# A Gauss Function Based Approach for Unbalanced Ontology Matching

Qian Zhong[1] Hanyu Li[2] Juanzi Li[1] Guotong Xie[2] Jie Tang[1] Lizhu Zhou[1] Yue Pan[2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084
{zhongqian, ljz, tangjie}@keg.cs.tsinghua.edu.cn, dcszlz@tsinghua.edu.cn

[2] IBM China Research Laboratory, Beijing 100193
{lihanyu, xieguot, panyue}@cn.ibm.com

## ABSTRACT

Ontology matching, aiming to obtain semantic correspondences between two ontologies, has played a key role in data exchange, data integration and metadata management. Among numerous matching scenarios, especially the applications cross multiple domains, we observe an important problem, denoted as unbalanced ontology matching which requires to find the matches between an ontology describing a local domain knowledge and another ontology covering the information over multiple domains, is not well studied in the community.

In this paper, we propose a novel Gauss Function based ontology matching approach to deal with this unbalanced ontology matching issue. Given a relative lightweight ontology which represents the local domain knowledge, we extract a "similar" sub-ontology from the corresponding heavyweight ontology and then carry out the matching procedure between this lightweight ontology and the newly generated sub-ontology. The sub-ontology generation is based on the influences between concepts in the heavyweight ontology. We propose a Gauss Function based method to properly calculate the influence values between concepts. In addition, we perform an extensive experiment to verify the effectiveness and efficiency of our proposed approach by using OAEI 2007 tasks. Experimental results clearly demonstrate that our solution outperforms the existing methods in terms of precision, recall and elapsed time.

## Categories and Subject Descriptors

D.2.12 [**Interoperability**]: Data mapping; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Semantic networks

## General Terms

Algorithms, Experimentation, Performance

## Keywords

Gauss Function, Ontology matching, Unbalance

## 1. INTRODUCTION

With the growing needs of information sharing, more and more ontologies are established and distributed by different enterprises and institutes to describe the knowledge for the domains of interest. As a consequence, effectively and efficiently integrating the semantics from diverse ontologies to achieve interoperability, especially in web scale, becomes an important task and attracts wide attentions of the researchers in community.

Much effort has been put into the design and development of the semantic integration systems, including [27, 19, 20, 10]. In these approaches, effectively finding the correspondences between concepts in different ontologies, also identified as ontology matching or alignment, plays the key role since it lays the cornerstone for the following query rewriting and query result merging in data integration applications [17, 28].

The process of ontology matching takes as input two ontologies and determines a set of relationships between concepts in the ontologies. Existing solutions utilize various techniques to attain satisfying matching results, such as name-based [14, 11], structure-based [26, 13, 15], instance-based [31, 16, 21], external knowledge-based [18, 9] and reasoning-based [30] methods. In addition, compound solutions which employ multiple techniques and aim to process various matching scenarios are proposed. Such solutions include COMA [15], RiMOM [29], H-Match [12] and Cupid [25].

However, we note that an important problem, unbalanced ontology matching, cannot be well processed by using the existing methods. Unbalanced ontology matching is prevalent in the data integration applications when people intend to merge data, exchange data and translate queries between one local ontology describing the knowledge of a specific domain and another global ontology which usually covers the information of multi-domains. Typical global ontologies are Cyc [23], FMA [9], and GEMET [1] which is the result of merging data from more than 40 domains.

Unbalanced ontology matching poses a great challenge on the existing approaches. For example, the huge number of concepts in the heavyweight global ontologies quickly deteriorate the performance of structure-based approaches which usually utilize in-memory structures to accomplish matching

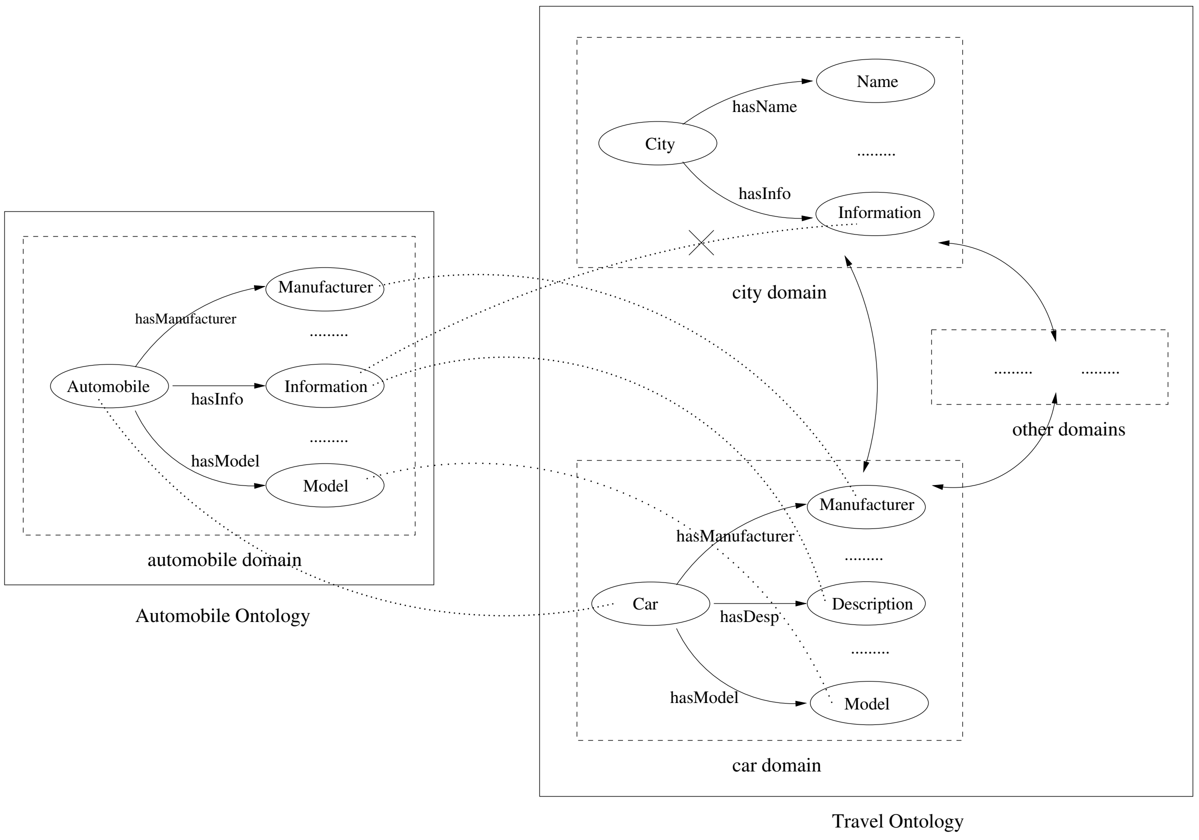


**Figure 1: Example of An Unbalanced Ontology Matching**

tasks. In addition, the heavyweight ontologies may contain a lot of noisy concepts which obstruct building correct correspondences between concepts.

Figure 1 shows an example of an unbalanced ontology matching. A local ontology, *Automobile*, introduces the information in *automobile* domain and contains concepts like *Automobile*, $Manufacturer$ and *Model*, etc. In contrast, ontology $Travel$ provides the data covering multiple domains, such as *car* and *city*. The matching task requires us to find the semantically equivalent concepts (correspondences) between two ontologies.

The simple way to obtain the correspondences in ontology *Automobile* and $Travel$ is to employ a name-based solution, for example, edit-distance to compare the distances between the strings of concepts in the ontologies. However, since edit-distance does not take the semantics other than strings into account, an error like "$Travel.Information$[1] matches $Automobile.Information$ very well" would be produced while the correct matching of $Automobile.Information$ obviously is $Travel.Description$.

The structure-based approaches use the ontology structures to reflect the relationships between concepts. For example, the similarity flooding approach [26] propagates the similarities between concepts to refine the matching results. It fixes the error match above due to the similarity propagation from the neighbor concepts. Concretely speaking, the high similarity score between *Automobile.Automobile* and $Travel.Car$ can enhance the similarity between concepts $Automobile.Information$ and $Travel.Description$. Similarly, $Travel.Information$ is deleted from the matching result of $Automobile.Information$ because of the low similarity between their neighbor concepts.

However, the structure-based solutions share a resource-consuming problem. To propagate the similarities, similarity flooding [26] builds an in-memory graph in which the nodes contain pairs of concepts from the ontologies involved in the matching and iteratively updates the similarities of the nodes. This may lead to the memory overflow when processing the large-size ontologies. In our experiments, a matching task involving an ontology with more than 28,000 concepts cannot be successfully finished by using this method.

In this paper, we propose a novel approach to address the issue of unbalanced ontology matching. Our solution is motivated by the observation that given ontology *Automobile*, the matching concepts in $Travel$ form a relatively small field (domain *car*) which describes the similar information as *Automobile* does. As a result, if such a small field could be successfully extracted from ontology $Travel$, then we can carry out the fine-grained solutions, such as similarity flooding [26], on this small field to gain better performance.

Concretely speaking, we first apply a simple yet fast matching method, e.g., a name-based approach, to find in the global ontology a set of concepts which possibly correspond to the concepts in the local ontology. Based on the concept set obtained, a sub-ontology which is maximally "similar" to

[1]The first and second parts denote the ontology and concept respectively.

the local ontology is extracted from the global one. Finally, a fine-grained matching method is carried out to obtain the final results.

In the procedure above, our proposed approach utilizes Gauss Function to calculate the relevance of one concept in the global ontology to the local ontology, and then determines whether this concept will occur in the later sub-ontology construction. Since the size of constructed sub-ontology (the number of concepts in ontology) is much less than that of the global ontology, our approach greatly improves the performance of unbalanced ontology matching in terms of precision, recall, and elapsed time. Experimental results based on datasets of OAEI [2] 2007 environment tasks clearly demonstrate it.

The rest of this paper is organized as follows. Section 2 gives the background knowledge. Section 3 describes our proposed method for unbalanced ontology matching. Section 4 presents the experimental results. Finally, we discuss related work in Section 5 and conclude in Section 6.

## 2. PROBLEM DEFINITION

This part defines the problems related to ontology and ontology matching used in the paper.

### 2.1 Ontology

An ontology usually provides a set of vocabularies to describe the information of interest. The major components of ontologies are concepts, relations, instances and axioms [29]. They are explained next respectively.

1. Concepts. A concept represents a set of entities or "things" within a domain. Concept is the core of ontology and a hierarchical structure could be used to organize concepts.

2. Relations. A relation describes the interaction between concepts. It is also called the property of a concept. Relations can be classified into two types: taxonomies that organize concepts in super or sub-concept hierarchy, such as $rdfs : subClassOf$, and associative relations that relate concepts beyond the hierarchy, for example, property $rdfs : seeAlso$. Like concepts, relations could be organized in a hierarchical structure.

3. Instances. Instances are the "things" represented by the concepts.

4. Axioms. Axioms are assertions in form of logic to constrain values for classes (concepts) or instances. For example, given concepts *teacher* and *course*, and relation *teaches* between them, an axiom may assert that one teacher instance must teach at least one course.

Among the components above, concepts, relations and axioms compose the schemas of ontologies. In this paper, we only take the matches between the concepts into account when executing ontology matching as many other approaches did. In addition, it is easy to understand that an ontology (schema) could be viewed as a directed graph where concepts and relations represent the vertexes and edges respectively.

**Example 1:** Consider the example shown in Figure 1. Concepts $Automobile.Automobile$ and $Automobile.Model$ are the vertexes in the graph representing ontology $Automobile$. In addition, relation $Automobile.hasModel$ between these two concepts is the edge connecting the corresponding vertexes. □

### 2.2 Ontology Matching

Given a source ontology $O_1$, a target ontology $O_2$, and a concept $c_i$ in $O_1$, we call the procedure to find the semantically equivalent concepts $\{c_j\}$ in $O_2$ to $c_i$ ontology matching, denoted as $M$. Formally, ontology matching $M$ is represented as

$$M(c_i, O_1, O_2) = \{c_j\}$$

Furthermore, $M$ could be extended to find the matches of a set of concepts $\{c_i\}$, which is denoted as

$$M(\{c_i\}, O_1, O_2) = \{c_j\}$$

or

$$M(O_1, O_2) = \{c_j\}$$

for short if $\{c_i\}$ contain all concepts in $O_1$.

In this paper, we focus on addressing the unbalanced ontology matching issue. That is, the source is a relatively lightweight ontology and the target is a heavyweight one. In the rest of the paper, we use $O_l$ and $O_h$ to denote the source and target ontologies respectively.

**Example 2:** Given concepts $Automobile.Information$ and $Automobile.Model$ of lightweight ontology $Automobile$ in Figure 1, the matching results of them in heavyweight ontology $Travel$ are $Travel.Description$ and $Travel.Model$ respectively. □

## 3. GAUSS FUNCTION BASED APPROACH

This part introduces our proposed approach. We first explain the outline and then discuss the details of the solution.

### 3.1 Approach Overview

Algorithm 1 shows the sketch of the solution. Given a lightweight ontology $O_l$ and a heavyweight ontology $O_h$, we utilize a simple measure method to quickly calculate the similarities between $O_h$ concepts and ontology $O_l$. Those concepts in $O_h$ with a high similarity value to $O_l$ would be put into a candidate concept set. After that, the relevances of these concepts to $O_l$ are calculated and a sub-ontology $O_s$ is correspondingly constructed from $O_h$. Finally, a fine-grained method is performed to find the matching result $M(O_l, O_s)$ and we output it as $M(O_l, O_h)$.

```
Algorithm 1 Ontology Matching
  Input: Lightweight ontology $O_l$, heavyweight ontology $O_h$.
  Output: Matching result $M(O_l, O_h)$.
    1. Select candidate concepts from $O_h$ based on the similarities between $O_h$ concepts and $O_l$.
    2. Construct a sub-ontology $O_s$ from $O_h$ according to the relevances of concepts to $O_l$.
    3. Find the matching result $M(O_l, O_s)$ between $O_l$ and $O_s$ and output it as $M(O_l, O_h)$.
```

## 3.2 Selecting Concepts from Heavyweight Ontology

Concept selection from $O_h$ are based on the similarities between concepts in $O_l$ and $O_h$ respectively. The similarity calculation process is straightforward (see Algorithm 2). A nested loop is carried out to obtain the similarity value $s_{ij}$ between the concepts $c_i$ and $c_j$, such that $c_i$ and $c_j$ are from $O_h$ and $O_l$ respectively. Next, all $s_{ij}$ for $c_i$ are summarized as similarity $s_i$ between $c_i$ and $O_l$ if $s_{ij}$ is greater than a threshold $\alpha$ (Line 9, Algorithm 2). The purpose of this step is to obtain the similarity between $c_i$ and $O_l$. Finally, concept $c_i$ is inserted into candidate set $C$ based on the comparison between $s_i$ and a threshold $\beta$.

**Algorithm 2** Selecting Candidate Concepts from $O_h$

```
 1: Input: Lightweight ontology $O_l$, heavyweight ontology
    $O_h$.
 2: Output: Concept set $C = \{c_i\}$, $c_i$ is the concept in $O_h$.
 3:
 4: let $C$ be empty
 5: for all concept $c_i$ in $O_h$ do
 6:   for all concept $c_j$ in $O_l$ do
 7:     calculate similarity $s_{ij}$ between $c_i$ and $c_j$
 8:     if $s_{ij} > \alpha$ then
 9:       $s_i = s_i + s_{ij}$
10:     end if
11:   end for
12:   if $s_i > \beta$ then
13:     add $c_i$ to $C$
14:   end if
15: end for
```

When calculating similarity $s_{ij}$ between $c_i$ and $c_j$ (Line 7 in Algorithm 2), any matching method can be adopted if it satisfies the requirement of quick computation. Among a large number of candidates, the name-based approaches are the simplest and most common ones which compare the strings, such as names, labels or comments of the concepts. In this paper, we employ edit-distance and WordNet-based methods in the step. We then simply introduce these two techniques and interested readers can refer to [24] for more details.

**Edit-Distance.** Given two words (strings) $w_i$ and $w_j$, the edit-distance between them is defined as

$$edit\_distance(w_i, w_j) = \frac{|\{op_k\}|}{\max(length(w_i), length(w_j))}$$

where $|\{op_k\}|$ denotes a series of operations required to convert $w_i$ to $w_j$ (typical operations include character insertion, update and deletion), and $length(w_i)$ is the number of characters in $w_i$. As a consequence, the edit-distance based similarity is given as

$$s_{edit} = \frac{1}{1 + edit\_distance(w_i, w_j)}$$

Consider two words "site" and "cite". The edit-distance between them is 0.25 since we can simply replace "s" with "c" (one operation). Then the corresponding similarity is computed as $1/(1 + 0.25) = 0.8$.

**WordNet.** The WordNet-based similarity value between two words (strings) are defined as follows.

$$s_{wordnet}(w_i, w_j) = \frac{2 \times \log p(s)}{\log p(s_i) + \log p(s_j)}$$

In the formula above, $s_i$ and $s_j$ represent the corresponding nodes of words $w_i$ and $w_j$ in the WordNet semantic tree. And $s$ denotes the first common ancestor node of $s_i$ and $s_j$. In addition, $p(s_i)$ is computed as $count(s_i)/total$ where $count(s_i)$ is the number of nodes in the subtree rooted at $s_i$ and $total$ denotes the total number of nodes in the entire semantic tree.

For example, to calculate the WordNet-based similarity between "hill" and "coast", we find the most specific node "geological-information" that subsumes both "hill" and "coast" in the WordNet taxonomy tree. And then we have $p(hill) = 0.0000189$, $p(geological\_info) = 0.00176$ and $p(coast) = 0.0000216$. Finally, the similarity is equal to 0.59 by following the given formula.

In Algorithm 2, we calculate both WordNet-based and edit-distance similarities. If one of these two values is equal to 1, the similarity between two concepts ($s_{ij}$) is then set to be 1 since one method is very confident about the equivalence of concepts (strings). Otherwise, the average value is adopted.

| Automobile / Travel | Auto | Manufacturer | Info | Model |
|---|---|---|---|---|
| Car | 1 | 0.27 | 0.05 | 0.24 |
| Manufacturer | 0.27 | 1 | 0.29 | 0.31 |
| Description | 0.05 | 0.04 | 0.45 | 0.19 |
| Model | 0.37 | 0.31 | 0.24 | 1 |
| City | 0.26 | 0.23 | 0.23 | 0.22 |
| Name | 0.31 | 0.31 | 0.26 | 0.34 |
| Information | 0.14 | 0.23 | 1 | 0.24 |

**Table 1: Similarity Matrix between Concepts in** *Automobile* **and** *Travel*

**Example 1:** This example explains how we select candidate concepts from ontology $Travel$ when matching $Travel$ with $Automobile$ in Figure 1. We first calculate the similarities between any two concepts by using the methods introduced from ontology $Automobile$ and $Travel$. The similarity matrix is shown in Table 1.

After that, the similarities between the concepts in $Travel$ and $O_{Automobile}$ are computed. Let threshold $\alpha$ be 0.3 (Line 8 in Algorithm 2). The summarized results are shown below.

$similarity(Car, O_{Automobile}) = 1$
$similarity(Manufacturer, O_{Automobile}) = 1.31$
$similarity(Description, O_{Automobile}) = 0.45$
$similarity(Model, O_{Automobile}) = 1.68$
$similarity(City, O_{Automobile}) = 0$
$similarity(Name, O_{Automobile}) = 0.96$
$similarity(Information, O_{Automobile}) = 1$

Assume threshold $\beta$ is 0.4. Then we get the candidate set $C$ which contains all the concepts above except $Travel.City$ [2]. □

[2]In our implementation of the approach, each calculated similarity $s_i$ is divided by the maximum of them to normalize these similarities to the range [0,1]. This is to allow using fixed threshold values over different ontologies.

## 3.3 Constructing Sub-ontologies

This part describes the procedure of constructing from $O_h$ the sub-ontology which is semantically "similar" to $O_l$. The basic idea is that for each concept in set $C$ ($C$ is generated in Algorithm 2), a relevance value which reflects how much the concept is related to $O_l$ is computed. The calculated results then determine whether a concept will participate in the later construction of sub-ontology $O_s$.

We observe two factors impact the relevance value of a concept to $O_l$. It is easy to understand that the first one is the similarity between this concept and $O_l$. In Example 1, $Travel.Model$ is possibly the most relevant concept to ontology *Automobile* since the similarity value between them is the highest.

The second factor, influence, is how much the neighbors of this concept are related to $O_l$ and how they propagate their similarities to this concept. Consider Example 1 again. $Travel.Description$ is not as relevant as $Travel.Information$ to ontology *Automobile* if only the similarity values are taken into account, since the former is 0.45 and the latter is 1. However, $Travel.Description$ neighbors $Trave.Car$, $Travel.Manufacturer$ and $Travel.Model$ are highly related to ontology *Automobile* (based on the similarities). Since these neighbors are very close to $Travel.Description$ in the graph, it is reasonable to deduce that $Travel.Description$ is potentially relevant to ontology *Automobile*.

Precisely describing the influences between concepts is a tough task. One observation is that the influence effects must decrease with the increasing distance between concepts. Consider the fact that Gauss Function satisfies this requirement and it is widely used, such in statistics to describe the normal distributions, in signal processing and in physics to serve for depicting the interactions between atoms. We use Gauss Function to simulate the influence values between concepts in ontologies.

### *3.3.1 Gaussian Function and Influence Computation*

A Gaussian Function is a function of the form

$$f(x) = ae^{-(\frac{x-b}{\sigma})^2}$$

where $a$, $b$ and $\sigma$ denote some real constants.

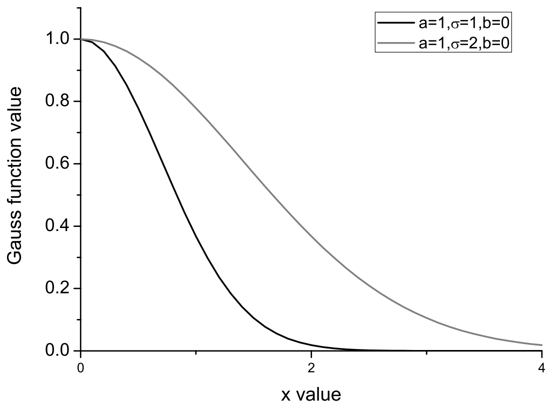


**Figure 2: Gauss Function**

Figure 2 shows an example of Gauss Function with different $\sigma$ values. Note that when $|x-b|$ is zero, $f(x)$ has a maximum value. And when $|x-b| \to \infty$, $f(x) \to 0$.

Following Gauss Function, we define the influence of concept $c_j$ to $c_i$, denoted by $\varphi_j(c_i)$, as follows:

$$\varphi_j(c_i) = s_j \times e^{-(|c_i - c_j|)^2} \ (i \neq j)$$

In the formula above, $s_j$ denotes the similarity of $c_j$ to an ontology (in our approach, $O_l$), and $|c_i - c_j|$ is the distance between $c_i$ and $c_j$, which is equivalent to the distance between their corresponding vertexes in the ontology graph, that is, the number of edges in the shortest path connecting two vertexes[3]. $\sigma$ in Gauss Function is simply evaluated as 1. This formula shows the influence from $c_j$ to $c_i$, and it can be observed that the influence value $\varphi_j(c_i)$ decreases with the increasing distance.

### *3.3.2 Collecting Relevant Concepts*

This part presents the method to collect relevant concepts from concept set $C$ for constructing the sub-ontology. The collection standard is based on the relevances of concepts to ontology $O_l$. Recall the relevance of a concept is determined by two factors, similarity value of the concept and influences from its neighbors. We define the relevance $r_i$ of concept $c_i$ in ontology $O_h$ to $O_l$ as

$$r_i = s_i \times \varphi(c_i)$$

while $\varphi(c_i)$ denotes the summary of $\varphi_j(c_i)$

$$\varphi(c_i) = \Sigma \varphi_j(c_i)$$

**Algorithm 3** Collecting Relevant Concepts

```
1: Input: Concept set $C$, ontology $O_h$, ontology $O_l$.
2: Output: Updated set $C$.
3:
4: for all $c_i$ in $C$ do
5:     calculate relevance $r_i$ of $c_i$ ($|c_i - c_j| \leq 2$) .
6:     if $r_i < \gamma$ then
7:         remove $c_i$ from $C$.
8:     end if
9: end for
```

Algorithm 3 demonstrates the process to collect the relevant concepts which contribute to the later sub-ontology construction. For each $c_i$ in $C$, we calculate its relevance $r_i$, and remove $c_i$ from $C$ if its $r_i$ is small ($r_i < \gamma$, $\gamma$ is a threshold) which indicates the concept is lowly relevant to $O_l$. In this step, only the near neighbor concepts (distances $\leq 2$) are considered for computing the influence values. This is because the value of $e^{-9}$ (distance $= 3$) in Gauss Function is less than $1.23 * 10^{-4}$, and ignoring those concepts with long distances will not affect the accuracy of relevance $r_i$.

**Example 2:** Continue the results in Example 1. We calculate the relevance values of all concepts in $C$. The results (after normalization) are shown below.

$relevance_{Car} = 1$
$relevance_{Manufacturer} = 0.42$
$relevance_{Description} = 0.15$
$relevance_{Model} = 0.53$

[3]In case that the vertexes are distributed in unconnected graphs, the distance is $+\infty$

$relevance_{Name} = 0.02$
$relevance_{Information} = 0.02$

Let $\gamma$ be 0.05 (Line 6 in Algorithm 3). Then concepts $Travel.Name$ and $Travel.Information$ are removed from set $C$. □

### 3.3.3 Constructing Sub-ontology

Based on the concepts remained in set $C$, we construct an ontology $O_s$ which is "similar" to $O_l$ in semantics.

**Algorithm 4** Constructing Sub-ontology $O_s$

```
1: Input: Ontology $O_h$, ontology $O_l$, concept set $C$
2: Output: Sub-ontology $O_s$ of $O_h$
3:
4: generate a set of connected sub-graphs $\{g_i\}$ from $O_h$.
5: calculate average vertex degree $d_l$ of ontology $O_l$.
6: for all $g_i$ do
7:    calculate average degree $d_i$ of $g_i$.
8:    if $|d_l - d_i| > \tau$ then
9:       remove $g_i$.
10:   end if
11: end for
12: output remaining sub-graphs as ontology $O_s$.
```

Algorithm 4 explains the procedure of the construction. We generate a set of connected sub-graphs $\{g_i\}$ from ontology $O_h$. Each $g_i$ must satisfy the following restrictions:

- All vertexes in $g_i$ are the concepts in $C$.
- All edges in $g_i$ are relations from $O_h$ connecting concepts in $C$.
- $g_i$ is as big as possible (contains as many vertexes and edges as possible).

After obtaining the sub-graphs from $O_h$, we calculate the average vertex degree of $O_l$ and compare it with that of each $g_i$[4]. Those $g_i$ whose degrees are greatly different from that of $O_l$ are deleted, since they are not the matching results due to the dissimilar structures. Finally, we merge all remaining sub-graphs and output the result as sub-ontology $O_s$.

**Example 3:** We continue to process set $C$ in Example 2 and build a sub-ontology which is equivalent to the graph shown in *car* domain of Figure 1. □

## 3.4 Finding Matching Results

After successfully constructing sub-ontology $O_s$, we then carry out a fine-grained method to discover the matches between $O_l$ and $O_s$. Since the size of $O_s$ is much smaller than that of $O_h$, any relatively accurate (even resource-consuming) techniques, such as the structure-based solutions, can be used for the best matching results.

In Example 4 and our experiments, we employ similarity flooding [26] in this step. Similarity flooding [26] constructs an in-memory graph and utilizes iterative similarity computations to judge the correspondences. Follows we give a sketch of the method. More details could be found in [26].

[4]Given a graph $G = (V, E)$, average vertex degree is defined as $|E|/|V|$ while $|E|$ and $|V|$ denote the number of edges and vertexes in $G$ respectively.

1. Given two ontologies $O_1$ and $O_2$, build a directed similarity graph $G$ in which vertexes contain pairs of concepts in $O_1$ and $O_2$. In addition, if both concepts in one vertex have the same relation (in two ontologies respectively) with the concepts in another vertex, an edge is then constructed between the two vertexes.

2. Assign a weight $w_{ij}$ to each edge $< v_i, v_j >$ in $G$. The value of $w_{ij}$ is set to be $1/n$ where $n$ is the out-degree of head vertex $v_i$.

3. Associate a similarity $s_i^0$ to each vertex $v_i$, which can be calculated by using other approaches, e.g., the name-based methods.

4. Compute $s_i^{n+1}$ for each vertex $v_i$ with the following formula

$$s_i^{n+1} = s_i^n + \sum_j s_j^n \times w_{ij} + \sum_j s_j^n \times w_{ji}$$

5. Repeat step 4, until the difference between $s_i^n$ and $s_i^{n+1}$ is less than a given threshold.

After obtaining the similarities between $O_l$ and $O_s$ by using similarity flooding, we output the results as $M(O_l, O_h)$.

**Example 4:** We finally obtain the results as follows after carrying out similarity flooding.

$M(Auto.Automobile, O_l, O_h) = \{Travel.Car\}$
$M(Auto.Manufacturer, O_l, O_h) = \{Travel.Manufacturer\}$
$M(Auto.Information, O_l, O_h) = \{Travel.Description\}$
$M(Auto.Model, O_l, O_h) = \{Travel.Model\}$

□

## 4. EXPERIMENTS

We present the details of experiments in this part.

## 4.1 Experiment Setup

We implement all solutions in Java and experiments are performed on a PC with AMD Athlon 4000+ dual core CPU(2.10GHz) and 2GB RAM. The operating system is Windows XP.

### 4.1.1 Datasets

We utilize OAEI [2] 2007 campaign datasets to perform our experiments. The three real-world ontologies contained in the datasets are listed below.

1. **GEMET:** The European Environment Agency GEMET ontology. It is a multi-language ontology. It involves more than 40 themes, such as agriculture, air, biology, climate, disasters, etc. Details could be found in [1, 3].

2. **AGROVOC:** AGROVOC thesaurus provided by Food and Agriculture Organization of the United Nations. It is a multilingual, structured and controlled vocabulary designed to cover the terminologies of all subject fields in agriculture, forestry, fisheries, food and related domains [4, 5].

| Ontology | ♯Concepts | Description Languages | Average Degree |
|---|---|---|---|
| GEMET | 5280 | bg. cs. da. de. el. en. en-US. es. et. eu. fi. fr. hu. it. nl. no. pl. ru. sk. sl. sv. | 8.09 |
| AGROVOC | 28439 | ar. cs. de. en. es. fr. hu. ja. pt. sk. th. zh. | 2.98 |
| NAL | 42326 | en. | 2.51 |

**Table 2: Characteristics of Datasets**

3. **NAL:** The Agricultural thesaurus released by the National Agricultural Library [6, 7]. It is an online vocabulary tool of agricultural terms in English and contains many agriculture related domains, including animals, livestock, economics, food, forest, etc.

Table 2 shows the characteristics of these ontologies.

### *4.1.2 Performance Metrics*

We use *precision*, *recall*, *F1-Measure* and *elapsed time* to measure the performance of our proposed solution. They are defined next.

*Precision* (P). It is the percentage of the correct discovered matches in all discovered matches.

*Recall* (R). It is the percentage of the correct discovered matches in all correct matches.

*F1-Measure* (F1). F1 considers the overall result of precision and recall.

$$F1 = \frac{2}{(1/R + 1/P)}$$

*Elapsed Time* (T). It is the total runtime of a task.

### *4.1.3 Workload*

The OAEI 2007 environment task organizers also provide three sets of reference alignment samples as well as official "correct" matching results, GEMET-AGROVOC (correspondences between the concepts in these two ontologies), GEMET-NAL and NAL-AGROVOC. These samples are classified into different domains and serve for the various purposes, e.g., evaluating precision or recall[5]. Interested readers can download these reference alignment samples from [8].

Based on these samples, we build our matching tasks as shown in Table 3 (the first two columns). The task names are composed of original ontology names, task purposes and domain names of the concepts. For example, task *ga_p_chem* is to discover the matches between GEMET and AGROVOC, and it aims to test precision using the concepts in chemistry domain.

For each task in Table 3, we create a lightweight source ontology from the first ontology involved and use the second heavyweight ontology as the target. For example, a lightweight source ontology is constructed from ontology GEMET for task *ga_p_chem* in Table 3(a). The sources generated contain all necessary concepts and relations. Particularly, we extract from GEMET (for GEMET-AGROVOC and GEMET-NAL related tasks) and NAL the concepts occurred in the corresponding reference alignment samples and all their super and sub-class concepts. In addition, the relations connecting these concepts are reserved, such as *rdfs:subClassOf* and *rdfs:seeAlso*. The characteristics of generated source ontologies are given in the last two columns of Table 3.

[5]OAEI 2007 introduces that some domain experts are responsible for finding the "correct" matching results. In addition, OAEI 2007 environment tasks include narrow matches, broad matches and exact matches while we only use exact match samples. This is because we define the ontology matching as "find the semantically equivalent concepts" in Section 2.2.

| Task | ♯Matches | Lightweight Source | |
|---|---|---|---|
| | | ♯Concepts | Average Degree |
| ga_p_chem | 14 | 46 | 4.67 |
| ga_p_geo | 23 | 43 | 5.51 |
| ga_p_misc | 28 | 89 | 4.28 |
| ga_p_tax | 21 | 47 | 5.30 |
| ga_p_nat | 35 | 88 | 5.30 |
| ga_p_risk | 21 | 63 | 4.52 |
| ga_r_agri | 61 | 179 | 6.57 |
| ga_r_geo | 87 | 172 | 6.47 |

(a)GEMET-AGROVOC

| Task | ♯Matches | Lightweight Source | |
|---|---|---|---|
| | | ♯Concepts | Average Degree |
| gn_p_chem | 30 | 82 | 5.43 |
| gn_p_geo | 17 | 40 | 5 |
| gn_p_misc | 29 | 107 | 4.43 |
| gn_p_tax | 15 | 33 | 5.09 |
| gn_p_nat | 23 | 67 | 4.87 |
| gn_p_risk | 30 | 95 | 4.98 |
| gn_r_agri | 61 | 182 | 6.55 |
| gn_r_geo | 77 | 172 | 6.51 |

(b)GEMET-NAL

| Task | ♯Matches | Lightweight Source | |
|---|---|---|---|
| | | ♯Concepts | Average Degree |
| na_p_chem | 141 | 283 | 1.92 |
| na_p_geo | 58 | 117 | 2.03 |
| na_p_misc | 231 | 575 | 1.80 |
| na_p_tax | 10 | 17 | 2 |
| na_r_anim | 10 | 39 | 2.36 |
| na_r_rod | 24 | 46 | 2.22 |
| na_r_oaks | 38 | 41 | 1.95 |
| na_r_eur | 62 | 71 | 3.44 |
| na_r_geo | 58 | 101 | 2.14 |

(c)NAL-AGROVOC

**Table 3: Workload**

### *4.1.4 Approaches*

We implement the following approaches which employ the name or structure based matching solutions.

**Name.** This is the method discussed in Section 3.2. We observe that most vocabularies occurred in the ontologies are the technical terms which are not covered by the WordNet semantic tree. As a result, only edit-distance is used in our experiments when calculating similarities between strings.

**SimFlood.** As mentioned before, similarity flooding [26] cannot be directly applied on the heavyweight ontologies due to the memory constraints. Instead, we implement a simplified similarity flooding (*SimFlood*). *SimFlood* calculates the similarities between concepts first using *Name*, and then propagates the similarities in an order of similarity values (from high to low). *SimFlood* propagates similarities only once, that is, only calculates $s^1$ in the step 4 of similarity flooding (Section 3.4). This thus avoids building an entire in-memory graph.

**Name-Gauss-SimFlood.** This is our proposed solution, where we adopt *Name* and *SimFlood* in the first and third steps in Algorithm 1 respectively.

**Name-Gauss-Flood.** This approach is similar to *Name-Gauss-SimFlood* except that we use entire similarity flooding algorithm[26] in the last step of Algorithm 1.

## 4.2 Effects of Constructing Sub-Ontologies

This part explores the effectiveness and efficiency of constructing sub-ontologies. We use *Name-Gauss-Flood* approach.

| Task | Size($O_h$) | Size($O_s$) | Ratio(Size($O_s$)/Size($O_h$)) |
|---|---|---|---|
| ga_p_chem | 28439 | 276 | 0.010 |
| ga_p_geo | | 268 | 0.009 |
| ga_p_misc | | 520 | 0.018 |
| ga_p_tax | | 302 | 0.011 |
| ga_p_nat | | 519 | 0.018 |
| ga_p_risk | | 399 | 0.014 |
| ga_r_agri | | 1039 | 0.037 |
| ga_r_geo | | 976 | 0.034 |
| **Average** | | **537** | **0.019** |

(a)GEMET-AGROVOC

| Task | Size($O_h$) | Size($O_s$) | Ratio(Size($O_s$)/Size($O_h$)) |
|---|---|---|---|
| gn_p_chem | 42326 | 1027 | 0.024 |
| gn_p_geo | | 530 | 0.013 |
| gn_p_misc | | 1248 | 0.029 |
| gn_p_tax | | 445 | 0.011 |
| gn_p_nat | | 829 | 0.020 |
| gn_p_risk | | 1117 | 0.026 |
| gn_r_agri | | 2074 | 0.049 |
| gn_r_geo | | 2059 | 0.049 |
| **Average** | | **1166** | **0.028** |

(b)GEMET-NAL

| Task | Size($O_h$) | Size($O_s$) | Ratio(Size($O_s$)/Size($O_h$)) |
|---|---|---|---|
| na_p_chem | 28439 | 1576 | 0.055 |
| na_p_geo | | 662 | 0.023 |
| na_p_misc | | 3021 | 0.106 |
| na_p_tax | | 103 | 0.004 |
| na_r_anim | | 264 | 0.009 |
| na_r_rod | | 313 | 0.011 |
| na_r_oaks | | 270 | 0.009 |
| na_r_eur | | 406 | 0.014 |
| na_r_geo | | 540 | 0.019 |
| **Average** | | **795** | **0.028** |

(c)NAL-AGROVOC

**Table 4: Sub-ontology Sizes**

Table 4 compares the sizes of sub-ontology $O_s$ and target ontology $O_h$. It can be found that the sizes of $O_s$ are much smaller than that of $O_h$. For example, in task *ga_p_chem* (the first in Table 4), there are 276 concepts remained in sub-ontology $O_s$. Compared with 28439 concepts in $O_h$, more than 99% concepts in $O_h$ are successfully removed during the construction of sub-ontology. In addition, the average ratios of sub-ontology size to target ontology size for three sets are 0.019, 0.028 and 0.028 respectively. This clearly reveals the effectiveness of constructing sub-ontologies.

| Task | Total | Step1 | Step2 | Step3 | Step2/Total |
|---|---|---|---|---|---|
| ga_p_chem | 321 | 287 | 1.36 | 33 | 0.0042 |
| ga_p_geo | 278 | 254 | 1.07 | 23 | 0.0038 |
| ga_p_misc | 628 | 509 | 4.96 | 114 | 0.0079 |
| ga_p_tax | 357 | 312 | 2.19 | 43 | 0.0061 |
| ga_p_nat | 767 | 618 | 7.77 | 141 | 0.0101 |
| ga_p_risk | 549 | 474 | 3.02 | 72 | 0.0055 |
| ga_r_agri | 1285 | 836 | 24.55 | 424 | 0.0191 |
| ga_r_geo | 1129 | 811 | 15.70 | 302 | 0.0139 |
| **Average** | **664** | **513** | **7.57** | **144** | **0.0088** |

(a)GEMET-AGROVOC

| Task | Total | Step1 | Step2 | Step3 | Step2/Total |
|---|---|---|---|---|---|
| gn_p_chem | 711 | 699 | 0.49 | 12 | 0.0007 |
| gn_p_geo | 346 | 341 | 0.4 | 5 | 0.0012 |
| gn_p_misc | 915 | 891 | 0.54 | 23 | 0.0006 |
| gn_p_tax | 227 | 225 | 0.40 | 2 | 0.0017 |
| gn_p_nat | 618 | 609 | 0.458 | 9 | 0.0007 |
| gn_p_risk | 792 | 774 | 0.569 | 17 | 0.0007 |
| gn_r_agri | 1549 | 1457 | 0.876 | 91 | 0.0006 |
| gn_r_geo | 1339 | 1270 | 0.743 | 68 | 0.0006 |
| **Average** | **812** | **783** | **0.56** | **28** | **0.0009** |

(b)GEMET-NAL

| Task | Total | Step1 | Step2 | Step3 | Step2/Total |
|---|---|---|---|---|---|
| na_p_chem | 2183 | 1941 | 40.48 | 202 | 0.0185 |
| na_p_geo | 1267 | 1216 | 7.82 | 43 | 0.0062 |
| na_p_misc | 5491 | 4522 | 140.45 | 829 | 0.0256 |
| na_p_tax | 148 | 147 | 0.37 | 1 | 0.0025 |
| na_r_anim | 358 | 351 | 1.12 | 6 | 0.0031 |
| na_r_rod | 249 | 245 | 0.64 | 3 | 0.0026 |
| na_r_oaks | 428 | 421 | 0.95 | 6 | 0.0022 |
| na_r_eur | 1064 | 1035 | 3.17 | 26 | 0.0030 |
| na_r_geo | 1929 | 1870 | 4.63 | 54 | 0.0024 |
| **Average** | **1457** | **1305** | **22.18** | **130** | **0.0073** |

(c)NAL-AGROVOC

**Table 5: Elapsed Times (in Seconds)**

Table 5 gives the elapsed times of our solution over all tasks. Three steps in the table correspond to the major steps of the approach shown in Algorithm 1, i.e., selecting concepts, constructing sub-ontologies and finding matching results. It can be observed that the elapsed times of constructing sub-ontologies (*step*2) are marginal compared to that of other two steps. For example, the average values of *step*2/*total* are all less than 0.01 for three sets of tasks, showing the efficiency of constructing sub-ontologies.

## 4.3 Comparative Experiments

In this part, we compare our proposed approach with other existing solutions in terms of precision, recall, $F1$ and elapsed time.

### 4.3.1 Precision

This part evaluates the precisions of four approaches. Figure 3 shows the comparison results of all related tasks. Note that the last item in each sub-figure is an overall precision value of tasks involved in the corresponding datasets. The overall precision value is defined as a weighted average value as follows:

$$P_{overall} = \sum w_i P_i, \quad w_i = \frac{m_i}{\sum m_i}$$

where $P_i$, $w_i$ and $m_i$ denote precision, weight and number of matches of a task (number of matches is shown in Table 3).

We note that in our experimental environments, $Name$ performs better in precision than $SimFlood$ but worse in recall (see Figure 3(a) and Figure 4(a))[6]. And our proposed solution, $Name$-$Gauss$-$SimFlood$ which is composed of both $Name$ and $SimFlood$ techniques, has better overall precision values in all three sets of tasks. This is contributed to the introduction of Gauss Function based influence calculation which results in an accurate yet small-size sub-ontology. Moreover, this sub-ontology makes it plausible to employ the similarity flooding method ($Name$-$Gauss$-$Flood$), leading to the better precision performance (see overall values in Figure 3(b) and (c)).

### 4.3.2 Recall

Similarly, we define an overall recall value as

$$R_{overall} = \sum w_i R_i$$

where $w_i$ is same as that in overall precision definition and $R_i$ denotes the recall value of one task.

The recall comparison results are shown in Figure 4. As expected, the proposed $Name$-$Gauss$-$Flood$ method performs best over three sets of tasks. This also proves the significance of sub-ontology construction.

### 4.3.3 F1 Measure

Figure 5 summarizes the results of precisions and recalls, and shows the evaluated $F1$ values. It is not surprising that $Name$-$Gass$-$Flood$ has the best performance since both its overall precisions and recalls outperform others.

### 4.3.4 Elapsed Time

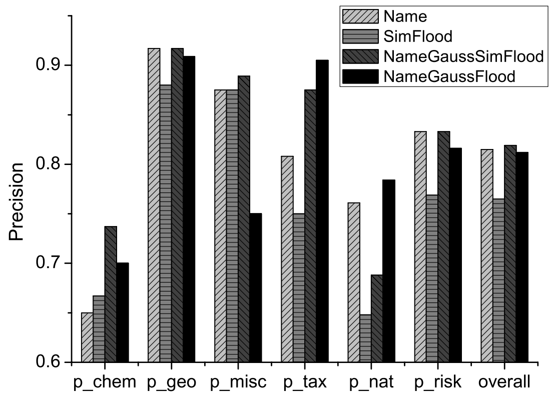
Figure 6 illustrates the elapsed times of four methods. In all three sets of tasks, the times consumed by $Name$-$Gauss$-$SimFlood$ and $Name$-$Gauss$-$Flood$ approaches are almost same to that of $Name$ solution. This is because the first step in $Name$-$Gauss$-$SimFlood$/$Name$-$Gauss$-$Flood$ employs the same method as $Name$ does. Besides that, the elapsed times of the following two steps are trivial compared to that in the first step (see Table 5). As a result, the comparable performances on elapsed times are expected.

## 4.4 Summary

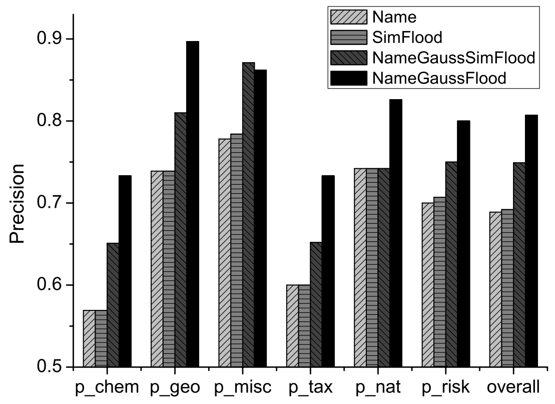
We summarize the experimental results as follows.

1. The Gauss Function based approach can filter out most noisy concepts from the heavyweight ontology by using the influences defined between concepts. As a result,

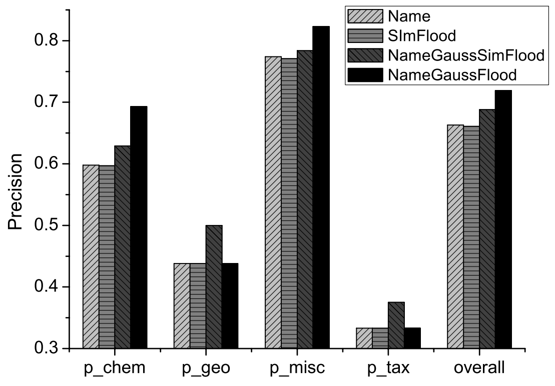
[6]Usually precision and recall contradict each other. This is similar to the relationship between false positive and false negative.



(a) GEMET-AGROVOC

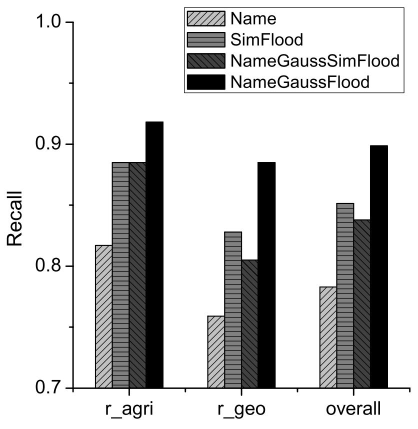


(b) GEMET-NAL

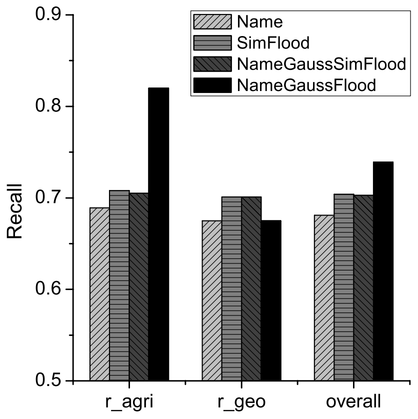


(c) NAL-AGROVOC

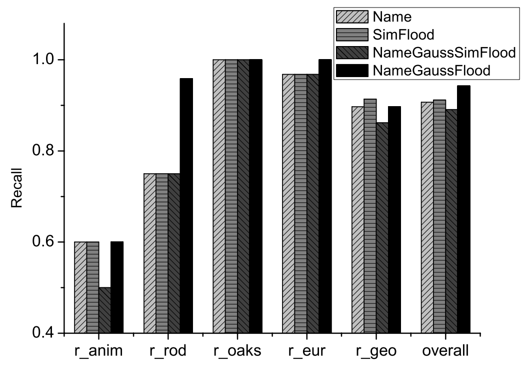
**Figure 3: Precision**



(a) GEMET-AGROVOC



(b) GEMET-NAL



(c) NAL-AGROVOC

**Figure 4: Recall**

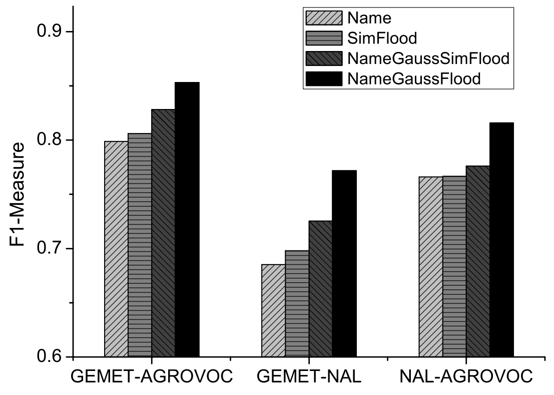


**Figure 5: F1 Measure**

our solution improves the precision scores of matching tasks.

2. In addition, our method successfully identifies the most relevant concepts in the target ontologies, leading to the constructions of sub-ontologies similar to the source ontologies. Therefore, it greatly improves the recall values.

3. Finally, the sub-ontology constructions are proved to be effective and efficient. This brings the comparative elapsed time performance to other solutions.

## 5. RELATED WORK

In this section, we review the research efforts that are related to this paper. The existing research works could be classified into the several categories below.

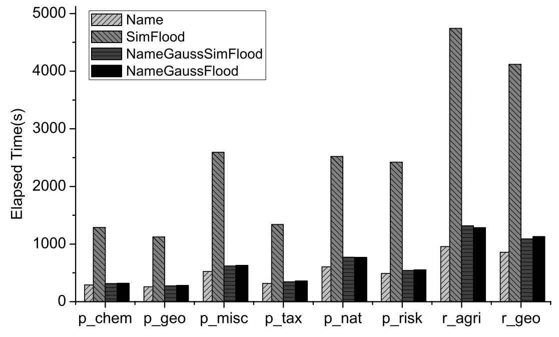
### 5.1 Name-based Approaches

The name-based approaches are the simplest solutions. They use names, labels or comments of concepts in the ontologies to suggest the semantic correspondences. Among them, one class of approaches, called string-based approaches, utilize the string structures to help identify the relationships between concepts. In [14], various string-based matching techniques, including edit-distance and token-based functions, e.g., *Jaccard similarity* and $TFIDF$, are compared. Another class of name-based methods employ Natural Language Processing ($NLP$) techniques to assess the similarities. An example is [11] which proposes a similarity calculation method by using thesaurus WordNet.

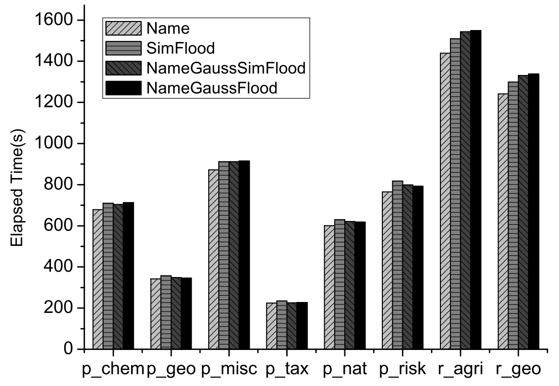
### 5.2 Structure-based Approaches

The structure-based techniques consider the structures of ontologies when generating matches. Some of them utilize the property information, for example, data types associated with concepts, to identify the similarities. For instance, [22] uses the cardinalities of properties to match concepts.

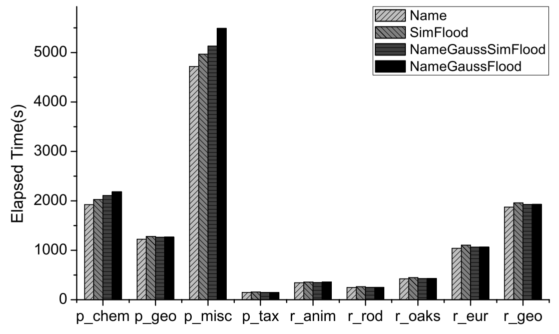
Other solution, like similarity flooding [26], creates a similarity propagation graph according to the structures of ontologies involved in matching tasks and iteratively computes similarities. Besides, [13] converts ontology matching to a



(a) GEMET-AGROVOC



(b) GEMET-NAL



(c) NAL-AGROVOC

**Figure 6: Elapsed Time**

graph theoretic problem, then a polynomial-delay algorithm is adopted to enumerate the satisfying assignments of Horn formulas.

## 5.3 Instance-based Solutions

As mentioned in Section 2, instances are also defined as components in ontologies. As a result, instance could be treated as "correct answers" of matching. In [31], the matching issues are formulated as classification problems and a machine learning technique is developed to learn the relationship between the similarity of instances and the validity of mappings between concepts. In addition, [21] constructs the semantic links between concepts based on the co-occurrence of instances. The basic idea behind is that the more significant the overlap of common instances of two concepts is, the more related the concepts are.

## 5.4 Background Knowledge Methods

Recently, some researchers suggest to use background knowledge to improve the performance of ontology matching. For example, when constructing the semantic correspondences between the otologies with different languages, a dictionary could be properly used to fill the gap between the languages. Background knowledge could be of various formats, such as large scale ontologies, online dictionaries and even the web pages distributed in Internet. [18] presents a novel approximate method to discover the matches between concepts in directory ontology hierarchies. It utilizes Google search engine to define the approximate matches between concepts, and finally shows a Google distance based weight measurement. In [9], Foundational Model of Anatomy (FMA) ontology is used as the context to match other medical ontologies into the concepts of anatomy domain.

## 5.5 Reasoning-based Techniques

As one of the components of ontologies, the axioms describe the semantics and logic in the ontologies. Some inference techniques then take advantages of the axioms. [30] proposes an interesting algorithm (ILIADS) that tightly integrates both data matching and logical reasoning to gain better performance of ontology matching. They achieve this by implementing an OWL Lite reasoner which can control the order of applying axioms.

## 6. CONCLUSION

In this paper, we propose a novel ontology matching approach to improve the performance of unbalanced ontology matching. The core of the solution is that we utilize Gauss Function to calculate the relevances of concepts in a heavyweight ontology to a lightweight ontology. Based on the relevance values computed, a sub-ontology which is maximally "similar" to the lightweight ontology is constructed. This makes it plausible to apply the fine-grained yet resource-consuming matching methods next. We carry out extensive experiments by using real world datasets. Experimental results clearly demonstrate the effectiveness and efficiency of the approach.

## 7. ACKNOWLEDGEMENTS

This work is supported by the Foundation of China under Grant No.90604025, the Major State Basic Research Development Program of China (973 Program) under Grant

No.2007CB310803 and No.2003CB317007. It is also supported by IBM Innovation Funding.

## 8. REFERENCES

[1] GEMET homepage http://www.eionet.europa.eu/gemet.

[2] Ontology Alignment Evaluation Initiative http://oaei.ontologymatching.org/.

[3] GEMET download site http://oaei.ontologymatching.org/2007/environment/gemet/gemet_2007_OWL.zip.

[4] AGROVOC homepage http://www.fao.org/aims/ag_intro.htm.

[5] AGROVOC download site http://oaei.ontologymatching.org/2007/food/agrovoc/agrovoc_2007_OWL.zip.

[6] NAL homepage http://agclass.nal.usda.gov/agt/.

[7] NAL download site http://oaei.ontologymatching.org/2007/food/nalt_2007_OWL.zip.

[8] Golden Standard download site http://oaei.ontologymatching.org/2007/results/environment/gold_standard/.

[9] Zharko Aleksovski, Michel Klein, Warner ten Kate, and Frank van Harmelen. Matching Unstructured Vocabularies Using a Background Ontology. In *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2006.

[10] Yuan An, Alex Borgida, and John Mylopoulos. Discovering the Semantics of Relational Tables Through Mappings. *Journal on Data Semantics,7:1-32*, 2006.

[11] Alexander Budanitsky and Graeme Hirst. Evaluating WordNet based Measures of Lexical Semantic Relatedness. *Computational Linguistics,32(1):13-47*, 2006.

[12] Silvana Castano, Alfio Ferrara, and Stefano Montanelli. Matching Ontologies in Open Networked Systems: Techniques and Applications. *Journal on Data Semantics,5:25-63*, 2006.

[13] Laura Chiticariu, Phokion G. Kolaitis, and Lucian Popa. Interactive Generation of Integrated Schemas. In *Proceedings of the 27th International Conference on Management of Data(SIGMOD)*, 2008.

[14] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A Comparison of String Metrics for Matching Names and Records. In *Proceedings of 9th International Conference on Knowledge Discovery and Data Mining(KDD) Workshop on Data Cleaning and Object Consolidation*, 2003.

[15] Hong-Hai Do and Erhard Rahm. COMA-A System for Flexible Combination of Schema Matching Approaches. In *Proceedings of the 28th International Conference on Very Large Data Bases(VLDB)*, 2002.

[16] AnHai Doan, Pedro Domingos, and Alon Halevy. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In *Proceedings of the 20th International Conference on Management of Data(SIGMOD)*, 2001.

[17] AnHai Doan and Alon Y. Halevy. Semantic Integration Research in the Database Community: A Brief Survey. *AI Magazine, 26(1):83-94*, 2005.

[18] Risto Gligorov, Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen. Using Google Distance to Weight Approximate Ontology Matches. In *Proceedings of The 16th International World Wide Web Conference(WWW)*, 2007.

[19] Laura M. Haas, Mauricio A. Hernàndez, Howard Ho, Lucian Popa, and Mary Roth. Clio Grows Up: From Research Prototype to Industrial Tool. In *Proceedings of the 24th International Conference on Management of Data (SIGMOD)*, 2005.

[20] Hai He, Weiyi Meng, Clement Yu, and Zonghuan Wu. WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of The Deep Web. In *Proceedings of 31st International Conference on Very Large Data Bases (VLDB)*, 2005.

[21] Antoine Isaac, Lourens van der Meij, Stefan Schlobach, and Shenghui Wang. An Empirical Study of Instance-Based Ontology Matching. In *Proceedings of The 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference(ISWC/ASWC)*, 2007.

[22] Mong Li Lee, Liang Huai Yang, Wynne Hsu, and Xia Yang. XClust: Clustering XML Schemas for Effective Integration. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM)*, 2002.

[23] Douglas B. Lenat and Ramanathan V. Guha. Building Large Knowledgebased Systems. *Addison Wesley, Reading (MA US)*, 1990.

[24] Dekang Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)*, 1998.

[25] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic Schema Matching with Cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB)*, 2001.

[26] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In *Proceedings of 18th International Conference of Data Engineering(ICDE)*, 2002.

[27] Tova Milo and Sagit Zohar. Using Schema Matching to Simplify Heterogeneous Data Translation. In *Proceedings of the 24th International Conference on Very Large Data Bases(VLDB)*, 1998.

[28] Natalya F. Noy. Semantic Integration: A Survey of Ontology-based Approaches. *ACM SIGMOD Record, 33(4):65-70*, 2004.

[29] Jie Tang, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang. Using Bayesian Decision for Ontology Mapping. *Web Semantics. 4(4): 243-262*, 2006.

[30] Octavian Udrea, Lise Getoor, and Renée J. Miller. Leveraging Data and Structure in Ontology Integration. In *Proceedings of the 26th International Conference on Management of Data(SIGMOD)*, 2007.

[31] Shenghui Wang, Gwenn Englebienne, and Stefan Schlobach. Learning Concept Mappings from Instance Similarity. In *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, 2008.