Finding Paraphrase Facts Based on Coordinate Relationships

Meng Zhao(✉), Hiroaki Ohshima, and Katsumi Tanaka

Graduate School of Informatics, Kyoto University,
Yoshida Honmachi, Kyoto 606–8501, Japan
{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. For example, consider an input sentence “Lemon is a high vitamin c fruit”. Its paraphrases are expressions or sentences that convey the same meaning but are different syntactically, such as “Lemons are rich in vitamin c”, or “Lemons contain a lot of vitamin c”. We aim at finding sentence-level paraphrases from the noisy Web, instead of domain-specific corpora. By observing search results of paraphrases, users are able to estimate the likelihood of the sentence as a fact. We evaluate the proposed method on five distinct semantic relations. Experiments show our average precision is 60.5 %, compared to TE/ASE method with average precision of 44.15 %. Besides, we can acquire 3 paraphrases more than TE/ASE method per input.

Keywords: Paraphrase acquisition · Coordinate relationship · Web mining · Mutual reinforcement

1 Introduction

Nowadays, it is intuitive to utilize the Web as a huge encyclopedia and trust information on the Web. However, those information is not always correct or true. For example, it has been reported that information on the Wikipedia, which is regarded as the biggest online encyclopedia, is not so credible [9]. Therefore, it is necessary to understand risks of Web information and distinguish facts from it. We assume information, which is often mentioned by people on the Web, is more likely to be correct or true. Consequently, such information is regarded as “fact” with a high possibility. On the contrary, we assume information, which is rarely mentioned by people on the Web, is more likely to be incorrect or untrue, consequently unlikely to be “fact”. Based on the assumption, a naive way to estimate the likelihood of a sentence as a fact is to observe its hit count on the Web. However, it always fails since the expression of a user-input sentence may be rarely used on the Web. Suppose a user wants to know whether lemon is a high vitamin c fruit or not. He thinks of a sentence like “Lemon is a high vitamin c fruit” and use it as a query to search on the Web. Neither Google[1] nor

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

A. Liu et al. (Eds.): DASFAA 2015 Workshops, LNCS 9052, pp. 135–151, 2015.
DOI: 10.1007/978-3-319-22324-7_12

Bing[2] return any matches for this query (at the time of writing the document). However, if it is rewritten as "Lemons are rich in vitamin c", or "Lemons contain a lot of vitamin c", adequate number of Web pages can be obtained. Hence, the user can infer that lemon is a high vitamin c fruit. In this paper, we aim at finding sentence-level paraphrases from the noisy Web, instead of domain-specific corpora. By observing search results of paraphrases, especially frequently-used ones, users are able to estimate the likelihood of an input sentence as a fact.

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 [6,17]. In the field of machine translation, [8,15,16] show that augmenting the training data with paraphrases generated by pivoting through other languages can alleviate the vocabulary coverage problem. In information extraction, [7,10,26] present approaches incorporating paraphrases to extract semantic relations among entities. In information retrieval, paraphrases have been used for query expansion [2,20,25]. A large proportion of previous work extract and generate paraphrases based on parallel corpora [3,5] or comparable corpora [4,21,23]. 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, [23] uses a corpus consisted of newswire articles written by six different news agencies.

Entity tuples that describe or are members of the same 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 **highConcentration** relation 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 [12], 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

[2] http://www.bing.com.

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 Sect. 2, we discuss some related work. Section 3 shows some preliminaries and Sect. 4 describes our basic idea. In Sect. 5, we illustrate the method to acquire paraphrases from the Web by a given sentence. We evaluate the proposed paraphrase acquisition method using five semantic relations in Sect. 6. Finally, Sect. 7 concludes the paper and gives an outline of our future work.

2 Related Work

2.1 Semantic Relation Extraction

Snowball [1], KnowItAll [11], TextRunner [26] 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.

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 [22] and [24] acquire paraphrases from news article. In [22], 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 [24], 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 [14]. 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. [13] 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 [19]. 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 [25], 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[3] 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 Table 1.

Let T be the set of all possible templates in the world, E be the set of all possible entity tuples in the world. Three predicates are defined as follows:

fact(e,t). It returns *true* when the statement of sentence mapped by e and t is actually the case or has really occurred, where $e \in E$, $t \in T$. If $fact$ holds for a certain pair of an entity tuple and a template, we call the entity pair is "suitable" for the template and vice-versa.

[3] 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)*

$\boldsymbol{para(t_i, t_j)}$. It returns *true* when template t_i and template t_j both convey the same meaning ("paraphrases"), where $t_i, t_j \in T$.

$\boldsymbol{coord(e_k, e_g)}$. It returns *true* when entity tuple e_k and entity tuple e_g hold the coordinate relation, where $e_k, e_g \in E$.

4 Basic Idea

In this paper, in order to find the paraphrases of a sentence query, we aim to find pairs of templates t_i and t_j and coordinates e_k and e_g that make the predicate $para(t_i, t_j)$ and $coord(e_k, e_g)$ are true.

In the ideal world, two templates are paraphrases if every entity tuple that is suitable for one templates is also suitable for the other template. Formally, let $t_i, t_j \in T$, and $E_{t_i} = \{e|fact(e, t_i)\}$, $E_{t_j} = \{e|fact(e, t_j)\}$. If $E_{t_i} = E_{t_j}$, then $para(t_i, t_j) = true$.

Similarly, two entity tuples are coordinates if every template that is suitable for one tuple is also suitable for the other tuple. Formally, let $e_k, e_g \in E$, and $T_{e_k} = \{t|fact(e_k, t)\}$, $T_{e_g} = \{t|fact(e_g, t)\}$. If $T_{e_k} = T_{e_g}$, then $coord(e_k, e_g) = true$.

However, even in the ideal world, we can easily find a counterexample to the above discussion of **para**. Suppose t_i is $\boldsymbol{X}$ *and* $\boldsymbol{Y}$, and t_j is $\boldsymbol{X}$ *or* $\boldsymbol{Y}$. This is an extreme case where both t_i and t_j are very general templates suitable for almost all entity tuples. Consequently, E_{t_i} might be equal to E_{t_j} so that $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ and $\boldsymbol{X}$ *or* $\boldsymbol{Y}$ are misjudged as paraphrases. One may add the following condition to exclude such noisy entity tuples: if $E_{t_i} = E_{t_j}$ and $\forall(e_k, e_g) \in E_{t_i} \times E_{t_i}, coord(e_k, e_g)$, then $para(t_i, t_j)$ is *true*. Soon we find another problem that the newly added condition is too strict and will likely miss many paraphrases. A single template may represent several relations. For example, $\boldsymbol{X}$ *direct* $\boldsymbol{Y}$ may be

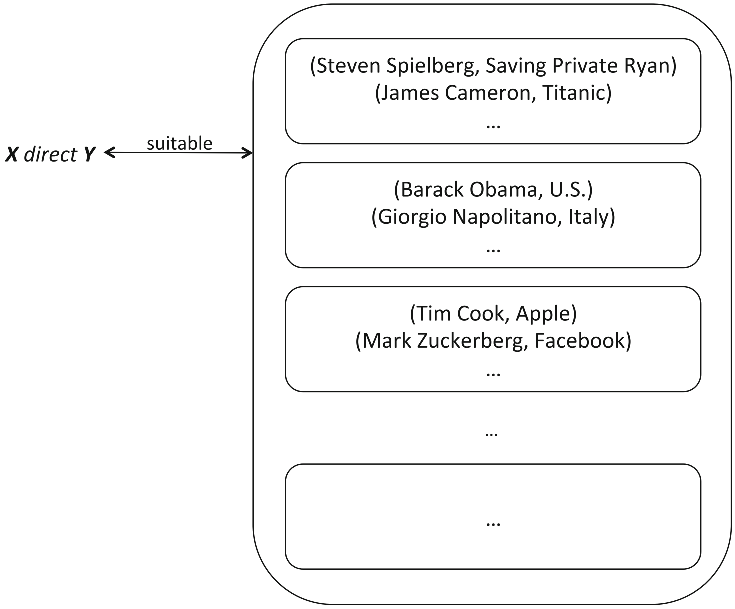

Fig. 1. An example for grouped entity tuples. Entity tuples in big frame are those suitable for the template ***X*** *direct* ***Y***, whereas entity tuples in small frame are those held the same relation.

interpreted as the **directorOf** relation[4], the **leaderOf** relation[5], or the **ceoOf** relation[6]. As a result, the entity tuples suitable for ***X*** *direct* ***Y*** are naturally grouped in accordance with the relation held by each tuple, shown in Fig. 1.

Hence, we moderate the conditions for **para** as follows:

- **If** $\exists E' \subset E_{t_i} \cap E_{t_j}$ **and** $|E'| > \alpha$, $\forall (e_k, e_g) \in E' \times E'$, $coord(e_k, e_g)$, **then** $para(t_i, t_j)$ **is** $true$.

Here α is a threshold.

Let us look at a single entity tuple. It is easy to find different relationships between the entities of the tuple. Take *(Mark Zuckerberg,Facebook)* as an example. There exists the **founderOf** relation[7] between *Mark Zuckerberg* and *Facebook*. There also exists the **ceoOf** relation between *Mark Zuckerberg* and *Facebook*. Based on our discussion that a relation can be expressed by listing all

[4] The **directorOf** relation exists between a director and his works, i.e. *(Steven Spielberg,Saving Private Ryan)*, *(James Cameron,Titanic)*.

[5] The **leaderOf** relation exists between a country and its current leader, i.e. *(Barack Obama,U.S.)*, *(Giorgio Napolitano,Italy)*.

[6] The **ceoOf** relation exists between a company and the chief executive officer of that company, i.e. *(Tim Cook,Apple)*, *(Mark Zuckerberg,Facebook)*.

[7] The **founderOf** relation exists between a person and his founded company, i.e. *(Larry Page,Google)*.

the paraphrases of that relation, we can see the similar phenomenon occurs that the templates suitable for *(Mark Zuckerberg,Facebook)* are naturally grouped in accordance with different relations. Following this, we modify conditions for **coord** as:

- **If** $\exists T' \subset T_{e_k} \cap T_{e_g}$ **and** $|T'| > \beta$, $\forall (t_i, t_j) \in T' \times T'$, $para(t_i, t_j)$, **then** $coord(e_k, e_g)$ **is** *true*.

Here β is a threshold.

However, in the real world, it is difficult to find all paraphrases by a single entity tuple perhaps because of idiomatic expressions and personal preferences. For example, consider the sentence "Guavas are rich in vitamin c", where the entity tuple is *(guavas,vitamin c)*, the template is ***X*** *are rich in* ***Y***. It might be difficult to find some of its paraphrases, such as ***X*** *are considered a high* ***Y*** *fruit*, or ***X*** *pop a powerful* ***Y*** *punch*, since people seldom use those expressions to describe the relation between guavas and vitamin c. Similarly, it is difficult to find all coordinate entity tuples by a single template, since the template might be specially used with a subset of entity tuples. Hence, we cannot find the exactly equal sets of entity tuples when considering the value of $para(t_i, t_j)$, and the exactly equal sets of templates when considering the value of $coord(e_k, e_g)$.

In Fig. 2(a), there are two templates t_1 and t_2. Under each template, there is a set of all suitable entity tuples shown in a big oval. Besides, the tuples are further grouped according to the relations they hold, shown in a small oval. If e_3 is coordinated to e_4, then we think they are interchangeable, meaning $\{e_1, e_2, e_4\} = \{e_1, e_2, e_3\}$. In addition, since e_7 is coordinated to e_5, e_6 under the same relation, we think people always use the expression of t_2 to describe e_7 but seldom use the expression of t_1. Therefore, although the sizes of two subsets are different, $\{e_5, e_6\}$ is regarded as equal to $\{e_5, e_6, e_7\}$. Finally, if all pairs of subsets are "equal", t_1 and t_2 are paraphrases, meaning $para(t_1, t_2) = true$. Similarly, in Fig. 2(b), there are two entity tuples e_1 and e_2. Under each tuple, there list all suitable templates in big oval. Besides, they are grouped according to different relations, shown in small oval. If t_3 is paraphrased to t_4, then we think they are interchangeable, meaning $\{t_1, t_2, t_4\} = \{t_1, t_2, t_3\}$. In addition, since t_7 is paraphrased to t_5, t_6 under the same relation, we think people always use the expression of t_7 to describe e_2 but seldom use it to describe e_1. Therefore, although the sizes of two subsets are different, $\{t_5, t_6\}$ is regarded as equal to $\{t_5, t_6, t_7\}$. Finally, if all pairs of subsets are "equal", e_1 and e_2 are coordinate entity tuples, meaning $coord(e_1, e_2) = true$.

5 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 in Sect. 4 that paraphrase relationship and coordinate relationship interdepend and mutually

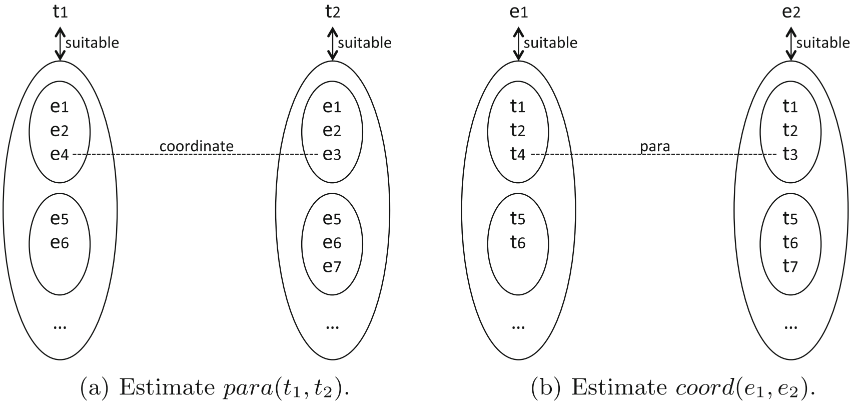

Fig. 2. A real-world situation

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 Sects. 5.1 and 5.2, respectively. Then details of our method are addressed in Sect. 5.3.

5.1 Template Extraction

As we mentioned in Sect. 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[8] 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 Sect. 6. The proposed template extraction algorithm takes all the words

[8] Replace entities in e with variables.

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.

5.2 Entity Tuple Extraction

As we mentioned in Sect. 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 [18]. 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[9] 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 this case, coordinate terms are used to increase the number of entity tuples extracted from the Web.

5.3 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.

[9] http://nlp.stanford.edu/software/tagger.shtml.

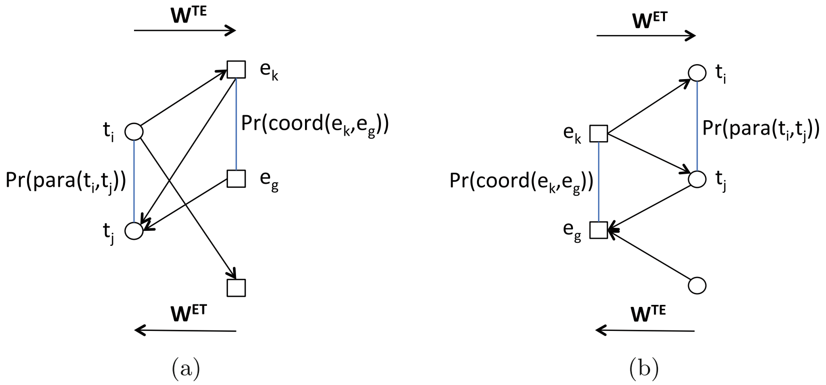

Fig. 3. An example of the mutual reinforcement between $Pr(para(t_i, t_j))$ and $Pr(coord(e_k, e_g))$.

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 $Pr(para(t_i, t_j))$, 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 $Pr(coord(e_i, e_j))$, 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 in Sect. 4, 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. 3(a), 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. 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 $Pr(coord(e_k, e_g))$. As a result, the value of $Pr(coord(e_k, e_g))$ is propagated to $Pr(para(t_i, t_j))$ according to the transition probability. Similarly, additional values are propagated from other paris of coordinate entity tuples in E to $Pr(para(t_i, t_j))$, then the value of $Pr(para(t_i, t_j))$ is updated. In Fig. 3(b), it shows the new value is propagated to $Pr(coord(e_k, e_g))$.

Formally, the mutually reinforcing calculations are written as:

$$Pr(para(t_i, t_j)) = \frac{1}{2}\Big(\sum_{e_k, e_g \in E} w^{te}_{ik} w^{et}_{gj} Pr(coord(e_k, e_g)) + \sum_{e_k, e_g \in E} w^{te}_{jg} w^{et}_{ki} Pr(coord(e_k, e_g))\Big)$$

$$Pr(coord(e_k, e_g)) = \frac{1}{2}\Big(\sum_{t_i, t_j \in T} w^{et}_{ki} w^{te}_{jg} Pr(para(t_i, t_j)) + \sum_{t_i, t_j \in T} w^{et}_{gj} w^{te}_{ik} Pr(para(t_i, t_j))\Big)$$

where $i, j \in [1, m]$, $k, g \in [1, n]$. Especially, when $i = j$, $Pr(para(t_i, t_j)) = 1$, which indicates the exactly equal case. Similarly, when $k = g$, $Pr(coord(e_k, e_g)) = 1$. After values for all pairs of templates are updated, a normalization is taken place. The same for all pairs of entity tuples. Besides, update continues until difference between each new value and old value is smaller than a threshold θ. 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.

6 Evaluation

6.1 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[10] 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 θ to 0.0001 and find values of $Pr(para(t_i, t_j))$ and $Pr(coord(e_k, e_g))$ 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:

[10] http://datamarket.azure.com/dataset/bing/search.

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.*

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)*

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***.

We find paraphrase templates and coordinate entity tuples for each of these inputs by the co-acquisition method described in Sect. 5. Our evaluation will consider only paraphrasing, i.e. given a sentence s, we will assess the quality of its paraphrases we acquire from the Web, whether they convey the same meaning with the given sentence. We do not assess whether it is really a fact.

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

6.2 Results

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 [13] 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 $\boldsymbol{X}$ *h*as announced plans to buy thermostat maker $\boldsymbol{Y}$ 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 surveyed 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.

7 Conclusion

Given a sentence, our proposed method aims to find its paraphrases from the noisy 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 Sect. 6.2, for some queries, we cannot 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, DL 2000, pp. 85–94 (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. Denning, P., Horning, J., Parnas, D., Weinstein, L.: Wikipedia risks. Commun. ACM **48**(12), 152–152 (2005)
10. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Commun. ACM **51**(12), 68–74 (2008)
11. 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. Artif. Intell. **165**, 91–134 (2005)
12. Harris, Z.S.: Distributional structure. Word **10**, 146–162 (1954)
13. Idan, I.S., Tanev, H., Dagan, I.: Scaling web-based acquisition of entailment relations. In: Proceedings of EMNLP, pp. 41–48 (2004)
14. 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)

15. 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)
16. 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)
17. 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)
18. Ohshima, H., Oyama, S., Tanaka, K.: Searching coordinate terms with their context from the web. In: Aberer, K., Peng, Z., Rundensteiner, E.A., Zhang, Y., Li, X., Unland, R. (eds.) WISE 2006. LNCS, vol. 4255, pp. 40–47. Springer, Heidelberg (2006)
19. Paşca, M., Dienes, P.: Aligning needles in a haystack: paraphrase acquisition across the web. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y., Unland, R. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 119–130. Springer, Heidelberg (2005)
20. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York (1986)
21. Shinyama, Y., Sekine, S.: Paraphrase acquisition for information extraction. In: Proceedings of the Second International Workshop on Paraphrasing, vol. 16, 65–71 (2003)
22. Shinyama, Y., Sekine, S., Sudo, K.: Automatic paraphrase acquisition from news articles. In: Proceedings of the Second International Conference on Human Language Technology Research, HLT 2002, pp. 313–318 (2002)
23. 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)
24. 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, ENLG 2009, pp. 122–125 (2009)
25. Yamamoto, Y., Tanaka, K.: Towards web search by sentence queries: asking the web for query substitutions. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part II. LNCS, vol. 6588, pp. 83–92. Springer, Heidelberg (2011)
26. 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)