Measuring the Weight of Relations Between Entities

Mizuki Oka and Yutaka Matsuo

Center for Knowledge Structuring, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Hongo, Tokyo, Japan
mizuki@cks.u-tokyo.ac.jp
matsuo@biz-model.t.u-tokyo.ac.jp
http://web.mac.com/mizuki.oka/

Abstract. Extracting relations among entities is an active research area of Semantic Web studies related to semantic research and information inference. Although many studies have proposed extraction of large-scale relational data, how to weight each relation has not been well studied. Intuitively, a relation between two entities might be more important than relations between other entities. Therefore, they should be weighted more. Our goal is to assign weights to relations accurately to use the important part of the extracted relations selectively. To this end, we propose a method that automatically weighs each pair of entities using a web-based search engine. The method is based on our hypothesis that a relation between an entity pair is stronger if one entity associates the other entity with a more general term than a less general term, assuming that the information is accessible by more people. Fundamentally, the method assigns a score to each entity pair according to search results returned by a search engine. The hit count of the search engine is used to measure how general the term is. The results of this experiment show that the method can weigh entity pairs more appropriately than conventional methods.

Key words: Semantic Web, Relation extraction, Relation weighting, Social network

1 Introduction

Extracting relations among entities is an active research area of Semantic Web studies related to semantic research [1, 2] and information inference [3, 4]. For example, numerous studies have explored relation extraction to construct large social networks from the Web [5–10]. Given such extracted large-scale relational data, the weight of relations plays an important role in selective use of the important parts for various applications. For example, let us consider two pairs of entities (*Steve Jobs*, *Apple Computer Inc.*) and (*Jerry Yang*, *Yahoo!*). Each pair of the entities hold various kinds of relations such as CEO relation and Founder relation. If we consider CEO relation and compare one pair of entities with the other, intuitively the pair (*Steve Jobs*, *Apple Computer Inc.*) is more

known to people than (*Jerry Yang*, *Yahoo*). The knowledge that *Steve Jobs* has a prominent CEO relation with the entity *Apple Computer Inc.* enables us to use the extracted relational data selectively.

Many studies have been proposed to extract relational data and represented in a triple such as <Steve Jobs, CEO, Apple> creating a knowledge base annotated in the Resource Description Framework (RDF). However, having a nominal CEO relation between a pair of entities does not necessarily mean that the relation is also well known to people. As more and more knowledge is described in RDF, it becomes essential to distinguish between (1) knowledge that is correct *but not known* to people and (2) knowledge that is correct *and known* to people. For example, searching the shortest path between two people in a social network using the latter knowledge gives a more reasonable result that more people agree with than the one which is computed using the former knowledge.

How can we then differentiate whether a pair of entities *Steve Jobs* and *Apple Computer Inc.* are having a nominal CEO relation or a well-known CEO relation? One approach is to apply a PageRank like algorithm on the graphs derived from the RDFs [1, 2]. Another approach is to use co-occurrence-based metrics using a Web search engine. It is based on a simple assumption that the co-occurrence of a pair of entities on the Web pages represents the strength of the relation. This approach considers a Web search engine as a reflection of society. However, the metrics are mostly used as a way to see whether any relation exists between X and Y and thus does not answer how a particular relation between entities such as *Steve Jobs* and *Apple Computer Inc.* is known to society. Moreover, a co-occurrence-based metric computes a weight dependent on how well-known the *Steve Jobs* and *Apple Computer Inc.* themselves are.

In this paper, we consider the Web as a reflection of society and also take the following concept into consideration: *the attributes of an entity X are the ones that retrieve their values.* For example, if an entity *Steve Jobs* and its attribute *CEO* can retrieve its value *Apple Computer Inc.*, then we consider that the CEO relation of *Steve Jobs* and *Apple Computer Inc.* is well-known to people. In other words, if X and t can retrieve an entity Y, then X and Y holds a t relation. Here, we regard that Y is *retrieved* if it appears in the top research result of the query X and t of a Web search engine. Based on this notion, we propose a novel approach that is based on the assumptions that a Web search engine acts as a tool to retrieve people's common knowledge and that the appearance of Y on the top search result of of the query X and t of a Web search engine is an evidence of well-known relation.

On top of these assumptions, our methods weigh an arbitrary RDF triple $< X, t, Y >$ using the following two hypothesis: (1) If a term t is more general, the weight of relation between X and Y should be weighted more and (2) the relation between X and Y has a stronger relation when entities can be strongly inferred from either direction. The proposed method uses a general Web search engine (e.g., Yahoo!) but a Semantic Web search engine (e.g., Swoogle) can also be used to measure the weight. Our method can be applied in various applications of the Semantic Web in which a large number of entities and their relations are

available. This allows, for instance, to help complex inferential tasks such as searching an important path to a person through strongly connected relations in a social network.

The remainder of this paper is presented as follows. Section 2 presents a description of the related work. Section 3 presents a description of basic ideas of our approach and detailed steps of the proposed method. Section 4 presents a description of our experiments and evaluations. Section 5 discusses the limitation and application in the Semantic Web. We conclude with a discussion of future work in Section 6.

2 Related Work

Aiming at weighting relations among entities, our work is related to identifying underlying relations among entities of social networks. Matsuo et al. used a supervised machine learning method to label relations of four types in a research community [10]. Whereas [11] employed an unsupervised approach overcoming the shortcomings of supervised approach, where the collective context information obtained during the extraction of social networks is used for identifying the underlying relations. However, these studies do not consider how to weigh the extracted relations among entities.

The notion of weights and ranking relations has been studied for information retrieval from the Semantic Web [12, 13] and for understanding an ontology [2]. Large repositories of semantic data extracted from Web pages have been created and are publicly available. Many of these repositories hold a knowledge base annotated in the Resource Description Framework (RDF). Searches are performed on such repositories based on the analysis of semantic relations between the annotations. Although we share the concept of weighting relations through the notion of relation associations with these works, the respective methods differ markedly because of the assumption of the usage of annotated data such as RDF.

Our work is also related to measurement of the semantic similarities between pairs of words using a search engine. Sahami et al. proposed a web-based kernel method by which search results of a query are used to measure the similarities between words including the contexts of the words. The two words are considered similar [14] if the search results of each word contain many common words. However, even if the contexts of the words are similar, they do not necessarily mean that the words have a strong relation. In fact, experimental results presented in Section 4 show that our method yields a better result than the web-based kernel method. Bollegala et al. proposes a method that measures a relational similarity given pairs of words. For example, assume we have two pairs of words that have an ACQUIRER–ACQUIREE relation such as (Google Inc., YouTube) and (Microsoft Corp., Powerset), their method measures the relational similarity between these pairs of words [15]. The purpose of their method differs from ours in that they aim to measure the relational similarity between pairs of words, whereas our aim is to measure the weight of the relation given a pair of entities.

query	**target entity**	**result**
Apple Computer Inc., *CEO*	Steve Jobs	○
Yahoo!, *CEO*	Jerry Yang	×
Apple Computer Inc., *entrepreneur*	Steve Jobs	×
Yahoo!, *entrepreneur*	Jerry Yang	○
Steve Jobs *CEO*	Apple Computer Inc.	○
Jerry Yang *CEO*	Yahoo!	○
Steve Jobs *entrepreneur*	Apple Computer Inc.	○
Jerry Yang *entrepreneur*	Yahoo!	○

Table 1. Appearance of the target entity (Y) on the top ranked search result obtained using various queries (X and t).

Other works related to ours are studies on relation extraction. For example, given the relation, COMPANY–CEO, a relation extraction system must extract the instance (Steve Jobs, Apple Computer Inc.) from the sentence "Steve Jobs is an Apple Computer Inc. CEO", which can then be represented in a triple such as <Steve Jobs, is a CEO of, Apple Computer Inc.> for the Semantic Web. Several studies have addressed the extraction of instances of a target relation such as HeadquartersIn [16], InstanceOf [17], and BornIn [18] from the Web. Although these studies extract instances of a particular relation that is specified in advance, recent studies show the possibility of extracting a diverse set of relation triples from the Web with no relation-specific input [19]. The system can output instances of entities to any relation that a user gives as input. This enables extraction of a pair of entities from heterogenous data corpus such as Web. Our work can be regarded as a means to weigh a set of entities when pairs of entities are extracted using these systems.

3 Method

3.1 Concept

Given a pair of entities X and Y, we define a relation that holds between the entities as R and designate a term that indicates the relation R as a relational term t. We assume that a number of relational terms $t_i \in T$ exists for R. Our approach then assigns a weight to the relation by analyzing the top search result of a query composed of X and t_i using a Web search engine. If the top search result contains Y, then the method regards it as an evidence of people's common knowledge that the entities X and Y have a relation t_i. It then computes a weight according to the generality of the term t_i, which is measured by its web hit counts. The overall weigh of the relation of the triple $< X, R, Y >$ is then calculated as a total sum of the each weight in terms of each relational term $t_i \in T$.

As an exemplary scenario for our approach, we use two sets of relations having a *CEO* relation, namely, (*Apple Computer Inc.*, *Steve Jobs*) and (*Yahoo!*,*Jerry*

Yang), and two relational terms *CEO* and *entrepreneur* that describe the relation. Given such data, our present goal is to weigh each pair of entities. Note that we denote the triple as a pair of entities having a relation R for simplicity.

Table 1 presents an analysis of whether the page contains the entity Y (e.g. *Steve Jobs*) in the top ranked search result of a query composing the other entity X (e.g. *Apple Computer Inc.*) and a relational term t (e.g. *CEO*). For example, when the query "*Apple Computer Inc. CEO*" is issued, the top ranked search result contains the entity *Steve Jobs* (marked as ∘ in the table) although the entity *Jerry Yang* does not appear in the top ranked page (marked as × in the table) of the query "*Yahoo! CEO*". When the term *entrepreneur* is used, the opposite result is obtained. The term *CEO* gives a much larger value than the term *entrepreneur* if you look at the hit count of each term using the search engine. Based on the hypothesis described in Sect. 1, the method therefore assigns a higher weight to the relation (*Apple Computer Inc.*, *Steve Jobs*) than the relation (*Yahoo!*, *Jerry Yang*). The method also tests the other direction of the pair with, for example, the entity *Apple Computer Inc.* and the query comprising *Steve Jobs* and *CEO*. As presented in Table 1, all combinations yielded results containing the target entity Y on the top ranked page. As we regard the weight of relation as the sum of both directions in the entities, the total weight for the relation CEO is higher on (*Apple Computer Inc.*, *Steve Jobs*). In the following section, we explain the precise steps used in our proposed method.

3.2 Procedure

Our method for relation weighting of pairs of entities includes the following steps.

1 Collect a pair of entities that holds a relation R.
2 Collect relational terms $T = \{t_i; i \in M\}$ that describe the relation of pairs of entities.
3 Put queries to a web-based search engine (e.g. Yahoo!) and examine the search results.
4 Calculate the weight of the relation accordingly.

Our method requires a pair of entities (e.g., personal name, company name) and a set of terms that describe the relation among the pair of entities as the input; it then outputs its weight.

Given a pair of entities, the next step is to collect a set of relational terms $T = \{t_i; i \in M\}$ to be used as the queries. As described in this paper, we ask human subjects to provide seed terms that describe the relation and automatically expand on the set using, for example, an online thesaurus.

3.3 Model and Weighting Measures

Given a pair of entities X and Y and the set of terms $t_i \in T$, the next step is to measure how much a term t_i contributes to associate the other entity Y the entity X. The scoring of each term plays a crucial role in measuring the

1. Given (X, Y) and $t_i \in T$.
2. For each $t_i \in T$, calculate a score s_i for a relation from X to Y.
3. Calculate the sum of the scores for each $t_i \in T$ and obtain the vector model $V(X, Y)$ using the search engine,

$$V(X, Y) = \sum_{i=1}^{M} s_i.$$

4. Repeat (2)–(3) to compute the weight of the relation from Y to X and obtain

$$V(Y, X) = \sum_{i=1}^{M} s_i.$$

5. Calculate

$$weight(X, Y) = V(X, Y) + \alpha V(Y, X).$$

Fig. 1. Overall procedure for weighting the relation between two entities.

weight. Several criteria pertain to scoring a term. We require a certain model that represents the terms to calculate the weight between a pair of entities. We define the model, which we designate as $V(X, Y)$, as a vector of terms. The model $V(X, Y)$ of a pair of entities X and Y is defined using a set of M relational terms $t_1, \ldots, t_M$. Each term in the model $V(X, Y)$ is assigned a score $s_1, \ldots, s_M$. Selection of scoring terms can be accomplished in several measures: We use three term-scoring functions as follows.

BINARY We assign a score to a term t_i, either 0 or 1, depending on whether the term is an associative term or not, as defined by the function $s_i = \delta(X|Y, t_i)$:

$$\delta(X|Y, t_i) = \begin{cases} 1 & Y \text{ appears in the top search result of the query } X \text{ and } t_i, \\ 0 & \text{otherwise.} \end{cases}$$

HITCOUNT We score a term t_i according to the hit count of the term, denoted as $s_i = n(t_i)/N \cdot \delta(X|Y, t_i)$, where $n(t_i)$ signifies the number of hit count of the word t_i and N stands for the number of the total Web pages [1].

HITCOUNT+ We score a term t_i according to the generality of the defined term and multiply it by the value of the hit count of entity X to incorporate the generality of the entity itself: $s_i = n(X) \cdot (n(t_i)/N) \cdot \delta(X|Y, t_i)$. This yields

[1] The number of assumed total Web pages N is set to 19.2 billion according to data provided at http://ysearchblog.com/2005/08/08/our-blog-is-growing-up-and-so-has-our-index.

a higher score for the term t_i when the entity X itself is more commonly used on the Web.

Using the given model, we compute the weight between a pair of entities $weight(X, Y)$ as the sum of the two vector models:

$$weight(X, Y) = V(X, Y) + \alpha V(Y, X),$$

where α acts to balance the scores between two directions in the pair of entities and is determined empirically. The total weighting score of the vector model $V(X, Y)$ is given by the simple summation of the score for each term in the model:

$$V(X, Y) = \sum_{i=1}^{M} s_i.$$

The performance of each weighting measure in the model vector is evaluated in our experiments described in Section 4. All steps of the method are presented in Fig. 1.

4 Experiments

Relation Type	Total	Examples
COMPANY–CEO	392	CEO*, chairman*, head*, captain, leadership, executive, chief, manager
PERSON–FIELD	156	field*, specialty*, profession*, subject, occupy, role, career, discipline
PERSON–BIRTHPLACE	189	birthplace*, origin*, root*, nascency, base, foundation, home, source
HUSBAND–WIFE	93	partner*, couple*, spouse*, buddy, bride, copartner, pair, fellow

Table 2. Extract of relational terms for relations of the four types. A word with * is a seed term.

4.1 Dataset

We prepared pairs of entities that contain 20 instances (i.e. named-entity pairs) for each of the following four relation types. They were selected manually from data sources such as news articles and Wikipedia for reference.

COMPANY–CEO This relation holds between pairs of company names (X, Y), where X is the chief executive officer (CEO) of a company Y. We consider both current and past CEOs of companies.

PERSON–FIELD This relation holds between pairs (X, Y), where a person X is an expert or is known for abilities in a field Y. Instances of this relation contain scientists and their field of expertise, athletes and the sports they are associated with, and artists and the area in which they perform.

PERSON–BIRTHPLACE This relation holds between pairs (X, Y), where X is the name of a person, and Y is the location (place) where X was born. We consider city names and country names as locations.

HUSBAND–WIFE This relation holds between pairs (X,Y), where a person X is a husband of person Y. We consider both current couples as well as historical couples.

As for the selection of a set of relational terms for entities of each type, three terms were selected manually as seed terms. An online thesaurus dictionary [2] was used to expand the number of terms to be used. As a result, 392 terms, 156 terms, 189 terms, and 93 terms were collected, respectively, for relations COMPANY–CEO, PERSON–FIELD, PERSON–BIRTHPLACE, and HUSBAND–WIFE. The three seed terms as well as an extract of the collected terms from the online thesaurus dictionary are presented in Table 2.

4.2 Evaluation

The evaluation of the method requires a gold standard dataset to compare against. We asked six computer science researchers (i.e. annotators) to assign a weight to each pair of entities in datasets of the four types. The questionnaire was conducted with a question "How would you rate the fame of the relation between the entity pair?" Each annotator was asked to assign a score of 1–4 according to the following criteria.

- Null, I have never seen/heard about the relation (score 1)
- Poor, I have seen/heard about the relation a few times (score 2)
- Medium, I have seen/heard about the relation several times (score 3)
- High, I often see/hear the relation (score 4)

With the collected scores, the summed score was used as a golden standard weight for each pair of entities. The list of pairs of entities and the sums of all the annotators' scores for each entity are depicted, respectively, in Tables 3, 4, 5, and 6 for relation types COMPANY–CEO, PERSON–FIELD, PERSON–BIRTHPLACE, and HUSBAND–WIFE.

The weighted values obtained using our methods were compared with the golden standard data obtained using Pearson's correlation coefficient. Results obtained using our methods are also compared with those obtained using conventional methods: Jaccard coefficient, Overlap coefficient [20] for co-occurrence-based metrics and Web-based-kernel method [14] for context-similarity based metrics.

Table 7 presents results of the correlation obtained using our methods and other conventional methods. Looking at those results, **Jaccard** and **Overlap** show similar performance; the Web-based kernel method (denoted as **WBK** in the table) shows much worse results than the co-occurrence based method, which

[2] http://thesaurus.reference.com/

CEO	COMPANY	score
Steve Jobs	Apple Computer Inc.	21
Eric Schmidt	Google Inc.	20
Steve Ballmer	Microsoft Corp.	17
Jeff Bezos	Amazon.com Inc.	15
Mark Zuckerberg	Facebook Inc.	13
Terry Semel	Yahoo!	9
Jeff Kindler	Pfizer Inc.	9
Lawrence Ellison	Oracle Corp.	8
Bruce Chizen	Adobe Systems Inc.	8
Samuel Palmisano	IBM Corp.	8
John Thompson	Symantec Corp.	8
Kenneth Chenault	American Express Co.	7
Brian Roberts	Comcast Corp.	7
Paul Otellini	Intel Corp.	7
John Chambers	Cisco Systems Inc.	6
Kevin Rollins	Dell Inc.	6
James Tobin	Boston Scientific Corp.	6
Frederick Smith	Fedex Corp.	6
Sumner Redstone	Viacom	6
George David	United Technologies Corp.	6

Table 3. CEO–COMPANY

PERSON	FIELD	score
Aristotle	Philosopher	24
Tiger Woods	Golf	24
Pele	Soccer	24
Isaac Newton	Physics	24
Maria Sharapova	Tennis	23
Andre Agassi	Tennis	23
Albert Einstein	Physics	23
Carl Lewis	Athletics	23
Ian Thorpe	Swimmer	22
Richard Feynman	Physics	22
Venus Williams	Tennis	21
Cristian Ronaldo	Football	21
Carl Friedrich Gauss	Mathematics	20
Garry Kasparov	Chess	18
Roger Dederer	Tennis	16
Max Planck	Physics	16
Henri Poincare	Mathematics	15
Shane Warne	Cricket	6
Sachin Dendulkar	Cricket	6
Lata Mangeshkar	Singer	6

Table 4. PERSON–FIELD

confirms that the weights of relations do not necessary depend on the context similarity between entities. The results of the three measuring methods used in the proposed method vary considerably. The best result was obtained when the **HITCOUNT+** (denoted as **HIT+** in the table) term weighting measure was used. The relation type CEO–COMPANY produced the highest correlation with the value of 0.97. Other relation types also gave high values of 0.57–0.80, which show much higher correlation compared to conventional methods, which produced correlations of less than 0.20. However, when the other three weighting measures in the proposed method, namely, **BINARY** and **HIT** are used, they show worse performance than co-occurrence based methods. This result suggests that the generality of the term (measured through the term hit count of the search engine) as well as the generality of the entities themselves are important features in weighting the relation. When these features are considered properly, as in the case in the measure **HITCOUNT+**, we can reasonably infer that our method can estimate the weight of the relation better than conventional methods can.

We also evaluated the performance of the proposed method using the **HITCOUNT+** measure against the number of relational terms used. We varied the number of terms used from 10 to the total number of relational terms for each relation type. The result is shown in Fig. 2. Given a number of terms to use, a set of terms was selected randomly and used to compute the correlation with the golden standard data. This process was repeated 10 times for each number. The average correlation value is shown in the figure. Although the correlation values fluctuate when fewer than 50 relational terms are used, the correlation generally improves if more terms are used. The performance stabilizes when 50 relational terms are used.

PERSON	BIRTHPLACE	score
Pele	Brazil	22
Wolfgang Amadeus Mozart	Austria	21
Ludwig Van Beethoven	Germany	21
Leonard Da Vinci	Italy	21
Michelangelo	Italy	21
William Shakespeare	England	21
Issac Newton	England	21
Albert Einstein	Germany	18
Charles Robert Darwin	England	18
Charlie Chaplin	London	17
Marie Antoinette	Vienna	15
Carl Friedrich Gauss	Germany	14
Max Planck	Germany	14
Garry Kasparov	Russia	14
George Gershwin	New York	12
Henri Poincare	France	11
Franz Kafka	Prague	10
Luc Besson	New York	8
Andre Agassi	Las Vegas	7
Sachin Tendulkar	India	6

Table 5. PERSON–BIRTHPLACE

HUSBAND	WIFE	score
John Lennon	Yoko Ono	24
Adam	Eve	24
Barack Obama	Michele Obama	22
Julius Caesar	Cleopatra	21
Romeo Montague	Juliet Capulet	18
Luis XVI	Marie Antoinette	18
Brad Pitt	Angelina Jolie	18
David Beckham	Victoria Adams	17
Mario	Princess Peach	17
Andre Agassi	Stefi Graf	16
Bill Gates	Melinda Gates	13
Tom Cruise	Katie Holmes	13
Tom Hanks	Rita Wilson	12
Johnny Depp	Vanessa Paradis	9
Ashton Kutcher	Demi Moore	9
Steve Jobs	Laurene Powell	9
Augustus	Livia Drusilla	8
Ben Affleck	Jennifer Garner	8
Ben Stiller	Christian Taylor	6
Joel Madden	Nicole Richie	6

Table 6. HUSBAND–WIFE

Relation Type	BINARY	HIT	HIT+	Jaccard	Overlap	WBK
CEO–COMPANY	0.4327	0.2339	**0.9743**	0.5033	0.5037	0.4789
PERSON–FIELD	-0.4785	0.2658	**0.5923**	0.1580	0.1903	-0.398
PERSON–BIRTHPLACE	-0.2246	0.1202	**0.8038**	0.0850	0.08501	0.0797
HUSBAND–WIFE	0.0485	-0.3393	**0.5764**	0.2080	0.0371	-0.2151

Table 7. Pearson Correlation with the Golden Standard Dataset.

5 Discussion

5.1 Limitations of the proposed method

In the proposed method, the selection of a set of relational terms plays an important role. In our experiments, we manually selected the seed terms and used a thesaurus dictionary to expand on the set of relational terms. However, it is not guaranteed that all the possible and appropriate relational terms that represent the relation are included in the resulting set of terms. Selecting the set of relational terms is an important future issue in the proposed method. If, for example, we had access to a set of queries issued at the search engine site, then we might be able to use a set of terms that appear together with an entity as relational terms. Unfortunately, it is usually difficult to obtain such a query dataset unless we ourselves own a search engine service that many people use. One approach to obtain a set of terms representing a given relation is to extract terms that co-occur with an entity on the Web. Mori et al. [11] proposes such an approach. Their basic idea is to input a term, for example, CEO, to a search engine to extract terms that co-occur frequently with the term CEO as related terms. Another approach to extract the relational terms automatically is to adopt a method proposed by Bollegala et al. [15]. They represent the various semantic

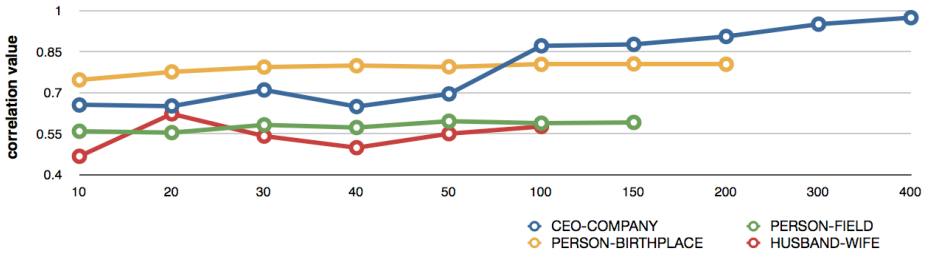

Fig. 2. Performance of the proposed method using the HITCOUNT+ measure against the number of relational terms used.

relations that exist between a pair of terms using automatically extracted lexical patterns from the Web, and cluster the extracted lexical patterns to identify the different patterns that express a particular semantic relation. Applications and evaluations of such different approaches in extracting relational terms to our method remain as important subjects for future work.

The proposed method calculates the weight of the relation between entities based on a statistical measure using the hit count on the Web. It renders it difficult for the method to measure the weight of the relation accurately when the information of the target entity is scarce on the Web. In this sense, our method relies on the assumption that information is increasingly going to be available on the Web because the means to create contents on the Web are increasing rapidly. Increasingly, people are engaging themselves in these activities.

One might also argue that the dependence of the search engine algorithm is inadequate to weigh the relation. However, we argue that it is becoming increasingly difficult to reach the information if it does not appear among the top results of the search engine because information is added to the Web every day. Moreover, search engines continue to improve themselves to incorporate people's expectations on search results as much as possible. Because of these factors, we argue that it is reasonable to use the top search engine result to measure the weight of the relation between a pair of entities as a tool to infer how strongly people associate the entity with the other entity.

5.2 Application

The proposed method is useful in several applications of the Semantic Web. As described in [3], the application of the Semantic Web is especially important in fields such as product selection and human resource management. The proposed method helps one to selectively look at important relations from a social point of view, mitigating the cumbersome work of going through a large amount of information. For example, in selecting a company to work for, a large amount of company related information can be extracted. Such relations include fields, products, services, office locations and job titles. One needs to process all these

relations to make a reasonable decision. The proposed method can help people make rational decisions by helping to reduce processing overload by allowing them to look at important relations selectively. A similar situation can be found in selecting other products such as schools and houses.

Another potentially promising area of the Semantic Web application in which the relation weighting is important is in human resource management. For example, searching for an appropriate employee for a project that requires a particular set of skills is an important task in many companies. Social networks of people serves as a useful mean for such tasks. It not only provides an overview of the relations among people as well as measuring the values of people's relation. For example, a social network for researchers extracted from the Web, called POLYPHONET, has been used at several academic conferences over six years [10]. This provides evidence of social network's usability and the potential to facilitate the discovery of researchers as well as promoting their mutual communication [21]. The proposed method of weighting relations helps to strengthen such activities.

6 Conclusion

Studies of relation extraction and term extraction have been conducted actively using the Web in the Semantic Web related studies. Taking advantage of such recent studies and extending them, we proposed a method that weights pairs of entities using a Web search engine.

Given a pair of entities and a set of terms indicating the relation between the entities, the method assigns a weight to a pair of entities (X,Y) by analyzing the top search result of a query composed of X and a term t. If the top search result contains Y, then the method regards it as evidence of people's common knowledge that the entities X and Y have a relation to the term t and that people can associate X with Y through the term t. Doing this for every term in the set of terms, it then assigns a weight to the pair of entities according to the generality of the terms.

Results of experiments demonstrate that the proposed metrics of weighting relations show positive correlations with the golden standard weighting created by human annotators. The results confirm our hypothesis that if a pair of entities has a strong relation, then a general term can associate the two entities, and that the top search result of the Web search engine offers useful metrics to evaluate the relation.

Finally, the present study shows that incorporating people's common knowledge is essential for applications of the Semantic Web. We believe that it continues to become more important, as more and more semantic data is becoming available.

References

1. Ding, L., Finin, T., Joshi, A., Peng, Y., Cost, R.S., Sachs, J., Pang, R., Reddivari, P., Doshi, V.: Swoogle: A Semantic Web Search And Metadata Engine. (2004)

2. Wu, G., Li, J., Feng, L., Wang, K.: Identifying potentially important concepts and relations in an ontology. In: 7th International Semantic Web Conference (ISWC2008). (October 2008)
3. Yamauchi, T.: The Semantic Web and Human Inference: A Lesson from Cognitive Science. In: Proc. of the 6th International Semantic Web Conference. (2007) 609–622
4. Sheth, A.P., Ramakrishnan, C.: Relationship web: Blazing semantic trails between web resources. IEEE Internet Computing **11**(4) (2007) 77–81
5. Kautz, H.A., Selman, B., Shah, M.A.: The hidden web. AI Magazine **18**(2) (1997) 27–36
6. Adamic, L., Adar, E.: Friends and neighbors on the web. Social Networks **25**(3) (2003) 211–230
7. Harada, M., ya Sato, S., Kazama, K.: Finding authoritative people from the web. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, New York, NY, USA, ACM (2004) 306–313
8. Culotta, A., Bekkerman, R., McCallum, A.: Extracting social networks and contact information from email and the web. In: Proc. of the Conference on Email and Spam. (2004)
9. Mika, P.: Flink: Semantic Web technology for the extraction and analysis of social networks. Web Semantics: Science, Services and Agents on the World Wide Web **3**(2-3) (2005) 211–223
10. Matsuo, Y., Mori, J., Hamasaki, M., Ishida, K., Nishimura, T., Takeda, H., Hasida, K., Ishizuka, M.: POLYPHONET: An Advanced Social Network Extraction System from the Web. In: WWW '06, ACM Press (2006) 397–406
11. Mori, J., Ishizuka, M., Matsuo, Y.: Extracting keyphrases to represent relations in social networks from web. In: IJCAI 07: International Joint Conference on Artificial Intelligence. (2007) 2820–2827
12. Aleman-Meza, B., Halaschek-Wiener, C., Arpinar, I.B., Ramakrishnan, C., Sheth, A.P.: Ranking complex relationships on the semantic web. IEEE Internet Computing **9**(3) (2005) 37–44
13. Anyanwu, K., Maduko, A., Sheth, A.: Semrank: ranking complex relationship search results on the semantic web. In: WWW '05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA, ACM (2005) 117–127
14. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM (2006) 377–386
15. Bollegala, D., Matsuo, Y., Ishizuka, M.: Measuring the similarity between implicit semantic relations from the web. In: 18th International World Wide Web Conference (WWW2009). (April 2009)
16. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: In Proceedings of the 5th ACM International Conference on Digital Libraries. (2000) 85–94
17. Pantel, P., Ravichandran, D.: Automatically labeling semantic classes. In: Proceedings of Human Language Technology/North American chapter of the Association for Computational Linguistics (HLT/NAACL-04), Boston, MA, USA (2004) 321–328
18. Pasca, M., Lin, D., Bigham, J., Lifchits, A., Jain, A.: Organizing and Searching the World Wide Web of Facts - Step One: The One-Million Fact Extraction Challenge. In: AAAI 2006, AAAI Press (2006)

19. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Proceedings of ACL-08: HLT, Columbus, Ohio, Association for Computational Linguistics (June 2008) 28–36
20. Manning, C.D., Schuetze, H.: Foundations of Statistical Natural Language Processing. First edition edn. The MIT Press (1999) ISBN: 0262133601.
21. Matsuo, Y., Yamamoto, H.: Community gravity: Measuring bidirectional effects by trust and rating on online social networks. In: 18th International World Wide Web Conference. (April 2009) 751–751