006; Nadeau et al., 2006) is an unsupervised iter- ative process in which a system continuously con- sumes its own outputs to improve its own perfor- mance. Wang (Wang and Cohen, 2008) showed that it is feasible to bootstrap the results of set ex- pansion to improve the quality of a list. The pa- per introduces an iterative version of SEAL called iSEAL, which expands a list in multiple iterations. In each iteration, iSEAL expands a few candi- dates extracted in previous iterations and aggre- gates statistics. The Bootstrapper utilizes iSEAL

# Experiments

## Datasets

We evaluated our approach using the evaluation set presented in (Wang and Cohen, 2007), which contains 36 manually constructed lists across three different languages: English, Chinese, and Japanese (12 lists per language). Each list contains all instances of a particular semantic class in a cer- tain language, and each instance contains a set of synonyms (e.g., USA, America). There are a total of 2515 instances, with an average of 70 instances per semantic class. Figure 4 shows the datasets and their corresponding semantic class names that we use in our experiments.

## Evaluation Metric

Since the output of ASIA is a ranked list of ex- tracted instances, we choose mean average pre- cision (MAP) as our evaluation metric. MAP is commonly used in the field of Information Re- trieval for evaluating ranked lists because it is sen- sitive to the entire ranking and it contains both re- call and precision-oriented aspects. The MAP for multiple ranked lists is simply the mean value of average precisions calculated separately for each ranked list. We define the average precision of a single ranked list as:

*|L|*

「 Prec(*r*) *×* isFresh(*r*)

to further improve the quality of the list returned by the Expander.

*AvgP rec*(*L*) =

*r*=1

Total # of Correct Instances

In every iteration, the Bootstrapper retrieves 25 web pages by using the concatenation of three seeds as query to each of Google and Yahoo!. In the first iteration, the Bootstrapper expands randomly-selected instances returned by the Ex- pander that exist in the initial set. In every upcom- ing iteration, the Bootstrapper expands randomly- selected unsupervised instances obtained in the last iteration that also exist in the initial set. This process terminates when all possible seed com- binations have been consumed or five iterations3 have been reached, whichever comes first. No- tice that from iteration to iteration, statistics are aggregated by growing the graph described in Sec- tion 3.2. We perform Random Walk with Restart (Tong et al., 2006) on this graph to determine the final ranking of the extracted instances.

3To keep the overall runtime minimal.

where *L* is a ranked list of extracted instances, *r* is the rank ranging from 1 to *|L|*, Prec(*r*) is the precision at rank *r*. isFresh(*r*) is a binary function for ensuring that, if a list contains multiple syn- onyms of the same instance, we do not evaluate that instance more than once. More specifically, the function returns 1 if a) the synonym at *r* is cor- rect, and b) it is the highest-ranked synonym of its instance in the list; it returns 0 otherwise.

## Experimental Results

For each semantic class in our dataset, the Provider first produces a noisy list of candidate in- stances, using its corresponding class name shown in Figure 4. This list is then expanded by the Ex- pander and further improved by the Bootstrapper.

We present our experimental results in Table 1. As illustrated, although the Provider performs badly, the Expander substantially improves the



Figure 4: The 36 datasets and their semantic class names used as inputs to ASIA in our experiments.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **English Dataset**  **NP**  **# NP +BS** | | | **NP**  **+NE** | **NP**  **+NE**  **+BS** | **Chinese Dataset**  **NP**  **# NP +BS** | | | **NP**  **+NE** | **NP**  **+NE**  **+BS** | **Japanese Dataset**  **NP**  **# NP +BS** | | | **NP**  **+NE** | **NP**  **+NE**  **+BS** |
| **1.** | 0.22 | 0.83 | 0.82 | 0.87 | **13.** | 0.09 | 0.75 | 0.80 | 0.80 | **25.** | 0.20 | 0.63 | 0.71 | 0.76 |
| **2.** | 0.31 | 1.00 | 1.00 | 1.00 | **14.** | 0.08 | 0.99 | 0.80 | 0.89 | **26.** | 0.20 | 0.40 | 0.90 | 0.96 |
| **3.** | 0.54 | 0.99 | 0.99 | 0.98 | **15.** | 0.29 | 0.66 | 0.84 | 0.91 | **27.** | 0.16 | 0.96 | 0.97 | 0.96 |
| **4.** | 0.48 | 1.00 | 1.00 | 1.00 | **\*16.** | 0.09 | 0.00 | 0.93 | 0.93 | **\*28.** | 0.01 | 0.00 | 0.80 | 0.87 |
| **5.** | 0.54 | 1.00 | 1.00 | 1.00 | **17.** | 0.21 | 0.00 | 1.00 | 1.00 | **29.** | 0.09 | 0.00 | 0.95 | 0.95 |
| **6.** | 0.64 | 0.98 | 1.00 | 1.00 | **\*18.** | 0.00 | 0.00 | 0.19 | 0.23 | **\*30.** | 0.02 | 0.00 | 0.73 | 0.73 |
| **7.** | 0.32 | 0.82 | 0.98 | 0.97 | **19.** | 0.11 | 0.90 | 0.68 | 0.89 | **31.** | 0.20 | 0.49 | 0.83 | 0.89 |
| **8.** | 0.41 | 1.00 | 1.00 | 1.00 | **20.** | 0.18 | 0.00 | 0.94 | 0.97 | **32.** | 0.09 | 0.00 | 0.88 | 0.88 |
| **9.** | 0.81 | 1.00 | 1.00 | 1.00 | **21.** | 0.64 | 1.00 | 1.00 | 1.00 | **33.** | 0.07 | 0.00 | 0.95 | 1.00 |
| **\*10.** | 0.00 | 0.00 | 0.00 | 0.00 | **22.** | 0.08 | 0.00 | 0.67 | 0.80 | **34.** | 0.04 | 0.32 | 0.98 | 0.97 |
| **11.** | 0.11 | 0.62 | 0.51 | 0.76 | **23.** | 0.47 | 1.00 | 1.00 | 1.00 | **35.** | 0.15 | 1.00 | 1.00 | 1.00 |
| **12.** | 0.01 | 0.00 | 0.30 | 0.30 | **24.** | 0.60 | 1.00 | 1.00 | 1.00 | **36.** | 0.20 | 0.90 | 1.00 | 1.00 |
| **Avg.** | 0.37 | 0.77 | 0.80 | 0.82 | **Avg.** | 0.24 | 0.52 | 0.82 | 0.87 | **Avg.** | 0.12 | 0.39 | 0.89 | 0.91 |

Table 1: Performance of set instance extraction for each dataset measured in MAP. NP is the Noisy Instance Provider, NE is the Noisy Instance Expander, and BS is the Bootstrapper.

quality of the initial list, and the Bootstrapper then enhances it further more. On average, the Ex- pander improves the performance of the Provider from 37% to 80% for English, 24% to 82% for Chinese, and 12% to 89% for Japanese. The Boot- strapper then further improves the performance of the Expander to 82%, 87% and 91% respectively. In addition, the results illustrate that the Bootstrap- per is also effective even without the Expander; it directly improves the performance of the Provider from 37% to 77% for English, 24% to 52% for Chinese, and 12% to 39% for Japanese.

The simple *back-off* strategy seems to be effec- tive as well. There are five datasets (marked with \* in Table 1) of which their hyponym phrases return zero web documents. For those datasets, ASIA au- tomatically uses the *back-off* strategy described in Section 3.1. Considering only those five datasets, the Expander, on average, improves the perfor- mance of the Provider from 2% to 53% and the Bootstrapper then improves it to 55%.

# Comparison to Prior Work

We compare ASIA’s performance to the results of three previously published work. We use the best-configured ASIA (NP+NE+BS) for all com- parisons, and we present the comparison results in this section.

## (Kozareva et al., 2008)

Table 2 shows a comparison of our extraction per- formance to that of Kozareva (Kozareva et al., 2008). They report results on four tasks: US states, countries, singers, and common fish. We evaluated our results manually. The results in- dicate that ASIA outperforms theirs for all four datasets that they reported. Note that the input to their system is a semantic class name plus one seed instance; whereas, the input to ASIA is only the class name. In terms of system runtime, for each semantic class, Kozareva *et al* reported that their extraction process usually finished overnight; however, ASIA usually finished within a minute.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N** | **Kozareva** | **ASIA** | **N** | **Kozareva** | **ASIA** |
|  | **US States** |  |  | **Countries** |  |
| 25 | 1.00 | 1.00 | 50 | 1.00 | 1.00 |
| 50 | 1.00 | 1.00 | 100 | 1.00 | 1.00 |
| 64 | 0.78 | 0.78 | 150 | 1.00 | 1.00 |
|  |  |  | 200 | 0.90 | 0.93 |
|  |  |  | 300 | 0.61 | 0.67 |
|  |  |  | 323 | 0.57 | 0.62 |
|  | **Singers** |  |  | **Common Fish** |  |
| 10 | 1.00 | 1.00 | 10 | 1.00 | 1.00 |
| 25 | 1.00 | 1.00 | 25 | 1.00 | 1.00 |
| 50 | 0.97 | 1.00 | 50 | 1.00 | 1.00 |
| 75 | 0.96 | 1.00 | 75 | 0.93 | 1.00 |
| 100 | 0.96 | 1.00 | 100 | 0.84 | 1.00 |
| 150 | 0.95 | 0.97 | 116 | 0.80 | 1.00 |
| 180 | 0.91 | 0.96 |  |  |  |

Table 2: Set instance extraction performance com- pared to Kozareva *et al*. We report our precision for all semantic classes and at the same ranks re- ported in their work.

## (Pas¸ca, 2007b)

We compare ASIA to Pasca (Pas¸ca, 2007b) and present comparison results in Table 3. There are ten semantic classes in his evaluation dataset, and the input to his system for each class is a set of seed entities rather than a class name. We evaluate every instance manually for each class. The results show that, on average, ASIA performs better.

However, we should emphasize that for the three classes: movie, person, and video game, ASIA did not initially converge to the correct in- stance list given the most natural concept name. Given “movies”, ASIA returns as instances strings like “comedy”, “action”, “drama”, and other kinds of movies. Given “video games”, it returns “PSP”, “Xbox”, “Wii”, etc. Given “people”, it returns “musicians”, “artists”, “politicians”, etc. We ad- dressed this problem by simply re-running ASIA with a more specific class name (i.e., the first one returned); however, the result suggests that future work is needed to support automatic construction of hypernym hierarchy using semi-structured web documents.

## (Snow et al., 2006)

Snow (Snow et al., 2006) has extended the Word- Net 2.1 by adding thousands of entries (synsets) at a relatively high precision. They have made several versions of extended WordNet available4. For comparison purposes, we selected the version (+30K) that achieved the best F-score in their ex- periments.

4<http://ai.stanford.edu/>˜rion/swn/

Table 3: Set instance extraction performance com- pared to Pasca. We report our precision for all se- mantic classes and at the same ranks reported in his work.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Target Class** | **System** | **25** | **Precision**  **50 100** | | **@**  **150** | **250** |
| Cities | Pasca | 1.00 | 0.96 0.88 | | 0.84 | 0.75 |
| ASIA | 1.00 | 1.00 | 0.97 | 0.98 | 0.96 |
| Countries | Pasca | 1.00 | 0.98 | 0.95 | 0.82 | 0.60 |
| ASIA | 1.00 | 1.00 | 1.00 | 1.00 | 0.79 |
| Drugs | Pasca | 1.00 | 1.00 | 0.96 | 0.92 | 0.75 |
| ASIA | 1.00 | 1.00 | 1.00 | 1.00 | 0.98 |
| Food | Pasca | 0.88 | 0.86 | 0.82 | 0.78 | 0.62 |
| ASIA | 1.00 | 1.00 | 0.93 | 0.95 | 0.90 |
| Locations | Pasca | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| ASIA | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Newspapers | Pasca | 0.96 | 0.98 | 0.93 | 0.86 | 0.54 |
| ASIA | 1.00 | 1.00 | 0.98 | 0.99 | 0.85 |
| Universities | Pasca | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 |
| ASIA | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Movies | Pasca | 0.92 | 0.90 | 0.88 | 0.84 | 0.79 |
| Comedy Movies | ASIA | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| People | Pasca | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Jazz Musicians | ASIA | 1.00 | 1.00 | 1.00 | 0.94 | 0.88 |
| Video Games | Pasca | 1.00 | 1.00 | 0.99 | 0.98 | 0.98 |
| PSP Games | ASIA | 1.00 | 1.00 | 1.00 | 0.99 | 0.97 |
| **Average** | Pasca | 0.98 | 0.97 | 0.94 | 0.90 | 0.80 |
| ASIA | 1.00 | 1.00 | 0.99 | 0.98 | 0.93 |

For the experimental comparison, we focused on leaf semantic classes from the extended Word- Net that have many hyponyms, so that a mean- ingful comparison could be made: specifically, we selected nouns that have at least three hyponyms, such that the hyponyms are the leaf nodes in the hypernym hierarchy of WordNet. Of these, 210 were extended by Snow. Preliminary experiments showed that (as in the experiments with Pasca’s classes above) ASIA did not always converge to the intended meaning; to avoid this problem, we instituted a second filter, and discarded ASIA’s re- sults if the intersection of hyponyms from ASIA and WordNet constituted less than half of those in WordNet. About 50 of the 210 nouns passed this filter. Finally, we manually evaluated preci- sion and recall of a randomly selected set of twelve of these 50 nouns.

We present the results in Table 4. We used a fixed cut-off score5 of 0.3 to truncate the ranked list produced by ASIA, so that we can compute precision. Since only a few of these twelve nouns are closed sets, we cannot generally compute re- call; instead, we define *relative recall* to be the ratio of correct instances to the union of correct instances from both systems. As shown in the re- sults, ASIA has much higher precision, and much higher relative recall. When we evaluated Snow’s extended WordNet, we assumed all instances that

5Determined from our development set.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Name** | **Snow’s Wordnet (+30k)**  **# Right # Wrong Prec.** | | | **Relative**  **Recall** | **# Right** | **ASIA**  **# Wrong** | **Prec.** | **Relative**  **Recall** |
| Film Directors | 4 | 4 | 0.50 | 0.01 | 457 | 0 | 1.00 | 1.00 |
| Manias | 11 | 0 | 1.00 | 0.09 | 120 | 0 | 1.00 | 1.00 |
| Canadian Provinces | 10 | 82 | 0.11 | 1.00 | 10 | 3 | 0.77 | 1.00 |
| Signs of the Zodiac | 12 | 10 | 0.55 | 1.00 | 12 | 0 | 1.00 | 1.00 |
| Roman Emperors | 44 | 4 | 0.92 | 0.47 | 90 | 0 | 1.00 | 0.96 |
| Academic Departments | 20 | 0 | 1.00 | 0.67 | 27 | 0 | 1.00 | 0.90 |
| Choreographers | 23 | 10 | 0.70 | 0.14 | 156 | 0 | 1.00 | 0.94 |
| Elected Officials | 5 | 102 | 0.05 | 0.31 | 12 | 0 | 1.00 | 0.75 |
| Double Stars | 11 | 1 | 0.92 | 0.46 | 20 | 0 | 1.00 | 0.83 |
| South American Countries | 12 | 1 | 0.92 | 1.00 | 12 | 0 | 1.00 | 1.00 |
| Prizefighters | 16 | 4 | 0.80 | 0.23 | 63 | 1 | 0.98 | 0.89 |
| Newspapers | 20 | 0 | 1.00 | 0.23 | 71 | 0 | 1.00 | 0.81 |
| **Average** | 15.7 | 18.2 | 0.70 | 0.47 | 87.5 | 0.3 | 0.98 | 0.92 |

Table 4: Set instance extraction performance compared to Snow *et al*.

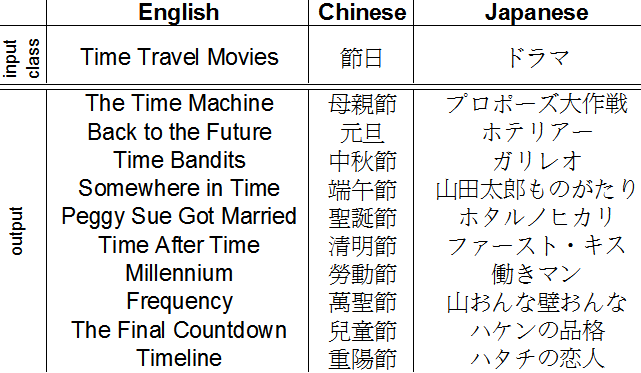


Figure 5: Examples of ASIA’s input and out- put. Input class for Chinese is “holidays” and for Japanese is “dramas”.

were in the original WordNet are correct. The three incorrect instances of Canadian provinces from ASIA are actually the three Canadian terri- tories.

# Conclusions

In this paper, we have shown that ASIA, a SEAL- based system, extracts set instances with high pre- cision and recall in multiple languages given only the set name. It obtains a high MAP score (87%) averaged over 36 benchmark problems in three languages (Chinese, Japanese, and English). Fig- ure 5 shows some real examples of ASIA’s in- put and output in those three languages. ASIA’s approach is based on web-based set expansion using semi-structured documents, and is moti- vated by the conjecture that for many natural classes, the amount of information available in semi-structured documents on the Web is much larger than the amount of information available in free-text documents. This conjecture is given some support by our experiments: for instance,

ASIA finds 457 instances of the set “film direc- tor” with perfect precision, whereas Snow *et al*’s state-of-the-art methods for extraction from free text extract only four correct instances, with only 50% precision.

ASIA’s approach is also quite language- independent. By adding a few simple hyponym patterns, we can easily extend the system to sup- port other languages. We have also shown that Hearst’s method works not only for English, but also for other languages such as Chinese and Japanese. We note that the ability to construct semantic lexicons in diverse languages has obvi- ous applications in machine translation. We have also illustrated that ASIA outperforms three other English systems (Kozareva et al., 2008; Pas¸ca, 2007b; Snow et al., 2006), even though many of these use more input than just a semantic class name. In addition, ASIA is also quite efficient, requiring only a few minutes of computation and couple hundreds of web pages per problem.

In the future, we plan to investigate the pos- sibility of constructing hypernym hierarchy auto- matically using semi-structured documents. We also plan to explore whether lexicons can be con- structed using only the *back-off* method for hy- ponym extraction, to make ASIA completely lan- guage independent. We also wish to explore whether performance can be improved by simul- taneously finding class instances in multiple lan- guages (e.g., Chinese and English) while learning translations between the extracted instances.

# Acknowledgments

This work was supported by the Google Research Awards program.

# References

Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soder- land, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134.

Marti A. Hearst. 1992. Automatic acquisition of hy- ponyms from large text corpora. In *In Proceedings of the 14th International Conference on Computa- tional Linguistics*, pages 539–545.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy.

2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio, June. Association for Computational Linguistics.

Zornitsa Kozareva. 2006. Bootstrapping named entity recognition with automatically generated gazetteer lists. In *EACL*. The Association for Computer Lin- guistics.

David Nadeau, Peter D. Turney, and Stan Matwin.

2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In Luc Lamontagne and Mario Marchand, editors, *Canadian Conference on AI*, volume 4013 of *Lec- ture Notes in Computer Science*, pages 266–277. Springer.

Marius Pas¸ca. 2007a. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In *WWW ’07: Proceedings of the 16th international conference on World Wide Web*, pages 101–110, New York, NY, USA. ACM.

Marius Pas¸ca. 2007b. Weakly-supervised discovery of named entities using web search queries. In *CIKM ’07: Proceedings of the sixteenth ACM con- ference on Conference on information and knowl- edge management*, pages 683–690, New York, NY, USA. ACM.

Patrick Pantel and Deepak Ravichandran. 2004.

Automatically labeling semantic classes. In Daniel Marcu Susan Dumais and Salim Roukos, ed- itors, *HLT-NAACL 2004: Main Proceedings*, pages 321–328, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *CIKM ’04: Proceed- ings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145, New York, NY, USA. ACM.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006.

Semantic taxonomy induction from heterogenous evidence. In *ACL ’06: Proceedings of the 21st Inter- national Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 801– 808, Morristown, NJ, USA. Association for Compu- tational Linguistics.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan.

2006. Fast random walk with restart and its appli- cations. In *ICDM*, pages 613–622. IEEE Computer Society.

Richard C. Wang and William W. Cohen. 2007.

Language-independent set expansion of named enti- ties using the web. In *ICDM*, pages 342–350. IEEE Computer Society.

Richard C. Wang and William W. Cohen. 2008. Iter- ative set expansion of named entities using the web. In *ICDM*, pages 1091–1096. IEEE Computer Soci- ety.

Richard C. Wang, Nico Schlaefer, William W. Co- hen, and Eric Nyberg. 2008. Automatic set ex- pansion for list question answering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 947–954, Hon- olulu, Hawaii, October. Association for Computa- tional Linguistics.