Towards Web Search by Sentence Queries: Asking the Web for Query Substitutions

Yusuke Yamamoto[1,2] and Katsumi Tanaka[1]

[1] Graduate School of Informatics, Kyoto University, Japan
{yamamoto, tanaka}@dl.kuis.kyoto-u.ac.jp
[2] JSPS Research Fellow

Abstract. In this paper, we propose a method to search the Web for sentence substitutions for a given sentence query. Our method uses only lexico-syntactic patterns dynamically generated from the input sentence query to collect sentence substitutions from the Web on demand. Experimental results show that our method works well and can be used to obtain sentence substitutions for rare sentence queries as well as for popular sentence queries. It is also shown that our method can collect various types of sentence substitutions such as paraphrases, generalized sentences, detailed sentences, and comparative sentences. Our method searches for sentence substitutions whose expressions appear most frequently on the Web. Therefore, even if users issue the sentence query by which Web search engines return no or few search results for some reasons, our method enables users to collect more Web pages about the given sentence query or the sentences related to the query.

1 Introduction

Web search engines like Google and Bing are great tools for searching the Web. People can efficiently obtain what they want by conveying their information needs as queries to Web search engines. There are three possible ways to generate queries for Web search engines: keyword query, phrase query, and sentence query. Using keyword queries or phrase queries, people can obtain many Web pages containing the queries, but sometimes many irrelevant Web pages are also collected. In contrast, when using sentence queries, people can convey their information needs in more detail, expecting to obtain very relevant Web pages.

Although sentence queries are very useful for clearly representing information needs, people rarely use them because the sentence queries often cause users failure to get Web pages about the queries. There are two possible cases not to obtain Web pages well using sentence queries. The first case is that the meaning of sentence queries is correct but the expressions of the queries are rare on the Web. For example, when users want to search for Web pages describing that Germany is famous for beer and they issue sentence query "beer is famous in Germany", if the expression of the sentence query appears in few Web pages, Web search engines return few results. The second case is that what sentence queries means is wrong or rare on the Web. For example, if users misunderstand

that the capital of *Austria* is Canberra and they issue sentence query "the capital of *Austria* is Canberra", Web search engines do not return any results although they can returns Web pages describing "the capital of *Australia* is Canberra". The both cases result from the cause that if expressions of sentence queries rarely appear on the Web, search engines cannot return any relevant Web pages.

As for keyword query search, most Web search engines provide query substitution functions to deal with the cases that users' queries have miss spelling, that the queries' expression or meaning is rare or abstract, or that the queries are partially wrong because of users' misunderstandings [1, 4]. Users can obtain more Web pages which they want to browse using suggested keyword queries. However, unfortunately, there is no Web search engines provide query substitution functions for sentence queries as far as we know. For more flexible Web search with natural language, query substitution for sentence query is important.

In this paper, we propose a method to search for sentence substitutions for sentence queries, for obtaining more Web search results for the initial sentence queries and sentences related to them. Our main idea is to search the Web for sentence substitutions of a sentence query by considering popular expressions and popular topics on the Web. In our method, given a sentence query, we first collect paraphrases for the sentence query from the Web by issuing keywords consisting of the sentence query to Web search engines. After that, we extract sentence substitutions from Web search engines' indices by applying lexico-syntactic pattern mining with the sentence query and its paraphrases. Our proposed method does not require huge corpora in advance because necessary and fresh corpora are collected by using Web search engines on demand. The method also does not need language dictionaries or tools like POS taggers or parsers. Therefore, our method can search for sentence substitutions for any type of sentence query.

2 Related Work

One possible output of our method is paraphrase. Many studies have been done on paraphrasing in the field of natural languages processing. Qiu et al. presented a framework to recognize paraphrases from text corpora, focusing on the dissimilarity between sentences [6]. Kaji et al. proposed a method to paraphrase from expressions for written language to ones for spoken language based on occurrence in written and spoken language corpora [5]. As in these studies, most approaches are based on off-line processing through machine learning or deep natural-language processing, and they require huge corpora for paraphrasing in advance. Moreover, most are focused on only paraphrases based on types of phrase substitution. In contrast, our method collects not only paraphrases but also related sentences (generalized sentences, specified sentences, comparative sentences, and so on) as sentence substitutions for a given sentence query, and these are collected from Web search engine indices on demand.

In the field of information extraction, lexico-syntactic patterns are often used for entity extraction [3, 7]. KnowItAll is a system for searching the Web for entity names in the same class as a given example using lexico-syntactic patterns

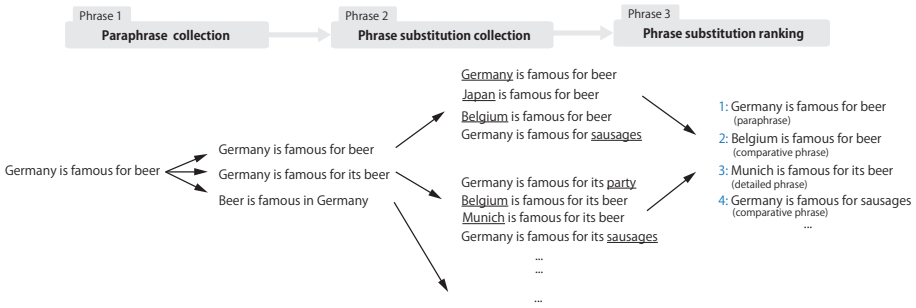

Fig. 1. Workflow of our method for the sentence query, *"Germany is famous for beer"*.

like *"such as"* and *"and other"* [2]. KnowItAll learns effective syntactic patterns for entity extraction in advance from many relevant and irrelevant terms for expected entity names. In our previous work, we used lexico-syntactic pattern mining techniques to develop HontoSearch, a system which collects comparative sentences for a given sentence that helps users check the credibility of a given sentence [8]. The goal of this study is to comparative sentences for credibility judgment on a given sentence. On the other hand, our sentence substitution method provides paraphrases, generalized sentences, and specialized sentences for a given sentence as well as comparative sentences, for the purpose of assisting users to efficiently obtain Web pages by sentence queries..

3 Method

Given sentence query q, we wish to search the Web for sentence substitution s of q. We define this as $q \mapsto s$. The goal of our work is to collect sentence substitutions $S = \{s|q \mapsto s\}$ from the Web and rank them using ranking function rank$(s|q)$. The workflow of our approach is shown below (Fig.1 illustrates the workflow when sentence *Germany is famous for beer* is given): We first collect paraphrases $P = \{p_1, p_2, .., p_m\}$ of q from the Web using core terms of q (Phase 1). The core terms are the ones which consist of the sentence query and are not stopwords. After that, we obtain Web search results by using P and collect q's sentence substitutions S from the search results (Phase 2). To collect S from the Web search results without using specific language parsers such as POS taggers, we use a combination of multiple lexico-syntactic patterns which we can generate with S. After collecting S, we rank each sentence substitution $s \in S$ for q through rank$(s|q)$, which evaluates s's frequency of appearance on the Web and its relevance for q (Phase 3).

3.1 Searching for Paraphrases

Given sentence query q, we first collect sentences that contain all core terms in sentence query q and that have a low edit distance between them and q. We

regard such sentences as paraphrases of q. For example, given sentence query q ="*Germany is famous for beer*", "*Germany is famous for its beer*" contains all of the core terms {*Germany, beer, famous*} of q, and the edit distance between it and q is low. Therefore we regard sentence "*Germany is famous for it beer*" as one of possible paraphrases of q.

We search the Web for paraphrases P of sentence query q as follows:

Phase 1. Searching the Web for paraphrases

1. The given q is divided into terms. Stop words are then omitted from a list of the terms. The remaining terms are denoted as $T_q = \{t_1, t_2, ..., t_n\}$ and we call T_q the *core terms*.
2. Text contents (snippets) that contain all terms in T_q are gathered by issuing query "$t_1 \wedge t_2 \wedge ... \wedge t_n$" to a conventional Web search engine. They are denoted as $\mathrm{Doc}(T_q)$.
3. The system extracts the strings that contain all terms in T_q and are minimal-length from each split sentence in $\mathrm{Doc}(T_q)$.
4. For each p_c of the extracted strings, the term-based edit distance $\mathrm{dist}_{edit}(q, p_c)$ is calculated. If $\mathrm{dist}_{edit}(q, p_c)$ is lower than threshold θ_{edit}, p_c is added to a set of paraphrases P.

3.2 Searching for Sentence Substitutions

Given a sentence query, we suppose that its sentence substitutions have the following features: *Lexical-syntactic patterns of the sentence substitutions are similar to that of the sentence query or those of its paraphrases.* For example, given sentence query q ="*Germany is famous for beer*", sentence "*Munich is famous for beer*" has the same lexico-syntactic pattern as that of q (*X is famous for beer*). Also, sentence "*Munich is famous for its beer*" has the same syntactic pattern as that of q's paraphrase "*Germany is famous for its beer*" (*X is famous for its beer*). This indicates that we can collect sentence substitutions using lexico-syntactic patterns of the sentence query and its paraphrases.

In the next phase, we search the Web for sentence substitutions for a given sentence based on the above hypothesis. Here, given sentence q, we denote the lexico-syntactic pattern to focus on term t in q as $\mathrm{pt}(q, t)$. For example, given sentence q = *"Germany is famous for beer"*, $\mathrm{pt}(q,$ "*Germany*"$)$ represents *"(*) is famous for beer"*. We use only lexico-syntactic patterns of sentence query q and its paraphrases P to search for sentence substitutions. The following is a workflow of the method.

Phase 2. Searching for sentence substitutions

1. Given q and core term $t \in T_q$, the system generates lexico-syntactic patterns of each of collected paraphrases P, $\mathrm{Pt}(P, t) = \{\mathrm{pt}(p, t) | p \in P\}$, by replacing term t in each paraphrase with asterisk.

2. The system issues each $\mathrm{pt}(p,t) \in \mathrm{Pt}(P,t)$ as a sentence query to a conventional Web search engine, and then the system gathers Web search results $\mathrm{Doc}(\mathrm{pt}(p,t))$ for each pattern.
3. For each $\mathrm{pt}(p,t) \in \mathrm{Pt}(P,t)$, the substrings that match the asterisk of $\mathrm{pt}(p,t)$ are extracted from snippets of $\mathrm{Doc}(\mathrm{pt}(p,t))$. We denote extracted substrings as $E = \{e_1, e_2, .., e_m\}$. Moreover, for each $e \in E$, the number of e extracted by only $\mathrm{pt}(p,t)$ is temporarily retained as $\mathrm{num}(e, \mathrm{pt}(p,t))$.
4. For $e \in E$, the replaceability of e for t is scored by considering $\mathrm{num}(e, \mathrm{pt}(p,t))$ for $p \in P$ and the characteristics of the paraphrases used to extract e.
5. If the replaceability of e for t is higher than threshold θ_{rep}, we replace t with e in the paraphrases with which e is extracted. After that, the replaced sentences are added to a list of sentence substitutions S.
6. For all $t \in T_q$, the operations from Step 1 to 5 are executed.

In Steps 1 and 2, we focus on a certain core term $t \in T_q$ and prepare a corpus for collecting sentence substitutions for $p \in P$ from the Web. For example, given sentence $q =$"*Germany is famous for beer*", we get $P = \{$*Germany is famous for beer, Germany is famous for its beer, Germany famous for beer*$\}$ in Phase 1. We now wish to obtain sentence substitutions of q focusing on the core term $t =$*"Germany"*. We issue sentence queries, *"* is famous for beer"*, *"* is famous for its beer"*, and *"* famous for beer"* to a conventional Web search engine.

In Step 3, we extract all substrings that match asterisk of $\mathrm{pt}(p,t)$ in $\mathrm{Doc}(\mathrm{pt}(p,t))$. For example, when we have a snippet *"Prague, the Czech Republic is famous for its beer, and they have the ..."* for $\mathrm{pt}(p, \text{"}Germany''\text{)} =$ *"* is famous for its beer"*, we can extract substrings, *Republic*, *Czech Republic*, *the Czech Republic*, *, the Czech Republic*, and *Prague, the Czech Republic* from the snippet. These are then added to a list of substrings E.

In Step 4, we score the replaceability of extracted substring $e \in E$ for core term t using the following functions:

$$\mathrm{rep}(e|t) = \sum_{p_i, p_j} \min(\mathrm{num}(e, \mathrm{pt}(p_i,t)), \mathrm{num}(e, \mathrm{pt}(p_j,t)) \cdot \mathrm{sim}_{edit}(q,p_i) \cdot \mathrm{sim}_{edit}(q,p_j) \tag{1}$$

where function $\mathrm{sim}_{edit}(q,p) = 1 - \mathrm{dist}_{edit}(q,p)$.

Formula 1 means that we estimate the replaceability of the extracted substrings for a core term in a give sentence query by considering the following hypotheses: (1) The more similar a paraphrase used for substring extraction is to a sentence query, the more replaceable an extracted substring can be with a core term of the sentence query. (2) The more kinds of paraphrase the substring is more frequently extracted through, the more replaceable the substring can be with the core term of the sentence query. In function min of Formula 1, we check the frequency of substrings that can be mutually obtained through pairs of paraphrases. This operation enables us to extract only the terms or the sentences which are grammatically replaceable for parts of paraphrases from Web snippets without using lexical analyzers or syntax analyzers.

3.3 Scoring of Sentence Substitutions

In the next phase, we score sentence substitutions. In our hypothesis, if a sentence substitution is important for a given sentence query, the substitution should meet the following conditions: (1) The sentence substitution appears frequently on the Web. (2) The context in which the substitution appears on the Web is similar to the context in which the sentence query appears on the Web.

Based on the above hypothesis, the score of sentence substitution s for sentence query q is calculated using the following formula $\mathrm{rank}(s|q)$:

$$\mathrm{rank}(s|q) = \mathrm{WebCount}(s) \cdot \mathrm{sim}_{context}(q, s). \quad (2)$$

$\mathrm{WebCount}(s)$ is the total number of Web pages that a search engine returns for sentence query s. $\mathrm{sim}_{context}(q, s)$ is the context similarity between sentence q and s on the Web. This similarity is defined as the cosine similarity between feature vectors of q and s. The feature vector of a sentence is generated as follows: First, text contents which contain the sentence are gathered by issuing the sentence as query to conventional Web search engines. We regard the collected text snippets as a document featuring the sentence. In this paper, features of the sentence vectors are defined as terms which appear in the *documents*, and a tf/idf algorithm is used to weight each feature.

4 Experiments and Results

We conducted experiments to evaluate how effective our method is for searching the Web for sentence substitutions. For this experiment, we prepared a set of 50 sentence queries and divided them into two classes. Class 1 contained 20 popular sentences, each of which appeared on at least six Web pages. Class 2 contained 30 rare sentences, each of which appeared on at most five Web pages. Table 1 shows examples from our test set.

We subjectively compared the performance of our method against that of a baseline method. The baseline method searches the Web for sentence substitutions simply by using lexico-syntactic patterns of a given sentence query. For example, given sentence query "*Germany is famous for beer*", the system generates patterns "** is famous for beer*", "*Germany is * for beer*", and "*Germany is*

Table 1. Examples of sentence queries in the test set. Each number in parentheses indicates how many Web pages the sentence query appears in.

Class	Sentence query
Popular sentence	Obama is the current president of the United States (200)
	The mouse was invented by Apple (35)
	Canberra is the capital of Australia (1378)
Rare sentence	Germany is very famous for beer (1)
	Europa was discovered by Galileo Galilei in 1610 (3)
	Sodium leads high blood pressure (0)

Table 2. Accuracy and Web access cost of two methods for popular sentence queries and rare ones. Numbers in front of the slash are for popular sentence queries, and numbers behind the slash are for rare queries.

Method	@1	@3	@5	@10	@20	Access
Our method	100/92.3	98.3/92.3	95.0/89.2	90.5/84.2	83.0/70.6	76.2/79.4
Baseline	100/88.9	93.3/79.7	88.0/76.6	81.0/65.6	70.5/46.4	3.5/3.7

*famous for **". The system then collects the sentences matching these patterns by using the method in Step 3 of Phase 2. Collected sentence substitutions are ranked according to the number of them extracted.

In our implementation, we used Yahoo! Search Boss APIs[3] to access Web search engine indices. In Phase 1, the number of search results for collecting paraphrases was fixed at 1000, and threshold θ_{edit} was set to 0.5. In Phase 2, The number of search results for collecting sentence substitutions was fixed at 50, and threshold θ_{rep} was 1. In Phase 3, we collected 50 Web documents to generate the feature vector of each sentence substitution.

4.1 Performance of Searching for Sentence Substitutions

We evaluated our algorithm from four viewpoints: (1) the accuracy of sentence substitutions ranking, (2) the processing time, and (3) the number of valid sentence substitutions. To evaluate the accuracy of the substitutions ranking, we checked the average percentage of valid sentence substitutions in the top N of the ranking. The value is denoted as @N. Here valid sentence substitutions means valid paraphrase or valid alternative sentences (generalized sentences, specified sentences, or comparative sentences) for a given sentence. We subjectively checked whether obtained substitutions are valid paraphrases or not. Note that for some sentence queries in the test set, the system failed to obtain more than N sentence substitutions. In such cases, when we obtained k results for any of such sentences (k < N), we supposed that (N-k) irrelevant results were additionally obtained and we calculated @N. To evaluate the processing time of each method, we counted the average number of Web accesses.

Table 2 shows the accuracy and the Web access cost of our method and the baseline method for both popular sentence queries and rare ones. For popular queries, both our method and the baseline method provided high accuracy for any @N. In contrast, for rare sentences, the baseline method provided low accuracy for @10 and @20, compared to our method. Regarding Web access cost, the baseline method was far less than our method. These results suggest that if we ignore results under the top 10 ranking, the baseline method might be better than our method because the baseline method can provide reasonably high accuracy with a low processing cost. However when we checked the number of valid sentence substitutions of the two methods for searching valid sentence substitutions, the relative situation changed.

[3] http://developer.yahoo.com/search/boss/

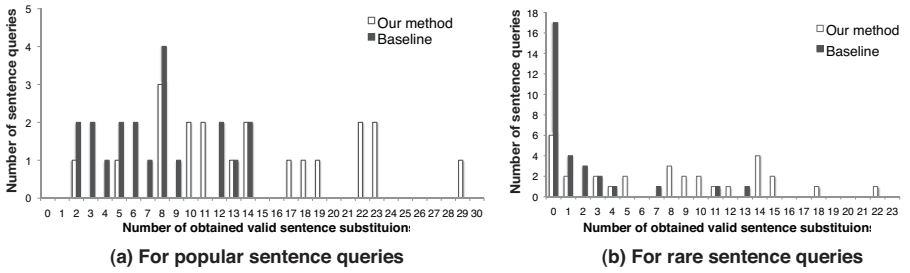

Fig. 2. Histogram of the number of obtained valid sentence substitutions. Graph (a) and (b) are histograms for popular sentence queries and rare ones, respectively.

We checked the number of valid sentence substitutions obtained by both methods. Fig.2 shows histograms of the number of the valid sentence substitutions obtained through our method and the baseline method for popular sentence queries and rare ones. For both the popular sentence queries and the rare ones, our method on average collected more kinds of valid sentence substitution than the baseline method. Notably, our method collected many sentence substitutions for most rare queries, while the baseline method collected few or no sentence substitutions for most rare queries. When our proposed method is used, similar sentences (paraphrase candidates) of a given sentence query are generated, and the system then searches the Web for sentence substitutions using multiple similar sentences. Fig.2 indicates that this operation worked well for collecting more sentence substitutions, expecially for rare sentence queries.

These results indicate that our method is more robust than the baseline method. Although Table 2 shows that the baseline method could be better than our method in some respects, people do not always search with popular sentence queries that appear on a lot of Web pages. Therefore, our method should be useful for rare sentence queries even if the processing cost is somewhat high.

4.2 Discussion

There are various types of sentence substitution. Therefore, we checked 874 obtained sentence substitutions through our method and manually categorized them into five classes: *paraphrases, generalized sentences, detailed sentences, comparative sentences, and irrelevant sentences for sentence queries.* Paraphrases are sentences semantically similar to sentence queries, but whose expression differs from one of the sentence queries. Generalized sentences are sentences whose meaning is broader or more general than the meaning of the sentence queries. Detailed sentences are those whose meaning is more specific or more detailed that of the sentence queries. Comparative sentences are useful for comparison with sentence queries from specific aspects.

Table 3 shows the percentage of obtained sentence substitutions belonging to each of the five classes. About 42% of the obtained sentence substitutions

Table 3. Percentage of sentence substitutions belonging to each of the following classes: paraphrases, generalized sentences, detailed sentences, comparative sentences, or irrelevant sentences.

Class	Ratio	Examples
Paraphrase	41.6%	*Europa was discovered by Galileo Galilei in 1610* ↦ *Galileo Galilei discovered Europa in 1610*
Comparative	27.7%	*Oil will be depleted by 2050* ↦ *Oil reserves will be depleted by 2030*
Detailed	17.2%	*Kyoto was the capital of Japan in the past* ↦ *Kyoto was the capital of Japan over 1000 years*
Generalized	7.8%	*2014 winter olymics will be held in Sochi* ↦ *2014 winter olympics are to be held in Russia*
Irrelevant	5.7%	*Ozone is potent greenhouse gas* ↦ *Ozone and is a potent greenhouse gas*

were paraphrases for the sentence queries. The second and third most obtained sentence substitutions were comparative sentences and detailed sentences. In our method, the system searches the Web for sentence substitutions with the lexico-syntactic patterns generated from sentence queries and their paraphrases. These results indicate that our approach worked well and the system succeeded in obtaining both comparative sentences and detailed sentences for the sentence queries. On the other hand, the system obtained far fewer generalized sentences than comparative sentences and detailed sentences. This suggests that our approach does not work well for collecting generalized sentences. One possible reason for this is that the lexico-syntactic patterns generated from sentence queries keep the context of the sentence queries and so they have greater potential for collecting paraphrases, comparative sentences, and detailed sentences which include the context of the sentence query rather than generalized sentences.

5 Conclusion and Future Work

In this paper, we have proposed a method to search the Web for sentence substitutions for a given sentence query. If users issue the sentence substitutions to Web search engines, users can obtain more Web pages about the query than the initial query. Our proposed method obtains sentence substitutions from the Web using lexico-syntactic patterns generated from the input sentence and its paraphrases. Our method does not need language tools such as POS taggers or parsers. Also, huge corpora do not need to be prepared in advance because the method collects fresh corpora through Web search engines on demand. Experimental results have shown that our method can accurately collect many and various sentence substitutions, especially for rare sentences on the Web.

Several problems remain regarding the search for sentence substitutions. For example, we need a method to segment core sentences of a given sentence query to generate effective lexico-syntactic patterns. Furthermore, we need to deal with cases where input sentence queries contain inappropriate or unpopular keywords

on the Web so that we can still robustly collect sentence substitutions. In the future, we plan to develop a method to automatically classify obtained sentence substitutions into classes such as paraphrases, generalized sentences, detailed sentences, and comparative sentences. We believe that such a method will enhance the ability to search for Web pages using sentence queries and knowledge mine using lexico-syntactic sentence patterns.

Acknowledgments

This work was supported in part by the following projects and institutions: Grants-in-Aid for Scientific Research (No. 18049041) from MEXT of Japan, a Kyoto University GCOE Program entitled "Informatics Education and Research for Knowledge-Circulating Society," the National Institute of Information and Communications Technology, Japan, and Grants-in-Aid for Scientific Research (No. 09J01243) from JSPS.

References

1. Craswell, N., Szummer, M.: Random Walks on the Click Graph. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2007). pp. 239–246 (2007)
2. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Web-Scale Information Extraction in KnowItAll (Preliminary Results). In: Proceedings of the 13th international conference on World Wide Web (WWW 2004). pp. 100–110 (2004)
3. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the 14th conference on Computational linguistics (ACL 1992). pp. 539–545 (1992)
4. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating Query Substitutions. In: Proceedings of the 15th international conference on World Wide Web (WWW 2006). pp. 387–396 (2006)
5. Kaji, N., Okamoto, M., Kurohashi, S.: Paraphrasing Predicates from Written Language to Spoken Language Using the Web. In: Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004). pp. 241–248 (2004)
6. Qiu, L., Kan, M.Y., Chua, T.S.: Paraphrase Recognition via Dissimilarity Significance Classification. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006). pp. 18–26 (2006)
7. Ravichandran, D., Hovy, E.: Learning Surface Text Patterns for a Question Answering System. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002). pp. 41–47 (2002)
8. Yamamoto, Y., Tanaka, K.: Finding Comparative Facts and Aspects for Judging the Credibility of Uncertain Facts. In: Proceedings of the 10th international conference on Web Information Systems Engineering (WISE 2009). pp. 291–305 (2009)