DEIM Forum 2015 A3-3

# Query Paraphrasing Towards Better Search by Incorporating Coordinate Relationship

Meng ZHAO†, Hiroaki OHSHIMA†, and Katsumi TANAKA†

† Department of Social Informatics, Graduate School of Informatics, Kyoto University
Yoshida Honmachi, Kyoto, 606–8501 Japan
E-mail: †{zhao,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

**Abstract** We propose a method to acquire paraphrases from the Web in accordance with a given sentence query. For example, consider the query "Guavas are rich in vitamin c". Its paraphrases are expressions or sentences that convey the same meaning but are different syntactically, such as "Guavas are well known for their high concentration of vitamin c", or "Guavas are very high in vitamin c". We aim at improving the poor performance of querying the Web by long queries, especially sentence queries, since they often result in failure. By issuing paraphrase queries to the Web, users are able to gather more search results about the given sentence query.

**Key words** paraphrase acquisition, coordinate relationship, Web mining, mutual reinforcement

## 1. Introduction

Nowadays, search engines, such as Google(注1) and Bing(注2), have become the major gateways to the huge amount of information on the Web, since they enable users to obtain useful information by issuing queries based on their information needs. It has been reported that a large fraction of search engine queries only consist of two or three keywords on average. One possible reason for such queries could be that long queries, especially sentence queries, often result in failure since they may not match any Web pages. Consequently, users are enforced to think of a few keywords that are likely to be associated with their information needs. However, it may be difficult to refine a complicated information need into two or three keywords. There have been studies to narrow the gap between short queries and information needs behind them. We concentrate on a different aspect of the problem where long queries, especially sentence queries, are treated as well represented users' information need in more detail. A conceivable reason why long queries, especially sentence queries always fail in retrieving useful information is that the expressions of such queries may be rarely used on the Web and as a result very few Web pages are returned. For example, users may want to find more information about pectin in apples and think of a sentence query like "apples pop a powerful pectin punch". None of the two aforementioned search engines return any matches for such query (at the time of writing the document). However, if the query is rewritten as "apples contain a lot of pectin", or "apples are rich in pectin", adequate number of Web pages with detailed information can be obtained. In this paper, we aim to improve the poor performance of sentence queries by expanding them with their paraphrases, especially frequently-used ones.

Paraphrases are linguistic expressions that restate the same meaning using different variations. In the most extreme case, they may not be even similar in wording. It has been shown that paraphrases are useful in many applications. For example, paraphrases can help detect fragments of text that convey the same meaning across documents and this can improve the precision of multi-document summarization [6] [16]. In the field of machine translation, [8] [14] [15] show that augmenting the training data with paraphrases generated by pivoting through other languages can alleviate the vocabulary coverage problem. In information extraction, [25] [9] [7] present approaches incorporating paraphrases to extract semantic relations among entities. In information retrieval, paraphrases have been used for query expansion [19] [2] [24]. A large proportion of previous work extract and generate paraphrases based on parallel corpora [5] [3] or comparable corpora [4] [20] [22]. However, there are limitations in using those corpora. For example, the quality of obtained paraphrases strongly depends on the quality of the corpus, a high-quality corpus can cost a great deal of manpower and time to construct. Moreover, it may be hard to cover all possible genres. For example, [22] uses a corpus consisted of newswire articles written by six different news agencies.

Entity tuples that describe or are members of the same

(注1)：http://www.google.com
(注2)：http://www.bing.com

relationships may be defined as "coordinate tuples" to each other. For example, *(guavas,vitamin c)* and *(tomatoes,potassium)* are coordinate tuples since there is a **high-Concentration** relation (注3) between *guavas* and *vitamin c*, so is between *tomatoes* and *potassium*. We think it is not easy to find all variations of paraphrases by just one entity tuple, and such variations exist in expressions of its coordinate ones. For example, given the sentence "Guavas are rich in vitamin c", it might be difficult to find part of its paraphrases, such as "Guavas are considered a high vitamin c fruit", since it is seldom used by the entity tuple *(guavas,vitamin c)*. However, such paraphrases can be acquired from the expressions of its coordinate entity tuples, i.e. the former paraphrase can be easily found via *(tomatoes,potassium)*. Thus, we can capture more paraphrases by mining the expressions of coordinate entity tuples.

The distributional hypothesis, attributed to Harris [11], has been the basis for Statistical Semantics. It states that words that occur in the same contexts tend to have similar meanings. Moreover, its extension that if two phrases, or two text units, occur in similar contexts then they may be interchangeable has been extensively tested. Our idea is based on the extended hypothesis: if two templates share more common coordinate entity tuples, then they may be paraphrase templates; if two entity tuples share more common paraphrase templates, then they may be coordinate entity tuples. Thus, paraphrase templates and coordinate tuples are in a mutually reinforcing relationship, and this relationship can be used to find more paraphrase templates or coordinate tuples.

We assume a sentence is mapped to a template and an entity tuple. Thus, given a sentence query, it can be separated into a template and a corresponding entity tuple. The proposed method first extracts templates that connect that entity tuple and entity tuples mentioned by that template. Several filters and limitations are added to eliminate partial inappropriate templates and entity tuples. A mutually reinforcing approach is proposed to simultaneously identify different templates that convey the same meaning with the given template, and entity tuples which hold the same relation with the given entity tuple. Finally, paraphrase queries can be generated by substituting the given entity tuple into discovered paraphrase templates.

Our contributions can be summarized as follows. First, we propose a method for detecting sentence-level paraphrases and our method does not require deep natural language processing such as dependency parsing. Second, paraphrases are not limited to word-level, or phrase-level. Given a sentence query, our method outputs its paraphrases at the sentence level. Third, instead of using high-quality input data restricted to a particular genre, our method can employe the Web as its data source.

The remainder of the paper is organized as follows. In Section 2., we discuss some related work. Section 3. shows some preliminaries. In Section 4., we illustrate the method to acquire paraphrases from the Web by a given sentence query. We evaluate the proposed paraphrase acquisition method using five semantic relations in Section 5.. Finally, Section 6. concludes the paper and gives an outline of our future work.

## 2. Related Work

### 2.1 Semantic Relation Extraction

Snowball [1], KnowItAll [10], TextRunner [25] are famous information extraction systems. All of them extract valuable information from plain-text documents by using lexical-syntactic patterns. Snowball and TextRunner require a handful of training examples from users, while KnowItAll emphasizes its distinctive ability to extract information without any hand-labeled training examples.

In Snowball, given a handful of example tuples, such as organization-location tuple <*o,l*>, Snowball finds segments of text in the document collection where $o$ and $l$ occur close to each other, and analyzes the text that "connects" $o$ and $l$ to generate patterns. It extracts different relationships from the Web by the bootstrap method. Besides, Snowball's patterns include named-entity tags. An example is *<ORGANIZATION>'s headquarters in <LOCATION>*. *<ORGANIZATION>* will only match a string identified by a POS tagger as an entity of type *ORGANIZATION*. So does *<LOCATION>*.

In KnowItAll, its input is a set of predicates that represent classes or relationships of interest. A generic representation of rule templates for binary predicates is *relation(Class1,Class2)*. For example, the predicate *CeoOf(PERSON,COMPANY)* corresponds to the pattern *<PERSON> is the CEO of <COMPANY>*. It learns effective patterns to extract relevant entity names.

In TextRunner, extractions take the form of a tuple $t = (e_i, r_{i,j}, e_j)$, where $e_i$ and $e_j$ are strings meant to denote entities, and $r_{i,j}$ is a string meant to denote a relationship between them. A deep linguistic parser is deployed to obtain dependency graph representations by parsing thousand of sentences. For each pair of noun phrases $(e_i, e_j)$, TextRunner traverses the dependency graph, especially the part connecting $e_i$ and $e_j$, to find a sequence of words that composes a potential relation $r_{i,j}$ in the tuple $t$.

(注3) : We define this as a food contains a high amount of a certain nutrient.

### 2.2 Paraphrase Acquisition

Paraphrase acquisition is a task of acquiring paraphrases of a given text fragment. Some approaches have been proposed for acquiring paraphrases at word, or phrasal level. However, these techniques are designed only suitable for specific types of resources. Both [21] and [23] acquire paraphrases from news article. In [21], Shinyama et al. considered that news articles reported the same event of the same day by different news agents can contain paraphrases. Thus, they proposed an automatic paraphrase acquisition approach based on the assumption that named entities are preserved across paraphrases. Pairs of similar sentences whose similarity is above a certain threshold are chosen. For any pair, if the two sentences share the same number of comparable named entities, then patterns in the two sentences are linked as paraphrases. In [23], news article headlines,which are already grouped by news aggregators such as Google News, are taken for further processing. $k$-means clustering and pairwise similarity are applied to find paraphrases, respectively. These work has explicit access to, and relies strongly on clues such as the news articles that describe the same event.

To acquire paraphrases, some works proposed methods based on deep natural language processing, i.e. dependency parsing. Lin and Pantel introduced an unsupervised method to discover inference rules from text in [13]. Inference rules include not only exact paraphrases, but also related and potentially useful expressions. Their core idea is also based on an extension to the distributional hypothesis: if two paths in dependency trees tend to occur in similar contexts, the meanings of the paths tend to be similar. The words that fill the slots of a path is regarded as a context for the path. Idan et al. [12] took a verb lexicon as the input and for each verb searches the Web for related syntactic entailment templates. Although they did not use the term "coordinate", they used a similar concept called "anchors" referred to lexical elements describing the context of a sentence. Different from our method, they first extract promising anchor sets for the verb lexicon, then extract templates (dependency parse-tree fragments) for which an entailment relation holds with the verb lexicon from sentences containing the promising anchor sets.

Paşca and Dienes proposed a method differed from previous ones in [18]. They use inherently noisy, unreliable Web documents rather than clean, formatted documents so that the paraphrases are not limited to a specific domain or a narrow class. Their proposed method is based on the assumption that if two sentence fragments have common word sequences at both extremities, then the variable word sequences in the middle are potential paraphrases of each other. So actually, their acquired paraphrases are almost word-, or phrase-level ones, while our work aims to get sentential paraphrases.

In [24], Yamamoto and Tanaka also concentrated on improving search results responded by sentence queries. Unlike we focus on paraphrases, they generally collected several types of sentence substitutions, including paraphrases, generalized sentences, detailed sentences and comparative sentences. Based on the criteria that sentence substitutions which appears frequently on the Web and whose context is similar to that of the input sentence query should be ranked higher, a ranking algorithm is also stated.

## 3. Preliminaries

We assume a sentence consists of a template and an entity tuple. Thus, given a sentence, it can be separated into a template and a corresponding entity tuple. For example, "Google has purchased Nest Labs" consists of the template ***X** has purchased **Y*** and the entity tuple *(Google,Nest Labs)*. For further illustration, we borrow the idea about the definition of a relation in [7]. They advocated a relation can be expressed extensionally by enumerating all the instances of that relation. Take the **acquisition** relation(注4) for example. An extensional definition of **acquisition** is a set of all pairs of two companies in which one company acquired another, i.e. *(Google,Nest Labs)*, *(Adobe Systems,Macromedia)*. In this paper, entity tuples hold the same relation are defined to be "coordinated" to each other. For simplicity, relations are all binary relations. Thus, in the former example, *(Adobe Systems,Macromedia)* is a coordinate entity tuple of *(Google,Nest Labs)*. Bollegala et al. [7] also introduced an intensional definition of a relation by listing all the paraphrases of that relation. Therefore, finding paraphrases of a template can also be regarded as a way to survey a certain relation. Terminologies used in this paper are listed in Table.1.

## 4. Our Method

In this paper, the problem to be solved is as follows: given a sentence, its paraphrases are automatically acquired from the Web, and they are ranked in accordance with paraphrase degree. We have stated our basic idea before, that paraphrase relationship and coordinate relationship interdepend and mutually reinforce each other. Hence, at the very beginning, it is necessary to gather templates and entity tuples. Brief introductions of template extraction and entity tuple extraction are given in Section 4.1 and Section 4.2, respectively. Then details of our method are addressed in Section 4.3.

(注4)：The **acquisition** relation exists between two companies such that one company acquired another.

Table 1 Terminologies.

| | |
|---|---|
| sentence | Google has purchased Nest Labs. |
| entity tuple | *(Google,Nest Labs)* |
| substitution | **X**=*Google*, **Y**=*Nest Labs* |
| template | **X** has purchased **Y**. |
| paraphrase templates | **X** buys **Y**, **X** has acquired **Y**, **X** finalizes acquisition of **Y** |
| paraphrases | Google buys Nest Labs. |
| | Google has acquired Nest Labs. |
| | Google finalizes acquisition of Nest Labs. |
| coordinate entity tuples | *(Microsoft,Nokia),(Yahoo,Tumblr),(Amazon,Goodreads)* |

### 4.1 Template Extraction

As we mentioned in Section 1., we use the Web as our data source, so we search the Web and extract templates from it. Suppose a given sentence is $s$ which consists of a template $t$ and an entity tuple $e$. $t$ is actually made by replacing two entities in $e$ respectively with two variables **X** and **Y** in the sentence $s$. An example is shown in Table 1. The entity tuple is *(Google,Nest Labs)*. We replace *Goolge* with variable **X** and *Nest Labs* with variable **Y** and get the template ***X** has purchased **Y***. An AND query generated from $e$ is issued to the Web, i.e "Google AND Nest Labs". We gather templates from the top $N$ search results of the query(注5) that satisfy the following conditions.

1) A template must contain exactly one occurrence of each **X** and **Y** (i.e. exactly one **X** and one **Y** must exist in a template).

2) The maximum length of a template is $L_{max}$ times of that of $s$.

3) The minimum length of a template is $L_{min}$ times of that of $s$.

4) Information such as date, money, quantity, are removed if $s$ doesn't contain such information.

5) Templates must be consistent of $s$ (if $s$ is a question, gathered templates must limit to questions; if $s$ is a declarative sentence, gathered templates must also be declarative ones).

The values of parameters $N$, $L_{max}$ and $L_{min}$ are set experimentally, as explained later in Section 5.. The proposed template extraction algorithm takes all the words in a sentence into account, and is not limited to extract templates only from the portion of a sentence that appears between two entities. Besides, we assume an overlong template is more likely to contain additional information, while a too-short template is more likely to miss some information. Both the situations lead to non-paraphrases. Therefore, we consider two length limitations to exclude some inappropriate templates in advance and reduce the number of templates gathered from the Web. The consideration of 4), 5) is because of similar reasons.

### 4.2 Entity Tuple Extraction

As we mentioned in Section 1., we use the Web as our data source, so we search the Web and extract entity tuples from it. Suppose a given sentence is $s$ which consists of a template $t$ and an entity tuple $e$. Still use the example presented in Table 1. We first search coordinate terms of two entities in $e$, respectively, using the bi-directional lexico-syntactic pattern-based algorithm [17]. For example, we get *Yahoo*, *Microsoft*, *Apple* and etc. as coordinate terms of *Google*; *Samsung*, *Dropcam* and etc. as coordinate terms of *Nest Labs*. Next, we issue wildcard queries generated by $t$ and either of the two entities in $e$ or their coordinate terms to the Web and extract the other ones from the top $M$ search results. To detect entities in sentences, we run a POS tagger(注6) and only annotate sentences exactly contained the queries with POS tags. Then nouns or noun phrases are selected out. For example, queries, such as "Google has purchased *", or "Yahoo has purchased *", are formed to extract corresponding companions. As a result, entity tuples like *(Google,YouTube)*, or *(Google,Titan Aerospace)* are extracted by the former query, entity tuples like *(Yahoo,Tumblr)*, or *(Yahoo,Blink)* are extracted by the latter query.

We use coordinate terms for the following two reasons. First, there is too massive information on the Web. If we only search by $t$ (i.e. "* has purchased *") and extract entity tuples from corresponding portions of sentences, many irrelevant tuples are gathered, such as *(God,freedom)*. Hence, coordinate terms are used to reduce the number of irrelevant tuples. Second, there might be few entity tuples extracted from the Web if the binary relation in $e$ is one-to-one type. For example, in sentence "The capital of Japan is Tokyo", relation between *Japan* and *Tokyo* belongs to one-to-one type, since we can only find *Tokyo* as the answer for which city the capital of *Japan* is, and vise versa, we can only find *Japan* as the answer for *Tokyo* is the capital of which country. Thus, it is difficult to get other entity tuples from wildcard query "The capital of * is Tokyo" or "The capital of Japan is *". In

(注5)：Replace entities in $e$ with variables.

(注6)：http://nlp.stanford.edu/software/tagger.shtml

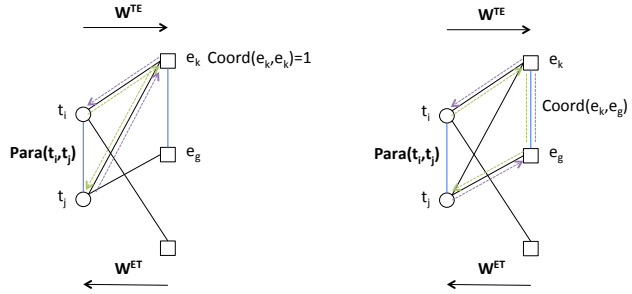


Figure 1 Illustration of the paraphrase degree calculation.

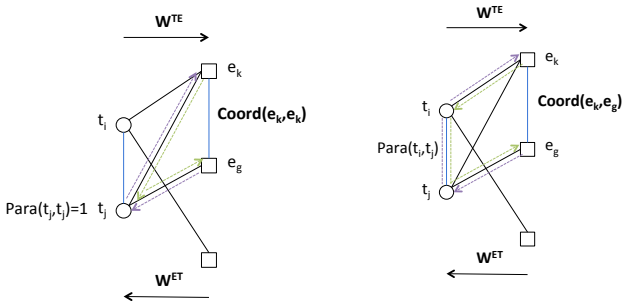


Figure 2 Illustration of the coordinate degree calculation.

this case, coordinate terms are used to increase the number of entity tuples extracted from the Web.

### 4.3 The Mutual Reinforcement Algorithm

Assuming that the set of all extracted templates is $T$, and the set of all extracted entity tuples is $E$. Suppose there are $m$ templates in $T$ and $n$ entity tuples in $E$. Let $W^{TE} \in \mathbb{R}^{m \times n}$ denote the transition matrix from $T$ to $E$, whose entry $w_{ij}^{te}$ is the proportion of $e_j$'s occurrence in $t_i$'s top search results. Let $W^{ET} \in \mathbb{R}^{n \times m}$ denote the transition matrix from $E$ to $T$, whose entry $w_{ij}^{et}$ is the proportion of $t_j$'s occurrence in $e_i$'s top search results.

Since we want to know the quality of a paraphrase rather than treat all paraphrases equally, we introduce paraphrase degree between two templates $t_i$ and $t_j$ as $Para(t_i, t_j)$, which returns a value between 0 and 1. A high value will be returned when $t_i$ and $t_j$ are more likely to be paraphrased to each other. Similarly, we introduce coordinate degree between two entity tuples $e_i$ and $e_j$ as $Coord(e_i, e_j)$, which returns a value between 0 and 1. A high value will be returned when $e_i$ and $e_j$ are more likely to be coordinated to each other.

As we mentioned before, if two templates are paraphrased to each other, they are interchangeable; if two coordinate entity tuples are coordinated to each other, they are interchangeable. In Fig.1, it shows two different situations to consider the paraphrase degree between $t_i$ and $t_j$. One is exactly equivalence of $t_i$'s suitable entity tuples and $t_j$'s suitable entity tuples, such as $e_k$ in Fig.1(a). If we can find many such entity tuples, the paraphrase degree between $t_i$ and $t_j$ is high. Another is interchangeability of $t_i$'s suitable entity tuples and $t_j$'s suitable entity tuples, i.e. $e_k$ and $e_g$ are interchangeable with the degree of $Coord(e_k, e_g)$, shown in Fig.1(b). As a result, the value of $Coord(e_k, e_g)$ is propagated to $Para(t_i, t_j)$ according to the transition probability. Similarly, additional values are propagated from other paris of coordinate entity tuples in $E$ to $Para(t_i, t_j)$, then the value of $Para(t_i, t_j)$ is updated. In Fig.2, it shows the new value is propagated to $Coord(e_k, e_g)$ in two similar situations.

Formally, the mutually reinforcing calculations are written as:

$$Para(t_i, t_j) = \frac{1}{2}\Big(\sum_{e_k, e_g \in E} w_{ik}^{te} w_{gj}^{et} Coord(e_k, e_g) + \sum_{e_k, e_g \in E} w_{jg}^{te} w_{ki}^{et} Coord(e_k, e_g)\Big)$$

$$Coord(e_k, e_g) = \frac{1}{2}\Big(\sum_{t_i, t_j \in T} w_{ki}^{et} w_{jg}^{te} Para(t_i, t_j) + \sum_{t_i, t_j \in T} w_{gj}^{et} w_{ik}^{te} Para(t_i, t_j)\Big)$$

where $i, j \in [1, m]$, $k, g \in [1, n]$. Especially, when $i = j$, $Para(t_i, t_j) = 1$, which indicates the exactly equal case. Similarly, when $k = g$, $Coord(e_k, e_g) = 1$. After values for all pairs of templates are updated, a normalization is taken place. The same for all pairs of entity tuples. Besides, up-

date continues until difference between each new value and old value is smaller than a threshold $\theta$. As a result, the paraphrase degree of two templates will be high if they share many common entity tuples, or have many interchangeable tuples; the coordinate degree of two entity tuples will be high if they share many common templates, or have many interchangeable templates. Finally, we get paraphrases of the given sentence by substituting its entity tuples into discovered paraphrase templates.

## 5. Evaluation

### 5.1 Experimental Setting

In this section, we introduce experiments to validate the main claims of the paper.

Given a sentence, it is costly to find all templates and all entity tuples through the whole Web. For our experiments, we set $N$ as 1000, viz. we limit data to the top 1000 search results obtained from Bing Search API(注7) for each AND query formed by an entity tuple. Besides, to exclude overlong or too-short templates extracted from the Web, we set $L_{max} = 2$, $L_{min} = 0.5$. We set $M$ as 250, viz. we extract entity tuples by a wildcard query in its top 250 search results. Moreover, since the calculation of $W_{TE}$ requires many accesses to the Web, we only consider 40 most frequently occurring templates. We fix the value of threshold $\theta$ to 0.0001 and find values of $Para(t_i, t_j)$ and $Coord(e_k, e_g)$ to converge after $20 \sim 25$ updates.

One claim of this paper is that paraphrase relationship and coordinate relationship mutually reinforce each other, so paraphrase templates can be selected out. To verify this, we evaluate the performance on the following five semantic relations:

(1) **highConcentration:** *We define this as a food contains a high amount of a certain nutrient.*

(2) **acquisition:** *We define this as the activity between two companies such that one company acquired another.*

(3) **founderOf:** *We define this as the relation between a person and his founded company.*

(4) **headquarter:** *We define this as the relation between a company and the location of its headquarter.*

(5) **field:** *We define this as the relation between a person and his field of expertise.*

In Table 2, we list five input sentences of the above semantic relations, and the entity tuple extracted from each sentence, respectively. Thus, templates are easily obtained by substituting entity tuples with variables. For example, in the first sentence, let **X**=*lemons*, **Y**=*vitamin c*, we have template ***X*** *are rich in* ***Y***.

(注7)：http://datamarket.azure.com/dataset/bing/search

We find paraphrase templates and coordinate entity tuples for each of these inputs by the co-acquisition method described in Section 4.. However, as our objective in this paper is to find paraphrases of a sentence query, we only evaluate generated paraphrases that for each generated one, whether it conveys the same meaning with the input sentence, and how those paraphrases help find more related information.

### 5.2 Performances of paraphrase acquisition

In this section, we show the results of the experiments and analyze them. Table 3 shows the performance of our proposed method for each of the five semantic relations and their average. We calculate the precision as how many "true" paraphrases are in the paraphrases obtained by our method. From Table 3, we can see the sentence query for the **acquisition** relation achieved the best performance with the precision of 80.8%, while the sentence query for the **headquarter** relation preforms the worst with the precision of 40%. As there isn't much work in acquiring sentential-level paraphrases from the Web, it is hard to construct a baseline to compare against. However, we can analyze them in consideration of numbers reported previously for acquiring paraphrases from the Web. TE/ASE method [12] reports obtained precision of 44.15%, compared to our average precision of 60.5%. It is difficult to estimate the recall since we do not have a complete set of paraphrases for a given sentence. Instead of evaluating recall, we calculate the average number of correct paraphrases per input sentence. The average number of paraphrases per input is 5.5 of TE/ASE method, compared to our 8.6.

In order to find the reasons why our method succeeds or fails to acquire paraphrases, let us do in-depth analysis especially on the best performance query and the worst performance query, respectively. Table 4 shows some correct and incorrect paraphrases obtained by our method for the query from the **acquisition** relation. As we mentioned before, this query achieves the best performance. Actually, we extract more than 280 templates from the top 1000 search results of the AND query "Google AND Nest Labs". The most frequently occurring templates themselves are good candidates. Therefore, we get more paraphrases with a single input. On the other hand, take the incorrect paraphrase "Google has announced plans to buy thermostat maker Nest Labs." for example. Compared with the given sentence "Google has purchased Nest Labs.", it also contains a further explanation of *Nest Labs* that *Nest Labs* is a thermostat maker, and we think such additional information leads to non-paraphrases. Although its template ***X*** *has announced plans to buy thermostat maker* ***Y*** is suitable for few extracted entity tuples, it received the propagated value from the strong coordinate degree between other tuples and *(Google,Nest Labs)*. We sur-

Table 2 Input sentences.

| relation | sentence | entity tuple |
|---|---|---|
| **highConcentration** | Lemons are rich in vitamin c. | *(lemons,vitamin c)* |
| **acquisition** | Google has purchased Nest Labs. | *(Google,Nest Labs)* |
| **founderOf** | Larry Page founded Google. | *(Larry Page,Google)* |
| **headquarter** | Yahoo is headquartered in Sunnyvale. | *(Yahoo,Sunnyvale)* |
| **field** | Albert Einstein revolutionized physics. | *(Albert Einstein,physics)* |

Table 3 Performance of our method for paraphrase acquisition.

| relation | **highConcentration** | **acquisition** | **founderOf** | **headquarter** | **field** |
|---|---|---|---|---|---|
| # Obtained | 16 | 26 | 11 | 10 | 5 |
| # Paraphrases | 9 | 21 | 5 | 4 | 4 |
| Precision | 56.3% | 80.8% | 45.5% | 40% | 80% |
| Average Precision | 60.5% | | | | |
| Average # per input | 8.6 | | | | |

Table 4 An example of some discovered paraphrases.

| sentence | Google has purchased Nest Labs. |
|---|---|
| correct | Google has acquired Nest Labs. |
| | Google is buying Nest Labs. |
| | Google owned Nest Labs. |
| | Google is buys Nest Labs. |
| | Google has announced their acquisition of Nest Labs. |
| | Google finalizes acquisition of Nest Labs. |
| incorrect | Google has announced plans to buy **thermostat maker** Nest Labs. |
| | Google has acquired **smart-gadget company** Nest Labs. |

Table 5 Another example of some discovered paraphrases.

| sentence | Yahoo is headquartered in Sunnyvale. |
|---|---|
| correct | Yahoo is located in Sunnyvale. |
| | Sunnyvale is home to notable companies such as Yahoo. |
| | Yahoo headquarters in the Sunnyvale area. |
| | Yahoo headquarters in Sunnyvale. |
| incorrect | View all Yahoo jobs in Sunnyvale. |
| | Reviews on Yahoo in Sunnyvale. |

veyed the result of coordinate entity tuples and found that entity tuples such as *(Microsoft,Nokia)*, *(Yahoo,Tumblr)* get higher coordinate values than those of other queries. This leads a misjudgment of paraphrases. Table 5 shows some correct and incorrect paraphrases obtained by our method for the query from the **headquarter** relation. As we mentioned before, this query performs the worst. Actually, we extract even less than 40 templates from the top 1000 search results of the query "Yahoo AND Sunnyvale". The reasons we considered are that firstly, there are not so many search results contained both *Yahoo* and *Sunnyvale* in a single sentence; secondly, even they are in the same sentence, that sentence may be too short, or too long. Besides, advertisements also have an influence. Take the incorrect paraphrase "View all Yahoo jobs in Sunnyvale." for example. Such advertisements are suitable for almost all extracted entity tuples, so they get higher paraphrase values. From the above discussion, we can point out that if the number of extracted templates could increase (i.e. using high-valued coordinate entity tuples to gather more templates), our method's performance would improve to some extent. And we should give a penalty to a too-general template to restrict the value propagation, since it is likely to be an advertisement, or an automatically generated sequence by a website to increase its click rate.

## 6. Conclusion

Given a sentence, our proposed method aims to find its paraphrases from the Web. Here we incorporate coordinate relationship and take a mutually reinforcing way to calculate paraphrase degree and coordinate degree. Experiments show our average precision is 60.5%, compared to TE/ASE method with average precision of 44.15%. Besides, the average number of correct paraphrases is 8.6 of our method, compared to TE/ASE method of 5.5.

As we stated in Section 5. 2, for some queries, we can-

not get enough templates. One way to solve this problem is to use high-valued coordinate entity tuples to gather more templates, and even execute our method in a iterative way. However, it causes too many accesses to the Web, and sometimes, we still cannot find enough templates. Another way to solve this problem is to do syntactic analysis to eliminate some additional information, i.e. "thermostat maker". Furthermore, we will give a penalty to a too-general template to restrict the value propagation.

## Acknowledgment

This work was supported in part by the following projects: Grants-in-Aid for Scientific Research (Nos. 24240013, 24680008) from MEXT of Japan.

### References

[1] Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM Conference on Digital Libraries. pp. 85–94. DL '00 (2000)

[2] Anick, P.G., Tipirneni, S.: The paraphrase search assistant: Terminological feedback for iterative information seeking. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 153–159 (1999)

[3] Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 597–604 (2005)

[4] Barzilay, R., Elhadad, N.: Sentence alignment for monolingual comparable corpora. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. pp. 25–32 (2003)

[5] Barzilay, R., McKeown, K.R.: Extracting paraphrases from a parallel corpus. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. pp. 50–57 (2001)

[6] Barzilay, R., McKeown, K.R., Elhadad, M.: Information fusion in the context of multi-document summarization. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 550–557 (1999)

[7] Bollegala, D.T., Matsuo, Y., Ishizuka, M.: Relational duality: Unsupervised extraction of semantic relations between entities on the web. In: Proceedings of the 19th International Conference on World Wide Web. pp. 151–160 (2010)

[8] Callison-Burch, C., Koehn, P., Osborne, M.: Improved statistical machine translation using paraphrases. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 17–24 (2006)

[9] Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Commun. ACM 51(12), 68–74 (Dec 2008)

[10] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. ARTIFICIAL INTELLIGENCE 165, 91–134 (2005)

[11] Harris, Z.S.: Distributional structure. Word 10, 146–162 (1954)

[12] Idan, I.S., Tanev, H., Dagan, I.: Scaling web-based acquisition of entailment relations. In: Proceedings of EMNLP. pp. 41–48 (2004)

[13] Lin, D., Pantel, P.: Dirt - discovery of inference rules from text. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 323–328 (2001)

[14] Madnani, N., Ayan, N.F., Resnik, P., Dorr, B.J.: Using paraphrases for parameter tuning in statistical machine translation. In: Proceedings of the ACL Workshop on Statistical Machine Translation (2007)

[15] Marton, Y., Callison-Burch, C., Resnik, P.: Improved statistical machine translation using monolingually-derived paraphrases. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. pp. 381–390 (2009)

[16] McKeown, K.R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J.L., Nenkova, A., Sable, C., Schiffman, B., Sigelman, S., Summarization, M.: Tracking and summarizing news on a daily basis with columbia's newsblaster. In: Proceedings of the second international conference on Human Language Technology Research. pp. 280–285 (2002)

[17] Ohshima, H., Oyama, S., Tanaka, K.: Searching coordinate terms with their context from the web. In: Proceedings of the 7th International Conference on Web Information Systems. pp. 40–47. WISE'06 (2006)

[18] Paşca, M., Dienes, P.: Aligning needles in a haystack: Paraphrase acquisition across the web. In: Proceedings of the Second International Joint Conference on Natural Language Processing. pp. 119–130. IJCNLP'05 (2005)

[19] Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1986)

[20] Shinyama, Y., Sekine, S.: Paraphrase acquisition for information extraction. In: Proceedings of the Second International Workshop on Paraphrasing. vol. 16, pp. 65–71 (2003)

[21] Shinyama, Y., Sekine, S., Sudo, K.: Automatic paraphrase acquisition from news articles. In: Proceedings of the Second International Conference on Human Language Technology Research. pp. 313–318. HLT '02 (2002)

[22] Wang, R., Callison-Burch, C.: Paraphrase fragment extraction from monolingual comparable corpora. In: Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web. pp. 52–60 (2011)

[23] Wubben, S., van den Bosch, A., Krahmer, E., Marsi, E.: Clustering and matching headlines for automatic paraphrase acquisition. In: Proceedings of the 12th European Workshop on Natural Language Generation. pp. 122–125. ENLG '09 (2009)

[24] Yamamoto, Y., Tanaka, K.: Towards web search by sentence queries: Asking the web for query substitutions. In: Proceedings of the 16th International Conference on Database Systems for Advanced Application (DASFAA 2011). pp. 83–92 (2011)

[25] Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., Soderland, S.: Textrunner: Open information extraction on the web. In: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. pp. 25–26 (2007)