Chinese Semantic Class Learning from Web Based on Concept-Level Characteristics

Wenbo Pang[a], Xiaozhong Fan[a], Jiangde Yu[b], and Yuxiang Jia[c]

[a] School of Computer and Technology, Beijing Institute of Technology
Beijing 100081, China
hananpwb@126.com, {hananpwb, fxz}@bit.com
[b] School of Computer and Information Engineering, Anyang Normal University
Anyang 455000, China
jangder_yu@tom.com
[c] Institute of Computational Linguistics, Peking University
Beijing 100871, China
yxjia@pku.edu.cn

Abstract. The automatic extraction of semantic class instance is a foundational work for many natural language processing applications. One of its crucial problems is how to validate whether a candidate instances is a true class member. Different from the common validation approaches based on the cooccurrence between instances, we present a novel approach based on concept characteristics, including category features, interference semantic classes, and collective instance. Firstly, we analyze the common error instances produced by cooccurrence-based validation from the perspective of concept, and then utilize the concept characteristics to validate the candidate instances. We conduct experiments on eight semantic classes and achieved high accuracies and recall rates, especially on open semantic classes, such as *fish* and *singer*.

Keywords: semantic class, candidate validation, concept characteristics

1 Introduction

Semantic class learning is an important and well-studied task, which takes in a semantic class name as input (e.g. fruits) and automatically outputs its instances (e.g. apple, banana, orange, etc.). Although there are some existing semantic dictionaries, such as WordNet (Miller *et al.*, 1990), they lack the coverage to handle large open domains or rapidly changing categories: Vieira and Poesio (2000) found that, only 56% of antecedent/anaphoric coreferent pairs in hyponymy relations in the WSJ were in WordNet. So, automatic semantic class learning has been the motivating force underlying many applications in lexical acquisition, information extraction, and the construction of semantic taxonomies.

Many methods have been developed for automatic semantic class learning, under the rubrics of lexical acquisition, hyponym acquisition, semantic class identification, and web-based information extraction. Almost all of these approaches face the same crucial problem: how to validate whether a extracted instances is a true class member. Currently, the validation methods mainly are based on the co-occurrence between candidates (or true instances) (Kozareva *et al.*, 2008; Kozareva *et al.*, 2009; Wang and Cohen, 2009).

This kind of validation has three shortages: 1) once some error instances are introduced in the bootstrapping process for some unavoidable reason, they possibly bring more and more error instances, which makes they get a high score in a common re-ranking algorithm; 2) can not reject the error instances caused by people's usage habits or misunderstanding. For example, because 欧盟(European Union) often appears with 美国(America) and 日本(Japan) in the hyponym pattern of countries, although 欧盟 is not a country, it is also accepted by cooccurrence-based validation; and 3) miss the instances that cooccur only with some specific instances, such as 霓虹灯鱼(Neon fish), which only cooccurs with other kind of lamp fish. Because of these shortages, the systems employing cooccurrence-based validation would add other extra constraint to the candidate instances in order to improve system's accuracy, which will reduce the system's recall rate. For example, in Kozareva *et al.* (2008), the golden fish are 1102, but the maximum evaluated fish are 116.

What should a correct instance satisfy? Why do the error instances cooccur with the right instances? We answer these two questions from the viewpoint of concept characteristics. Then utilize three kind of concept characteristics, including the category features that characterizes the usage environment of a candidate instance or a semantic class, the interference semantic classes that are so close to the goal semantic class at the level of concept that people often use them together, and collective instance that is a collective of some correct instances, to validate the candidate instances.

2 Related Work

Weakly supervised learning approaches for automatic semantic class instance extraction have utilized syntactic information (Tanev and Magnini, 2006), cooccurrence statistics (Riloff and Shepherd, 1997), lexico-syntactic contextual patterns (Riloff and Jones, 1999), and local global contexts (Fleischman and Hovy, 2002). The current studies mainly focus on hyponym learning (e.g. "CLASS_NAME such as CLASS_MEMBER" for English) (Hearst, 1992; Snow *et al.*, 2006; Kozareva *et al.*, 2008; Kozareva *et al.*, 2009; Wang and Cohen, 2009).

The early approaches have only evaluated on fixed corpus (Riloff and Jones, 1999; Fleischman and Hovy, 2002). To exploit the huge web resources, Pasca (2004) learned semantic class instances and class groups by acquiring contexts around the pattern. The following studies always are based on web queries (Pasca and Van Durme, 2008; Kozareva *et al.*, 2009; Wang and Cohen, 2009). Following the current studies, in this paper, we use four patterns to extract class candidate instances from web queries.

To validate whether a candidate instance is a true class member, most of the approaches are based on the coocurrence between instances. A representative method is the hyponym pattern linkage graphs (Kozareva *et al.*, 2008), which captures two properties associated with pattern-based extractions: popularity (reflects the times that an instance could be discovered by other instances in the hyponym pattern) and productivity (reflects the times that an instance could lead to the discovery of other instances in the hyponym pattern).

3 The Concept-Level Characteristics of Semantic Class

3.1 Category Features

The usage environments of the instances belonged to the same semantic class should be similar, which is in keeping with the Firthian tradition that "You shall know a word by the company it keeps" (Firth, 1957). For example, as a member of *singers*, the instance would appear with star, sing, album, and concert. We call these strings that can reflect the usage environments of a candidate instance or a semantic class, as category features.

Introducing the category features into semantic class learning has two advantages. One is that the error instances that are introduced by weakly-restricted pattern could be eliminated. For example, in the sentence of 胡锦涛、江泽民等国家领导人入场(Hu Jintao, Jiang Zhemin and other country leaders enter.), since 国家(country) 领导人(leaders) is started with 国家(country), 胡锦涛(Hu Jintao), 江泽民(Jiang Zhemin), both of whom are members of presidents, will be incorrectly recognized as countries. If with the help of the category features, because the category features of these two presidents, such as 出席(attend), 会见(meet with) and 访问(visit), are different to those of *country*, such as 国(nation) and 经济(economic), the algorithm will realize this is a incorrect extraction.

The other advantage is that the low-frequency instances can been effectively recalled, even which seldom or do not appear in the hyponym patterns. The reason lies in that, if the category features of a instance are similar to those of a semantic class, it has the chance to be regarded as a true member of this class, even it only appears several times in the hyponym pattern. For example, 江鱼回 (a kind of fish) never appears in the hyponym pattern of fish, but because it appears once with 刀鱼 (saury) and 江鲶 (river catfish) in a parallel structure, and passes the category features validation, it is effectively recalled as a kind of fish.

3.2 Interference Semantic Classes and Collective Instances

Table 1 shows some error instances, which could pass common coocurrence-based validation approaches.

Table 1: Common Error Instances.

Class	Error Instance	True Class	Error Type
fish	河蟹(crab)	*crab*	interference semantic class's instance
	鱿鱼(squid)	*mollusk*	
province	宁夏(Ning Xia)	*autonomous region*	
country	欧盟(European Union)	*organization*	collective instance
	南美(South American)	*region*	

Before eliminating these error instances from the candidate set, let's analyze why these errors have happened. In our opinion, the reason lies in the conceptual system in people's brains. The real world is a serial of concepts in a people's brain, and these concepts closely connect with others to build a conceptual system, which makes that every semantic class has a set of relational semantic classes. Take *fish* as an example. From the viewpoint of the aquatic organisms, *fish* reminds people of crab and shrimp; from the viewpoint of the food, it reminds people of vegetables and milk. So, when people speak or write, these things would be listed together, for example, 鮰鱼、河蟹、鳗鱼等鱼类 (channel fish, crab, eel and other fish), where 河蟹 (crab) is a kind of crab. This phenomenon makes the first type error in Table 1. Thus, if we have obtained the relational semantic classes of the goal semantic class and extracted their member before validate the goal semantic class's candidate instances; the validation would be more effective. Here, we name the relational semantic classes as the interference semantic classes to the goal semantic class. In addition, because the concepts in people's brain are fuzzy, when people remind some concepts, the instances he uses are not always at a same concept level. For example, people would say 产品主要销往日本、韩国、欧洲等国家 (Products are mainly exported to Japan, Korea, Europe, and other countries). Here, Japan and Korea are the members of countries, but Europe is a region. This phenomenon makes the second type error in Table 1.

In this paper, we name a candidate instance, which takes the true members of the semantic classes as its hyponyms, as a collective instance. According to the above analysis, we validate the candidate instances based on the concept-level characteristics, including the category features, the interference semantic class and the collective instance. The architecture of the proposed algorithm is showed in Figure 1.

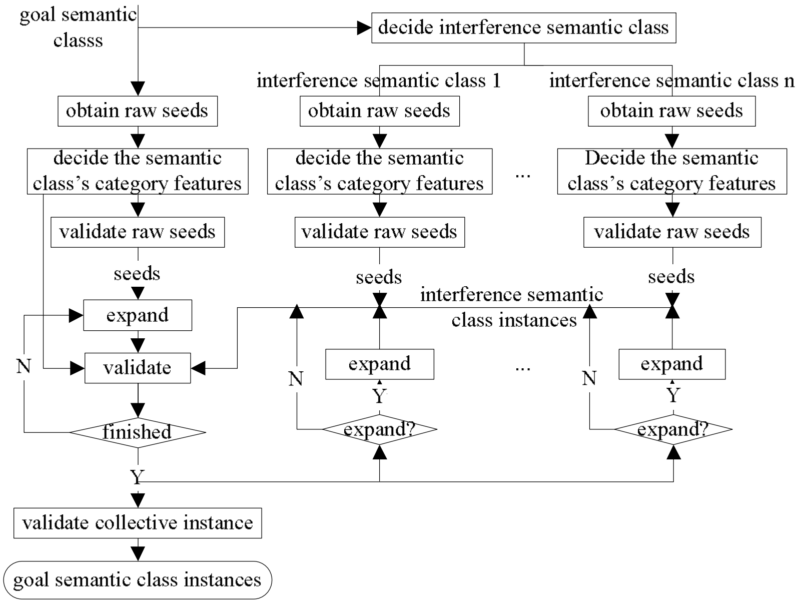
goal semantic classs
decide interference semantic class
interference semantic class 1
interference semantic class n
obtain raw seeds
obtain raw seeds
obtain raw seeds
decide the semantic class's category features
decide the semantic class's category features
...
Decide the semantic class's category features
validate raw seeds
validate raw seeds
validate raw seeds
seeds
seeds
seeds
interference semantic class instances
expand
N
validate
expand
...
expand
N
Y
N
Y
finished
expand?
expand?
Y
validate collective instance
goal semantic class instances

Figure 1: System Architecture.

4 Algorithm

4.1 Obtain Candidate Instances

Like the hyponym patterns in English, there are many hyponym patterns in Chinese (Wang and Cohen, 2009). We employ one single kind of them to query the web and extract semantic class instances:

Pattern 1: 等(and other) CLASS_NAME

Currently, the amount of results of a single query to a common web search engineers is limited. For example, Google provides no more than 1000 results for a single query. Which make the candidate instances extracted from a single query results are not enough. To get as many as possible candidate instances, we introduce other three patterns to bootstrap from the candidate seeds extracted from Pattern 1.

Pattern2:Candidate_Class_Member 等(and other) CLASS_NAME

Pattern3:Candidate_Class_Member 等(and other)

Pattern4:Candidate_Class_Member

Take the semantic class 国家(*country*) as an example, we demonstrate how to use these four patterns. Firstly, we query the web with Pattern 1, 等国家(and other countries), as the query keywords. Suppose we extract a candidate instance 中国(China) from the query results. Then, we fill this candidate instance into the other three patterns to form three query keywords for further query: 中国等国家(China and other countries), 中国等(China and other) and 中国(China).

Utilizing these four patterns, we extract the strings as candidate instances, which should exist in parallel structure and are separated by “、” (a kind of Chinese punctuation).

4.2 Validation

The extracted candidates should pass the following three-stage validation, including category features validation, interference semantic class validation, and collective instances validation.

4.2.1 Category Features Validation

Before conducting category features validation, it is necessary to extract the category features of the candidate and those of the target semantic class.

The category features of a string are used to characterize its usage environment. In this paper, we extract the category features of a string from the sentences, which are obtained from web query results based on the string's Pattern 3. Figure 2 describes this algorithm.

```
Input: a candidate instance $i$
       sentence set that $i$ is in Pattern 3 $S$
Output: the category features set $F$ of $i$
Variables: $T_s$=the threshold for $|S|$
Steps:
    if $|S| < T_s$
        return $F$=null
    else
        $\{< s, f >\}$=get the strings $s_i$ following $i$等, and count its frequency $f_i$. Descending-order
        sort s by their frequencies and set $s_i$'s $f_i = 0$. Here, the length of $s_i$ is limited to 2 and 3.
    for each $s_i$ in s
        for each $S_j$ in $S$
            if the fragment that follows to $i$等 in $S_j$ contains $s_i$
                $f_i$++ and remove $S_i$ from $S$
    construct the $F$ with high frequent $s_i$
    return $F$
```

Figure 2: Extract the category features of a string

To get a semantic class's category features, we cluster the category features of the most typical instances, which are obtained using Pattern 1, and take the common features as the class's category features.

When conducting a candidate instance's category features validation, we compare its features to those of the semantic class.

4.2.2 Interference Semantic Classes Validation

We employ Pattern A and Pattern B to obtain the interference semantic classes of goal semantic class.

Pattern A:分为(divided into)GOAL_CLASS _NAME

Pattern B:包括(include)GOAL_CLASS _NAME

Take the semantic class 国家(*country*) as an example, the query keywords constructed by these two patterns are 分为国家(divided into countries) and 包括国家(include countries).

Extract the strings that parallel with GOAL_CLASS _NAME in query results, and regard the strings whose frequencies are bigger than a threshold as interference class. For example, the strings贝类(shellfish),虾类(shrimp) would be extracted from分为鱼类、头足类、贝类、虾类、蟹类等(Divided into fish, cephalopods, shellfish, shrimp, crab, etc.).

The same candidate instance perhaps simultaneously exists in candidate set of the goal semantic class and that of an interference class. We name this situation as instance collision. When an instance collision happens, this trigger instance perhaps doesn't belong to the goal semantic class. Through comparing the frequencies that this instance appears in the Pattern 2 of the different semantic classes, we decide the trigger instance's real class. In this paper, we take the class with bigger appearance frequency as the winner. If the frequencies are equal, remove this instance from both classes.

For example, when find 河蟹(crab) exists both in fish class's candidate class set and in crab class's, we will compare whose appearance frequency is bigger, 河蟹等鱼类（crab and other fish）or 河蟹等蟹类(crab and other crab). In our experiments, the first is 6, and the second is 23, so 河蟹 is regard as a member of the crab.

4.2.3 Collective Instance Validation

Like the recognition method of interference class, two hyponym patterns are used to recognize collective instances.

Pattern I: CANDIDATE_INSTANCE分为(divided into)

Pattern II: CANDIDATE_INSTANCE包括(include)

After extracting the hyponyms of a candidate instance using these two patterns, if many of these hyponyms exist in the goal semantic class's candidate instance set, this candidate instance perhaps is a collective instance. For example, because the hyponyms of 欧盟(European Union) contains France, Germany, Italy and so on, which also exist in the candidate instance set of country, 欧盟 would be regard as a collective instance.

The collective instance finding algorithm is list in Figure 3.

```
Input: the candidate instance $i$
       the candidate instance set $I$
Output: whether $i$ is a collective instance.
Steps:
1. $\{< H_c, F_c >\}$=Use Patter I, II to obtain the candidate hyponyms of $i$, and count every candidate
   hyponym's frequency. Record the number of total sentences $N_s$
2. for each candidate hyponym $H_c$ in $\{< H_c, F_c >\}$
3.   if $F_c > T_f$
4.     put $H_c$ into $H$
5. n=the number of $h \in H \cap h \in I$
6. if $n/|C| > T_h$
7.   return true
8. else
9.   return false
```

Figure 3: Validate collective instance

Here,

$$T_f = \frac{n}{2 * log_{10} N_s}, \qquad T_h = 2 * log_{10} N_s$$

5 Experiments

5.1 Data

We evaluate the proposed algorithm on eight semantic classes: 国家 (*countries*), 省 (*China-provinces*), 美国总统 (*American Presidents*), 朝代 (*China-dynasties*), 星座 (*constellations*), 鱼类 (*fish*), 歌星 (*singers*), 自然风景区 (*natural scenic spots*). The first 5 classes are closed sets, and the other 3 classes are open sets. Our experimental results are manually reviewed for correctness. There are two reasons: 1) some instances have abbreviation, alias or different transliterations. For example, 沙特阿拉伯(Saudi Arabia) often is called 沙特 for short. 新西兰(New Zealand) sometimes be translated to 纽西兰. 2) There always are spelling errors in the web queries.

When calculating the experiments performance, we only keep the first correct instance and omit its following abbreviation, alias, different transliterations and spelling error formats.

The common web search engineer used in our experiments is BaiDu[1], a popular Chinese search engineer. It provides no more than 760 results for a single query.

5.2 Experimental Results

5.2.1 Examples of Category Features

Table 2 shows the top category features of *fish*, *crab* and *countries*.

[1] http://www.baidu.com/

Table 2: Category features of several semantic classes.

fish	鱼类(fish)	品种(kind)	多种(many kinds)	各种(all kinds of)
	特色(character)	鱼种(fish kind)	养殖(cultivation)	水产(aquatic)
crab	品种(kind)	特色(character)	海鲜(seafood)	多种(many kinds)
	水产(aquatic)	产品(product)	各种(all kinds of)	蟹类(crab)
countries	国家(country)	国的(of country)	地的(of location)	地区(region)
	国外(foreign)	国也(country too)	世界(world)	国际(international)

Table 3: Category features of several instances.

石斑 (grouper)	名贵(valuable)	海鲜(seafood)	海鱼(sea fish)	鱼类(fish)
	品种(kind)	鱼种(fish kind)	优质鱼(quality fish)	价格(price)
肉鱼 (meat and fish)	食品(food)	食物(food)	动物(animal)	在调味(at seasoning)
	荤菜(meat)	物资(material)	可以(could)	副食(non- staple food)

When compare the category features of *fish* with those of *crab*, we can find that five features are same. This reflects that fish and crab are close at conceptual level. At the same time, it shows that the category features contains not only the features of a single semantic class, but those of a semantic class set.

Table 3 shows the top 9 category features of candidate instances of *fish*, here,石斑(grouper) is a true member of fish, but肉鱼(meet and fish) is not.

If we only read the category features column of the above table, even do not know what are 石斑 and 肉鱼, we can easily infer that the first maybe a kind of fish and the second should be a kind of food.

5.2.2 Examples of Interference Semantic Class

Table 4 shows some of the interference semantic classes of *fish*.

Table 4: Interference Semantic Classes of *fish*.

贝类(shellfish)	哺乳类(mammals)	藻类(algae)	虾类shrimp
甲壳类crustaceans	爬行动物reptiles	软体动物mollusca	蟹类crab

When decide whether a candidate belongs to a interference semantic class, the instance collision mentioned in section 4.2.2 always happens. For example, 河蟹(crab), which is a kind of crab, appears 23 times in the template of *crab*, and 6 times in the template of *fish*. Because 6 is smaller than 23, according to the criteria of resolving collision, 河蟹 is decided as an instance of the interference semantic class. Another example is 墨鱼(cuttlefish), which is a kind of mollusca. It appears 5 times in the template of *mollusca*, and 2 times in the template of *fish*. It also is decided as an instance of the interference semantic class

5.2.3 Examples of Collective Instance

Table 5 shows how to validate whether a candidate instance is a collective instance. In this table, we take two candidate instance of *countries* as examples: 欧盟 (European Union), which is a collective instance, and 美国 (America) which is not. For each of them, we list their respective top 5 hyponyms. Here, "contained?" stands for whether the hyponym is contained in the candidate set of *countries*. Since the hyponyms of 欧盟 all are contained in the candidate set, the system will regard 欧盟 as a collective instance. And the hyponyms of 美国, such as 加利福尼亚, which is a state of America, are not contained in the candidate set of *countries*, 美国 will pass this validation, and be regarded as a true instance of *countries*.

Table 5: Validating Collective Instance.

欧盟 (European Union)	hyponym concept	法国 (France)	德国 (Germany)	英国 (Britain)	意大利 (Italy)	比利时 (Belgium)
	contained?	√	√	√	√	√
美国 (America)	hyponym concept	加利福尼亚 (California)	南卡罗来纳 (South Carolina)	政治 (politic)	经济 (economic)	总统 (president)
	contained?	×	×	×	×	×

Table 6: Performance Comparison.

N	Kozareva	ASIE	Ours
countries			
50	100%	100%	100%
100	100%	98%	100%
150	100%	99%	98 %
200	90%	93%	92%
300	61%	66%	68.2%
323	57%	62%	-
common fish			
10	100%	100%	100%
50	100%	100%	100%
75	93%	97%	100%
100	84%	97%	100%
116	80%	97%	100%
200	-	-	98.5%
500	-	-	97.2%
1092	-	-	88.6%

Table 7: Performance Comparison.

N	Kozareva	Ours
singer		
10	100%	100%
25	100%	100%
50	97%	100 %
75	96%	100%
100	96%	95%
150	95%	94%
180	-	94.4%
300	-	92%
500	-	91.6%
821	-	90.2%

5.2.4 System Performance

Table 6 shows a performance comparison of our system to that of Kozareva *et al.* (2008) and that of Wang and Cohen (2009). Because we can not obtain other Chinese extraction system, the compared systems both are on English.

We also compare our system on *fish* to that of Kozareva *et al.* (2008) as shown in Table 7.

From the Table 6 and Table 7, it is observed that our algorithm is more suitable to an open semantic class, such as *fish* and *singers*. In Kozareva *et al.* (2008), the number of fish is 116, and the number of singers is 180. However, in our results, the number of extracted fish is 1092, achieved 88.6% precision, and the number of extracted singers is 821, achieved 90.2% precision.

Table 8 lists our extraction performances on other 5 semantic classes.

5.3 Error Analysis

For fish, the first error instance is 鱿鱼(squid), which is a kind of mollusca. When we check the data record, we find that our system indeed extracted it as a kind of mollusca. But when deciding which it belongs to, fish or mollusca, our system compares the frequencies that 鱿鱼 appears in Patter 2 of these two different class: 鱿鱼等鱼类(squid and other fish) appears 26 times, but 鱿鱼等软件动物(squid and other mollusca) only appears 5 times. So, it was regarded as a fish's instance. This mistake is related to people's misunderstand that 鱿鱼 is a kind of fish. So, to some extent, this error shows that our proposed algorithm reflects the concept system in people's brain.

Another type of mistake our system will make, when an instance belongs the goal class and one of interference classes simultaneously. For example, 刘德华 is a singer; meanwhile, he also

Table 8: Performance.

Semantic Class	N	Accuracy	Semantic Class	N	Accuracy
省 (*China-provinces*)	10	100%	星座 (*constellations*)	20	100%
	23	100%		40	97.5%
	37	62.2%		80	93.8%
美国总统 (*American Presidents*)	20	100%		114	77.2%
	30	100%	自然风景区 (*natural scenic spots*)	25	100%
	44	95.5%		50	100%
	73	60.3%		100	100%
朝代 (*China-dynasties*)	20	100%		200	95.5%
	40	97.5%		500	95.4%
	80	78.8%		800	92%
	97	70.1%		983	89%

is a movie star, which is an interference class to singers. Because he is more famous as a movie star than as a singer, our algorithm excludes him from singers.

6 Conclusions and Future Work

The candidate instances validation based on conceptual characteristics, such as category features, interference semantic class and collective instance, is an effective way to filter out the error instances. Further, it makes more patterns could be utilized to obtain candidate instances, and improves the recall rate of open semantic classes.

In this paper, we attempt to build the hierarchy of a given semantic class, and then look a single instance from the perspective of semantic hierarchy. Unfortunately, since the approaches that are used to resolve the sub-tasks, such as obtaining the interference classes, are not sophisticated enough, the results are not satisfying. We will improve these approaches in the future works.

References

Firth, J.R. 1957. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*, pp.1–32.

Fleischman, M. and E. Hovy. 2002. Fine grained classification of named entities. *Proceedings of the 19th International Conference on Computational Linguistics, Volume 1*, pp. 267–273.

Hearst, M.A. 1992. Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th International Conference on Computational Linguistics, Volume 2*, pp. 539–545.

Kozareva, Z., E. Riloff and E. Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. *Proceedings of ACL-08: HLT*, pp. 1048–1056.

Kozareva, Z., E. Riloff and E. Hovy. 2009. Learning and Evaluating the Content and Structure of a Term Taxonomy. *AAAI-09 Spring Symposium on Learning by Reading and Learning to Read.*

Miller, G., R. Beckwith, C. Fellbaum, D. Gross and K. J. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 235–312.

Pasca, M. 2004. Acquisition of categorized named entities for web search. *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pp. 137–145.

Pasca, M. and B. Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from Web documents and query logs. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pp. 19–27.

Riloff, E. and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 117–124.

Riloff, E. and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the National Conference on Artificial Intelligence*, pp. 474–479.

Snow, R. D. Jurafsky and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pp. 801–808.

Tanev, H. and B. Magnini. 2006. Weakly supervised approaches for ontology population. *Proceedings of EACL-2006*, pp. 3–7.

Vieira, R. and M. Poesio. 2000. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4), 539–593.

Wang, Richard C. and William W. Cohen. 2009. Automatic Set Instance Extraction using Web. *Proceedings of the 18th International Conference on World Wide Web*, pp. 101–110.