# Intrinsic Property-based Taxonomic Relation Extraction from Category Structure

**DongHyun Choi and Eun-Kyung Kim and Sang-Ah Shim and Key-Sun Choi**
Semantic Web Research Center
KAIST
cdh4696, kekeeo, sashim, kschoi@world.kaist.ac.kr

## Abstract

We propose a novel algorithm to extract taxonomic (or *isa/instanceOf*) relations from category structure by classifying each category link. Previous algorithms mainly focus on lexical patterns of category names to classify whether or not a given category link is an *isa/instanceOf*. In contrast, our algorithm extracts intrinsic properties that represent the definition of given category name, and uses those properties to classify each category link. Experimental result shows about 5 to 18 % increase in F-Measure, compared to other existing systems.

## 1 Introduction

### 1.1 Problem Description

Taxonomies are a crucial component of many applications, including document clustering (Hotho et al., 2003) and database search (Byron et al., 1997). Due to their importance, many studies have examined methods of extracting taxonomic relations automatically - either from unstructured text (Cimiano et al., 2005; Cimiano(2) et al., 2005), or from structured data such as Wikipedia category structures (Ponzetto and Strube, 2007; Nastase and Strube, 2008; Suchanek et al., 2007). Many researchers have attempted to obtain taxonomic relations from unstructured text to constuct a taxonomy, but in most cases such a system shows poor precision and low recall. Approaches to extracting taxonomic relations from structured data show relatively high performance, but to obtain a taxonomy these require huge amounts of structured data. Recently, as large amounts of structured data such as the infoboxes and category structures of Wikipedia or DBpedia (Auer et al., 2007) have become available, an obstacle to this approach has been removed.

Although a category structure does contain some kind of hierarchical structure, in many cases it cannot be considered as an *isa/instanceOf* hierarchy. For example, the article "Pioneer 11[1]" on Wikipedia is categorized under "Radio frequency propagation", which is related to the "Pioneer 11" but is obviously not a taxonomical parent of "Pioneer 11".

In this paper, we propose a method for extracting taxonomic relations from a given category structure. More precisely, for a category link in the given category structure, the algorithm determines whether the link could be considered an *isa/instanceOf* relation, or if the link simply represents a broader term/narrower term/related term relation. For a given category link <A, B>, in which A is the upper category name and B is the lower category/article name, we attempt to get the definition of B to classify the link. More precisely, we analyze the upper categories of B from the given category structure, to get tokens that represents the definition of B. Once we get the tokens, we compare the tokens with the name of A, to classify the given category link. We call the tokens that represent the definition of B "intrinsic tokens" of B; a more precise definition will be presented in section 3.1.

To show the validity of this approach, the algorithm is applied to Wikipedia's category structure,

[1] Pioneer 11 was the probe for second mission of the Pioneer program (after its sister probe Pioneer 10) to investigate Jupiter and the outer solar system.



*Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010)*, pages 48–57,
Beijing, August 2010

to obtain taxonomic relations there. Wikipedia's category structure consists of categories, article titles and links between them. A Wikipedia article represents one document, and a category is the grouping of those articles by non-categorization-expert users. Each category has its own name, which is assigned by these users.

Although Wikipedia's category structure is built by non-experts, it can be thought of as reliable since it is refined by many people, and it contains 35,904,116 category links between 764,581 categories and 6,301,594 articles, making it a perfect target for an experimental taxonomic relation extraction algorithm.

After describing related works in section 2, our detailed algorithm is proposed in section 3, and its experimental results are discussed in section 4. In section 5, we make some conclusions and proposals for future work.

## 2 Related Works

Methods of taxonomic relation extraction can be divided into two broad categories depending on the input: unstructured or structured data. The extraction of taxonomic relations from unstructured text is mainly carried out using lexical patterns on the text. The Hearst pattern (Hearst, 1992) is used in many pattern-based approaches, such as Cimiano (2005).

In addition, there has been research that attempted to use existing structured data, like the Wikipedia category structure or the contents of a thesaurus. The system of Ponzetto (2007) determines whether or not the given Wikipedia category link is an *isa/instanceOf* relation by applying a set of rules to the category names, while Nastase (2008) defined lexical patterns on category names, in addition to Ponzetto (2007). The YAGO system (Suchanek et al., 2007) attempts to classify whether the given article-category link represents an *instanceOf* relation by checking the plurality of the upper category name.

The algorithm proposed in this paper focuses on the structured data, mainly the category structure, to gather *isa/instanceOf* relations. The system gets a category structure as input, and classifies each category link inside the category structure according to whether it represents an *isa/instanceOf* relation or not.

## 3 Algorithm Description

In section 3.1, we introduce the necessary definitions for *isa/instanceOf* relations and the required terms to describe the algorithm. In section 3.2, we will discuss the hypotheses based on the definitions described in section 3.1. Next, two binary classification algorithms will be proposed based on the hypotheses, which will determine whether the given category link is an *isa/instanceOf* relation or not.

### 3.1 Definitions

To define *isa* and *instanceOf* relations, Mizoguchi (2004) introduces the concept of *intrinsic property* and other related concepts, which are shown in the following definitions 1, 2 and 3:

***Definition 1:*** **Intrinsic property.** The intrinsic property of a thing is a property which is essential to the thing and it loses its identity when the property changes.

***Definition 2:*** **The ontological definition of a class.** A thing which is a conceptualization of a set X can be a class if and only if each element x of X belongs to the class X if and only if the intrinsic property of x satisfies the intensional condition of X. And, then and only then, <x *instanceOf* X> holds.

***Definition 3: isa*** **relation.** *isa* relation holds only between classes. <class A *isa* class B> holds iff the instance set of A is a subset of the instance set of B.

In addition, we define the following terms for algorithm description:

***Definition 4:*** **intrinsic token.** Token [2] T is an intrinsic token of B iff T represents the intrinsic property of B.

For example, when B is "Pioneer 11", the intrinsic tokens of B are "spacecraft", "escape[3]", "Jupiter", etc.

[2]For example, token is a segmented term in category names of Wikipedia category structure.

[3]Since the main purpose of Pioneer 11 is to escape from the solar system and fly into the deep space, we thought "escape" is the intrinsic token of "Pioneer 11". In the same context, "spacecraft escaping the solar system" is a taxonomical parent of "Pioneer 11".

***Definition 5:*** **category link.** <A, B> is called category link iff A is a category of B, and that fact is explicitly stated in the given category structure.

Consider the example of Wikipedia. If B is an article, <A, B> is called an **article-category link**, and if B is a category, <A, B> is called a **category-category link**. The article is a categorized terminal object.

***Definition 6:*** **category structure.** Category structure is the collection of category links, its component categories, and categorized terminal objects.

***Definition 7:*** **upper category set.** The upper category set of B is defined as the set of upper categories of B up to n step in the given category structure, and it is expressed as U(B, n).

For example, if the two category links <*Jupiter spacecraft, Pioneer 11*> and <*Jupiter, Jupiter spacecraft*> exist inside the given category structure, then *Jupiter spacecraft* is the element of U(Pioneer 11, 1), while *Jupiter* is not.

Figure 1 shows the category structure of U(Pioneer 11, 3),which we refer to throughout this paper to explain our algorithm.

### 3.2 Hypotheses

According to the classical Aristotelian view, categories are discrete entities characterized by a set of properties shared by their members. Thus, we make the following lemmas:

**Lemma 1:** If some objects are grouped into the same category, then they share at least more than one property.

According to definition 2, if x is an *instanceOf* X, then the intrinsic property of x satisfies the definition of X. Since the intrinsic property is the property related to the definition of the object, we can assume that in most categorization systems, the intrinsic property is the most frequently shared property among those objects categorized in the same category.

**Lemma 2.** Intrinsic properties are shared most frequently among objects in a category.

Lemma 2 means that, for example, the intrinsic token T of B will show up frequently among the names of upper categories of B. But lemma 2 does NOT mean that non-intrinsic tokens will not frequently appear among the upper category names. For example, the elements of U(Pioneer 11, 3) from the Wikipedia category structure contain the token "spacecraft" 4 times, but it also contain token "technology" 3 times. Therefore, we cannot directly use the token frequency to determine which one is the intrinsic token: rather, we make another assumption to get the "intrinsic score" for each token.

**Lemma 3.** Intrinsic tokens co-occur frequently with other intrinsic tokens.

Lemma 3 means that, if T1 is an intrinsic token of B, and T2 co-occurs with T1 inside the upper category names of B, then there is a high probability that T2 is also an intrinsic token of B. For example, for the category link <Jupiter spacecraft, Pioneer 11>, if the token "spacecraft" is an intrinsic token of "Pioneer 11", we can assume that the token "Jupiter" is also an intrinsic token of "Pioneer 11". Since some intrinsic tokens that are appropriate as modifiers are not appropriate as head words – for example, if the token "Jupiter" is used as a modifier, it will be a *good* intrinsic token of "Pioneer 11", but if it is used as a head word, choosing it as the intrinsic token of "Pioneer 11" would be *bad* choice – thus, we distinguish between intrinsic score as head word, and intrinsic score as modifier. If the intrinsic score of token T is high for article/category name B, then it means the probability is high that T is an intrinsic token of B. We assumed that only the co-occurrences as head word and its modifier are meaningful. **Corollary 3-1.** If a modifier co-occurs with a head word, and the head word is frequently an intrinsic token of an object, then the modifier is an intrinsic token of the object.

**Corollary 3-2.** If a head word co-occurs with a modifier, and the modifier is frequently an intrinsic token of an object, then the head word is an intrinsic token of the object.

### 3.3 Proposed Algorithm

Based on the hypotheses proposed in section 3.2, we propose two algorithms to get the intrinsic score of each token in the following sections. The first algorithm, a counting-based approach, uses only lemmas 1 and 2, and it will be shown why this algorithm will not work. The second algorithm, a graph-based approach, uses all of the hy-

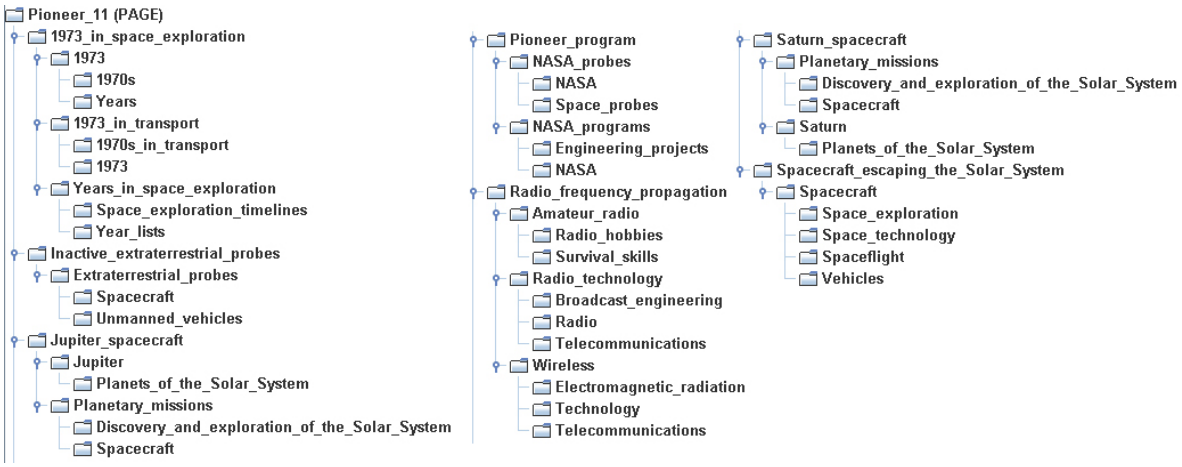


Figure 1: category structure of U(Pioneer 11, 3) from Wikipedia.

potheses to solve the problem.

For the given category link <A, B>, the intrinsic score of each token will be calculated based on its frequency inside U(B, n) while separately counting the token's intrinsic score as modifiers and the intrinsic score as head word. We here propose a scoring mechanism based on the HITS page ranking algorithm (Kleinberg, 1999): For the given category link <A, B>, we first construct a "modifier graph" using U(B, n), and then calculate the intrinsic score for each token in U(B, n) using the HITS algorithm. After that, the intrinsic score of each token will be used to calculate the score of <A, B>. If the score is higher than some predefined threshold, then <A, B> is classified as an *isa/instanceOf* link, and otherwise it is not.

### 3.3.1 Counting-based Approach

This method utilizes lemmas 1 and 2 to get the intrinsic score for each token, and then uses the score to determine whether the given category link is an *isa/instanceOf* link or not.

To utilize this approach, we first score each token from U(B, n) by counting the frequency of each token from the words of U(B, n). Table 1 shows the score of each token from U(Pioneer, 3) for figure 1.

For the "Pioneer 11" article, there are seven category links in Wikipedia's category structure: <1973 in space exploration, Pioneer 11>, <Inactive extraterrestrial probes, Pioneer 11>, <Jupiter spacecraft, Pioneer 11>, <Pioneer program, Pioneer 11>, <Radio frequency propagation, Pioneer 11>, <Saturn spacecraft, Pioneer 11>, and <Spacecraft escaping the Solar System, Pioneer 11>, as shown in figure 1. The scores of each link using a counting-based approach are acquired by adding the scores for each token in table 1 that is matched with single term occurrence in category names. Table 2 shows the result of counting-based approach.

| **Token** | **Score** |
|---|---|
| space | 6 |
| exploration | 5 |
| spacecraft, probe | 4 |
| 1973, technology, year, radio, solar, system, nasa | 3 |
| vehicle, radio, program, 1970s, extraterrestrial, transport, Saturn, Jupiter | 2 |
| escape, inactive, frequency, propagation, pioneer, ... | 1 |

Table 1: Score for each token from U(Pioneer 11, 3)

Although the link <1973 in space exploration, Pioneer 11> receives the highest score among those seven links, obviously the link does not represent *isa/instanceOf* relation. This shows that the counting approach does not guarantee accuracy. Table 1 shows that non-intrinsic tokens occur frequently (such as 'technology' in this exam-

| Article-Category Links | Score |
|---|---|
| <1973 in space exploration, Pioneer 11> | 3+6+5=14 |
| <Spacecraft escaping the Solar System, Pioneer 11> | 4+1+3+3=11 |
| <Inactive extraterrestrial probes, Pioneer 11>, | 1+2+4=7 |
| <Saturn spacecraft, Pioneer 11> | 2+4=6 |
| <Jupiter spacecraft, Pioneer 11> | 2+4=6 |
| <Radio frequency propagation, Pioneer 11> | 2+1+1=4 |
| <Pioneer program, Pioneer 11> | 1+2=3 |

Table 2: Scoring each category links using counting approach

ple). We call this an 'overloaded existence' error. To solve the problems described above, we apply Lemma 3, Corollary 3-1 and 3-2 to our calculation, and propose a second algorithm based on a graph-based approach, which will be explained in the next section.

### 3.3.2 Graph-based Approach

In this section, we propose a graph-based approach to get the intrinsic score of each token. To do this, we first construct a modifier graph from the words of U(B, n) for a given category link <A, B>, with each node representing a token from the elements of U(B, n), and each edge representing the co-occurrence of tokens inside each element of U(B, n). Next, we apply a well-known graph analysis algorithm to that graph, and get the intrinsic scores for each node. Finally, we use the score of each node to get the score of the given category link.

**Constructing modifier graph** Modifier graph constructed here is defined as a directed graph, in which each node represents each token inside U(B, n), and each edge represents a co-occurrence as modifier-head relation inside each category name of U(B, n). Using the subset of U(Pioneer 11, 3), we get the modifier graph of figure 2.[4]

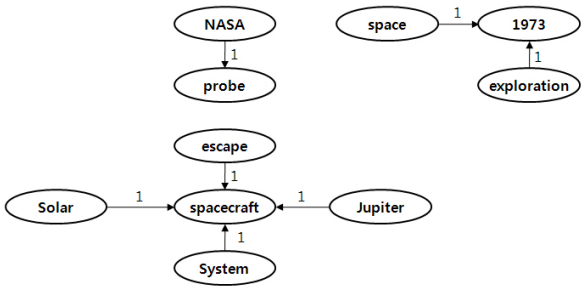


Figure 2: Modifier graph of the subset of U(Pioneer 11, 3): {Spacecraft escaping the Solar System, Jupiter spacecraft, 1973 in space exploration, NASA probes, Saturn}

**Calculating Intrinsic score** After constructing the modifier graph, we apply the HITS algorithm to the modifier graph. Since the HITS algorithm cannot reflect the weight of edges, a modified version of the HITS algorithm (Mihalcea and Tarau, 2005) is adopted:

$$\textit{Authority}(V_i) = \sum_{V_j \in In(V_i)} e_{ji} \cdot \textit{Hub}(V_j) \quad (1)$$

$$\textit{Hub}(V_i) = \sum_{V_j \in Out(V_i)} e_{ij} \cdot \textit{Authority}(V_j) \quad (2)$$

$In(V_i)$ represents the set of vertices which has the outgoing edge to $V_i$, $Out(V_i)$ represents the set of vertices which has the incoming edge from $V_i$, and $e_{ij}$ represents the weight of the edge from $V_i$ to $V_j$. The algorithm for calculating the scores is as follows:

1. Initialize the authority and hub score of each node to one.

2. Calculate hub score of each node using the formula 2.

3. Calculate authority score of each node using the formula 1.

4. Normalize authority & hub score so that the sum of authority score of every node and the sum of hub score of every node are one.

[4]We used the full set of U(B, n) to create the modifier graph for the full scale of experimentation in section 4.

5. Iterate from step 2 until the score of every node converges.

In the modifier graph, Authority score can be mapped to the intrinsic score of a node(token) as a head word, and Hub score can be mapped to the intrinsic score of a node(token) as a modifier.

**Scoring Category Link** Now, we can score the input category link. The score of category link <A, B> is given as follows:

$$Score(< A, B >) = Authority(h) + \sum_{a \text{ in } mod(A)} Hub(a) \quad (3)$$

Here, $Score(< A, B >)$ represents the final score of category link <A, B>, $h$ represents the head word of A, and $mod(A)$ represents the set of modifiers of A. Since the score of head word and modifiers are calculated based on the upper categories of B, this formula can integrate both meaning of A and B to classify whether the link is isa/instanceOf. Table 3 shows the scores of seven article-category links from table 2, calculated using the graph-based approach.

| Article-Category Links | Score |
|---|---|
| <Spacecraft escaping the Solar System, Pioneer 11> | 0.5972 |
| <1973 in space exploration, Pioneer 11> | 0.4018 |
| <Jupiter spacecraft, Pioneer 11> | 0.2105 |
| <Saturn spacecraft, Pioneer 11> | 0.2105 |
| <Inactive extraterrestrial probes, Pioneer 11>, | 0.0440 |
| <Radio frequency propagation, Pioneer 11> | 0.0440 |
| <Pioneer program, Pioneer 11> | 0.0132 |

Table 3: Scoring each category links using graph-based approach

The link <Spacecraft escaping the Solar System, Pioneer 11> gets the highest score, while the link <1973 in space exploration, Pioneer 11>, which got the highest score using counting-based approach, gets the second place. That proves the algorithm's effectiveness for distinguishing *isa/instanceOf* link from other non-*isa/instanceOf* links. But there is still a problem - although the first-ranked link is a *isa/instanceOf* link, the second-ranked is not, while the third and fourth-ranked links (<Jupiter spacecraft, Pioneer 11>, <Saturn spacecraft, Pioneer 11> are *isa/instanceOf* links. To get a better result, we propose four additional modifications in the next secton.

### 3.4 Additional Modifications to the Graph-based Approach

To better reflect the category structure and the property of category names to the scoring mechanism, the following four modifications can be made. Each of these modification could be applied independently to the original algorithm described in section 3.3.2.

**Authority Impact Factor (I).** In most cases, a category name contains only one head word, while it contains 2 or more modifiers. As Formula (3) is just the linear sum of the hub scores of each modifier and the authority score of the head word, the resultant score is more affected by hub score, because the number of modifiers is normally bigger than the number of head words. To balance the effect of hub score and authority score, we introduce authority impact factor *I*:

$$Score(< A, B >) = I \cdot Authority(h) + \sum_{a \text{ in } mod(A)} Hub(a) \quad (4)$$

The authority impact factor is defined as the average number of modifiers in the elements of U(B, n), since normally each category name contains only one head word.

**Dummy Node (D).** There are some category names that contain only one head word and no modifier, thus making it impossible to create the modifier graph.[5] Thus, for such category names we introduce dummy nodes to include their information into the modifier graph. In figure 3, you can observe the introduction of the dummy node 'dummy0'.

[5]For example, in figure 2, we cannot find node 'Saturn' while U(Pioneer 11, 3) contains category name 'Saturn'

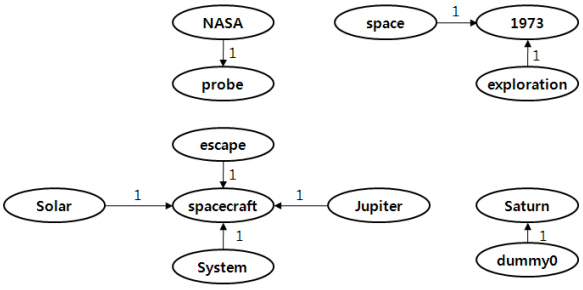


Figure 3: Modifier graph of the subset of U(Pioneer 11, 3), with dummy node.

**Category Distance Factor (C).** We define the category distance between category/article A and B as the minimum number of category links required to reach B from A by following the category links. Category distance factor *C* of a category name A from U(B, n) is the reverse of the category distance between A and B. We assumed that, if the distance between A and B is higher, then it is less probable for A to have the intrinsic property of B. Based on this assumption, category distance factor *C* of category name A is multiplied by the edge score of an edge generated by category name A.

Figure 4 shows the modifier graph of figure 2 that applies the category distance factor. Since the category distance between "Pioneer 11" and "NASA probe" is two, the score of edge (NASA, probe) is 1/2 = 0.5.

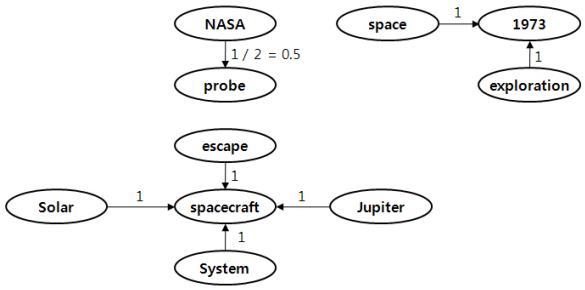


Figure 4: Modifier graph of the subset of U(Pioneer 11, 3), with category distance factor.

**Modifier Number Normalization Factor (W).** In the algorithm of building a modifier graph, the head word of a category name with many modifiers has the advantage over the head word of a category name with few modifiers, as if a category name contains n modifiers it will generate n edges incoming to its head word. To overcome this problem, we defined the modifier number normalization factor *W* for each category name: it is defined as the reverse of the number of modifiers in the category name, and it is multiplied by the edge score of an edge, generated by the category name, of the modifier graph. Figure 5 shows the modifier graph of figure 2 with the modifier number normalization factor. Since the category name "Spacecraft escaping the Solar System" has three modifiers, the scores of edge (escape, Pioneer 11), (solar, Pioneer 11) and (system, Pioneer 11) are 1/3 = 0.33.

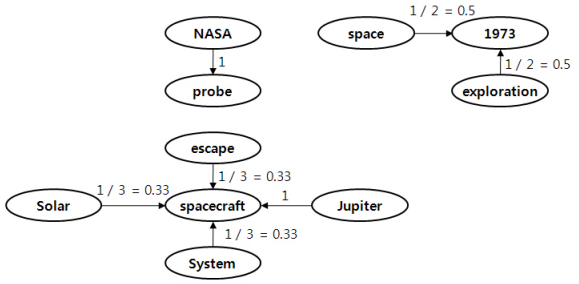


Figure 5: Modifier graph of the subset of U(Pioneer 11, 3), with modifier number normalization factor.

**Removing roleOf Relation (E).** To distinguish the roleOf relation from taxonomic relation,we introduce a new *E*. This feature simply classify the link <A, B> as non-*instanceOf* if category name A has endings like -er, -ers, -or, -ors, -ian, -ians. Since only the terminal node can represent the name of person in category structure, we applied this feature to classify only article-category links. One of the example from Wikipedia which should be judged as roleOf relation is <La Liga footballer, Cristiano Ronaldo>.

After applying above four modifications, we get the result in table 4. Now, top 3 links all represent *instanceOf* links.

| Article-Category Links | Score |
|---|---|
| <Spacecraft escaping the Solar System, Pioneer 11> | 2.1416 |
| <Jupiter spacecraft, Pioneer 11> | 2.1286 |
| <Saturn spacecraft, Pioneer 11> | 2.1286 |
| <1973 in space exploration, Pioneer 11> | 0.0241 |
| <Pioneer program, Pioneer 11> | 0.0062 |
| <Inactive extraterrestrial probes, Pioneer 11>, | 0.0026 |
| <Radio frequency propagation, Pioneer 11> | 0.0021 |

Table 4: Scoring each category links using graph-based approach with four modifications.

## 4 Implementation

We implemented a combinatory system that combines the algorithm suggested by this paper with existing lexical pattern-based algorithms. More precisely, we set two parameters $\alpha$ and $\beta$, in which $\beta$ has a consistently higher value than $\alpha$. If score of the given category link, which is retrieved by the proposed system, is higher than $\beta$, it is classified as *isa/instanceOf*. If the score is higher than $\alpha$ but lower or equal to $\beta$, the system uses an existing lexical pattern-based algorithm to classify the link. If the score is lower than or equal to $\alpha$, it is classified as not isa/instanceOf.

To test the system, we used Wikipedia's category structure, which contains 1,160,248 category-category links and 15,778,801 article-category links between 505,277 categories and 6,808,543 articles. We extract category links from the Wikipedia category structure and annotate them to construct the test corpus. During the process of choosing category links, we intentionally removed category links with names containing any of the following words: "stub", "wikiproject", "wikipedia", "template", "article", "start-class", "category", "redirect", "mediawiki", "user", "portal", "page", and "list". These words are normally used to represent Wikipedia maintenance pages. After we remove the links described before, we randomly choose 3,951 category-category links and 1,688 article-category links. Two annotators worked separately to annotate whether or not the given link is an *isa/instanceOf* link, and in the event of conflict they would discuss the case and make a final decision.

We carried out experiments on category-category link set and article-category link set separately, since their characteristics are different. We assumed that the taxonomic relation in a category-category link is an *isa* link, while the taxonomic relation in an article-category link is an *instanceOf* link. To acquire the upper category set, we set n=3 throughout the experiment. For head word extraction, the method of Collins (1999) is used, and for lemmatization we used the Lingpipe toolkit (Alias-i, 2008).

### 4.1 Experiments on category-category link

We divided the 3,951 category-category links into two equally-sized sets, and used one set as a training set and the other one as a test set. The training set was used to identify the $\alpha$ and $\beta$ values for *isa* link classification: in other words, the $\alpha$ and $\beta$ values that showed the best performance when applied to training set were selected as the actual parameters used by the system. As Wikipedia's category structure contains a huge number of category links, precision is more important than recall. As recall cannot be ignored, we chose the parameters that gave the highest precision on the training set, while giving a recall of at least 0.7. Also, we carried out experiments on three baseline systems.The first one determined every link as an *isa* link. The second one applied the head word matching rule (M) only, which says that for category-category link <A, B>, if the head words of A and B are the same, then <A, B> should be classified as an *isa* link. The third one applies the method of Ponzetto (P) (Ponzetto and Strube, 2007). The ruleset of Ponzetto includes Head word matching rule, Modifier-head word matching rule(Ex. <Crime, Crime Comics>: Head word of "Crime" and modifier of "Crime Comics" matches: Not *isa*), and the plurality rule used by YAGO system(Explained at the next chapter)).

Table 5 shows the baseline results, the results of existing systems, and our best results on the test set. Usage of authority score is represented as *A*, and usage of hub score is represented as *H*. Also,we did experiments on all possible combina-

tion of features A, H, I, D, C, W, M, P. For example, Comb(AHICDM) means that we used feature A, H, I, C, D to construct the modifier graph and score the category link, and for those whose score is between $\alpha$ and $\beta$ we used head word matching rule to classify them. At the table, P stands for Precision, R stands for Recall, and F stands for F-measure.

| Setting | P | R | F |
|---|---|---|---|
| **Baseline1** | 0.7277 | 1.0 | 0.8424 |
| **Baseline2(M)** | 0.9480 | 0.6335 | 0.7595 |
| **Baseline3(P)** | 0.9232 | 0.6516 | 0.7640 |
| **Comb1(AHM)** | 0.9223 | 0.7350 | 0.8181 |
| **Comb2(AHP)** | 0.8606 | 0.7211 | 0.7847 |
| **Comb3(AHICM)** | 0.9325 | 0.7302 | 0.8190 |

Table 5: Experimental result on test set of category-category links: Baseline vs. System best result

As you can observe, the precision of head-word matching (M) is high, meaning that in many cases the head word represents the intrinsic property. Also, its recall shows that for category-category links, at least more than half of the categories are categorized using the intrinsic property of the objects grouped within them, which strongly supports lemma 2 in section 3.2. The comparison of setting M and AHM, P and AHP shows that the intrinsic-property based approach increases recall of the existing system about 7-10 %, at the cost of of 2-6 % precision loss. This shows that, rather than looking only at the given category link and analyzing patterns on its name, by gathering information from the upper category set, we were able to significantly increase recall. However, it also shows that some "garbage" information is introduced through the upper category set, resulting in a 2-6 % precision loss. The best system shows about a 8-10 % increase in recall, with comparably good precision compared to the two baseline systems.

### 4.2 Experiments on article-category link

In a similar manner to the experiments on category-category links, we divided the 1,688 article-category links into two equally-sized sets, and used one set as a training set and the other one as a test set. The training set is used to determine the parameters for *instanceOf* link classification. The parameter setting procedure was the same as in the experiments on category-category links, except that we used the article-category links for the procedure. In this experiment, we also adapted three baseline systems. The first system classifies every link as an *instanceOf* link, the second system adapts the head word matching rule (M), and the third system applies the rule from Yago (Y) (Suchanek et al., 2007), which states that for article-category link <A, B>, if A is plural then the link could be classified as an *instanceOf* relation.

| Setting | P | R | F |
|---|---|---|---|
| **Baseline1** | 0.5261 | 1.0 | 0.6894 |
| **Baseline2(M)** | 0.7451 | 0.0856 | 0.1535 |
| **Baseline3(Y)** | 0.6036 | 0.5315 | 0.5653 |
| **Comb1(AHY)** | 0.6082 | 0.6718 | 0.6381 |
| **Comb2(ADWEY)** | 0.7581 | 0.7410 | 0.7494 |

Table 6: Experimental result on test set of article-category links on some settings

Table 6 shows the baseline results and the best results of the combinatory system. As you can observe from the above table, M (head word matching rule) does not work well in article-category links, although its precision is still high or comparable to that of other methods. Since in most cases an article represents one instance, in many cases they have their own name, making the recall of the head word matching rule extremely low. Also, the combination system 1 (AHY) shows comparable precision with Y but 14 % higher in reall, resulting 7 % increse in F-Measure.The best system shows about 18 % increase in F-measure, especially 15 % precision increase and 21 % recall increase compared to YAGO system.

## 5 Conclusion and Future work

In this paper, we explored a intrinsic token-based approach to the problem of classifying whether a category link is a taxonomic relation or not. Unlike previous works that classify category links, we acquired the definition of a lower category

name by extracting intrinsic tokens and using them to score the given category link, rather than by applying predefined lexical rules to the category link. Our intrinsic token-based approach leads to a significant improvement in F-measure compared to previous state-of-the-art systems. One possible future direction for research is automatic instance population, by using those extracted intrinsic tokens and gathering taxonomic relations from the category structure.

## Acknowledgments

This work was supported by the Industrial Strategic Technology Development Program (10035348, Development of a Cognitive Planning and Learning Model for Mobile Platforms) funded by the Ministry of Knowledge Economy(MKE, Korea).

## References

Soumen C. Byron, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. 1997. Using taxonomy, discriminants, and signatures for navigating in text databases. *Proceedings of the international conference on very large data bases*, 446–455.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339.

Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. 2005. Learning Taxonomic Relations from Heterogeneous Sources of Evidence. *Ontology Learning from Text: Methods, Evaluation and Applications*, 59–73.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the 14th conference on Computational linguistics*, 2:539–545.

Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. Ontologies improve text document clustering. *Proceedings of the IEEE International Conference on Data Mining*, 541–544.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632

Simone P. Ponzetto, and Michael Strube. 2007. Deriving a Large Scale Taxonomy from Wikipedia. *Proceedings of the AAAI07*.

Vivi Nastase, and Michael Strube. 2008. Decoding Wikipedia category names for knowledge acquisition . *Proceedings of the AAAI08*.

Riichiro Mizoguchi. 2004. Part 3: Advanced course of ontological engineering. *New Generation Computing*, 22(2): 193–220

Rada Mihalcea, and Paul Tarau. 2005. A Language Independent Algorithm for Single and Multiple Document Summarization. *Proceedings of IJCNLP 2005*.

Ian Niles, and Adam Pease. 2003. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. *Proceedings of the IEEE International Conference on Information and Knowledge Engineering*.

Soeren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. *Lecture Notes in Computer Science*, 4825/2007:722–735.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia *Proceedings of the 16th international conference on World Wide Web*, 697–706.

Michael Collins. 1999. Head-driven Statistical Models for Natural Language Parsing. *University of Pennsylvania PhD Thesis*.

Alias-i. 2008. LingPipe 3.9.1. *http://alias-i.com/lingpipe*.