Automatic Detection of Survey Articles

Hidetsugu Nanba[1] and Manabu Okumura[2]

[1] Hiroshima City University, 3-4-1 Ozuka-higashi, Asaminami-ku
Hiroshima, 731-3194, Japan
nanba@its.hiroshima-cu.ac.jp
[2] Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku
Yokohama, 226-8503, Japan
oku@pi.titech.ac.jp

Abstract. We propose a method for detecting survey articles in a multilingual database. Generally, a survey article cites many important papers in a research domain. Using this feature, it is possible to detect survey articles. We applied HITS, which was devised to retrieve Web pages using the notions of authority and hub. We can consider that important papers and survey articles correspond to authorities and hubs, respectively. It is therefore possible to detect survey articles, by applying HITS to databases and by selecting papers with outstanding hub scores. However, HITS does not take into account the contents of each paper, so the algorithm may detect a paper citing many principal papers in mistake for survey articles. We therefore improve HITS by analysing the contents of each paper. We conducted an experiment and found that HITS was useful for the detection of survey articles, and that our method could improve HITS.

1. Introduction

Survey articles are defined as research papers, in which research in a specific subject domain is well organized and integrated. We can grasp the outline of the domain in a short time from them. However, how to detect them automatically from a huge number of research papers have not been discussed so far. We therefore study automatic detection of survey articles.

In our study, we pay attention to citation relationships between papers. Survey articles usually cite principal papers in the domain, and this feature can be used to detect them. We first detect principal papers in a domain, and then detect papers cite principal papers.

In this paper, we use the HITS algorithm [2], which ranks Web pages based on the link information among a set of documents. The HITS algorithm assumes two kinds of prominent or popular pages: authorities, which contain definitive high-quality information, and hubs, which are comprehensive lists of links to authorities. In academic literature, survey articles correspond to hubs, while papers initiating new ideas correspond to authorities, respectively. Survey articles should therefore be detected by applying the HITS algorithm to a research paper database, and selecting the papers with outstanding hub scores.

However, the HITS algorithm may also mistakenly detect papers that cite many related papers for a survey article, because the algorithm does not take account of the contents of each document. We therefore aim to detect survey articles with improved HITS algorithm, taking account of the contents of each paper.

In the remainder of the paper, Section 2 introduces the HITS algorithm and some related works. Section 3 describes our method for detecting survey articles. To investigate the effectiveness of our method, we conducted an experiment, described in Section 4. Section 5 reports the experimental results. Section 6 presents conclusions.

2. Related Work

Kleinberg proposed HITS [2], which is an algorithm to determine authoritative pages by an analysis of the link structure. The algorithm considers two kinds of pages: hubs, which are valuable as sources of good links, and authorities, which are valuable because many pages link to them. The algorithm determines authoritative pages in two stages: (1) constructing a focused sub-graph of the WWW, and (2) computing hub and authority scores of each page.

In the first stage, the t highest-ranked pages for the query σ are collected from a text-based search engine. These t pages are called a root set R. Here, t is typically set to about 200. Then, R is expanded into a base set S by adding all pages pointing to $r \in R$, and pointed to by r, to find authoritative pages that do not contain the query σ.

In the second stage, the following equations are applied to the sub-graph that was made in the first step, and then hub and authority scores of each page are then calculated.

$$x_p = \sum_{q \;\; such \;\; that \;\; q \to p} y_q \tag{1}$$

$$y_p = \sum_{q \;\; such \;\; that \;\; p \to q} x_q \tag{2}$$

where "$q \to p$" means q links to p. The authority score of page x (x_p) is proportional to the hub scores of the pages linking to page p, and its hub score y_p is proportional to the authority scores of the pages to which it links. Again, power iteration is used to solve this system of equations.

Cohn and Chang [1] proposed the probabilistic HITS algorithm (PHITS), and applied it to a full-text citation index on the WWW called Cora[1], constructed by McCallum et al. [4]. The HITS algorithm was also applied to CiteSeer[2], which is another full-text citation index on the WWW, constructed by Lawrence et al. [3]. In both systems, full-text papers were classified into several categories automatically, and HITS or PHITS was adapted to the papers in each category. The papers in each category were sorted by their hub or authority scores. Though Cohn and Chang

[1] http://www.cs.umass.edu/~mccallum/code-data.html
[2] http://citeseer.ist.psu.edu/

reported that PHITS is useful in identifying authoritative papers [1], the effectiveness of using hubs to detect survey articles has not yet been examined. We therefore investigate this, and that our method can improve the HITS algorithm.

3. Detection of Survey Articles

We improve the HITS algorithm by taking account of the features of survey articles, and apply the improved algorithm to a research paper database to detect them.

Section 3.1 describes five features used for the improvement of HITS algorithm. Section 3.2 formulates our method incorporating the five features.

3.1 Features Used in Survey Detection

We show five features as follows.

Title of a Paper (WORD)
A good clue for detecting survey articles is the presence of particular phrases in their titles. Examples of such phrases are "survey," "sabei ("survey")," "review," "rebyu ("review")," "Trend," "torendo ("trend")," "state-of-the-art," and "doukou (state-of-the-art)." We therefore double (w_{hub_1}) the hub scores of research papers if cue phrases are contained in their bibliographic information, and multiply (w_{auth_1}) authority scores by 0.5 in the opposite case.

Citation Types (CITATION TYPE)
Generally, there are few citations to base on other researchers' theories or methods, because new methods or theories based on previous works are not usually proposed in survey articles. We therefore calculate *r*, the fraction of citations that are to other researchers' theories or methods in a research paper, and multiply the hub scores by sig(*r*) (w_{hub_2}), and multiply the authority scores of each paper by sig(*1/r*) (w_{auth_2}), where sig(*x*) is defined as 2/(1+exp(1-*x*)), which changes the range of the value x from 0.5 to 2. If *r* is zero, we set w_{auth_2} to two.

We use Nanba and Okumura's method for determining the reasons for citations [5]. The method identifies the following citation types (reasons for citation) by analysing contexts of citations in research papers using several cue phrases, and obtains an accuracy of 83%.

- Type B: Citations to other researchers' theories or methods.
- Type C: Citations to compare with related work or to point out problems.
- Type O: Citations other than types B and C.

In our study, we use a database, which contains research papers written in both Japanese and English. As, Nanba and Okumura's identification rules were developed

for analysing English research papers, we developed rules for Japanese research papers in a similar manner to Nanba and Okumura's rules.

Positional Deviation of Citations (DEVIATION)

Survey articles tend to cite related papers all through the articles, while other articles tend to cite them in particular sections, such as introduction and related work. We therefore take account of the positional deviation of citations in research papers. First, we count the number of sentences between citations (d_i). Second, we calculate the deviation of distances between citations using the following equation:

$$D = \sqrt{\frac{\sum_{i=1}^{n} (\bar{d} - d_i)^2}{n}} \times \frac{1}{text\_len} \quad (3)$$

where $\bar{d}$ is an average of all distances between citations. The positional deviation of citations (D) can be obtained by normalizing the standard deviation of distances between citations with the number of sentences (*text_len*) in the research paper. Score D increases as the deviation increases, while D approaches zero when a paper cites related papers at even intervals. We therefore multiply hub scores by sig(D) (w_{hub_3}), while multiply authority scores by sig($1/D$) (w_{auth_3}).

Size of a Research Paper (SIZE)

Generally, survey articles are longer than others. We compare the length L_i (the number of sentences) of each paper with the average length $\bar{L}$, then multiply authority scores by sig($L / \bar{L}$) (w_{auth_4}), while multiply hub scores by sig($\bar{L} / L$) (w_{hub_4}).

Cue Words (CUE)

Particular phrases, such as "this survey" and "we overview" (we call them positive cue phrases) often appear in survey articles, while phrases, such as "we propose" and "this thesis" (we call them negative cue phrases) do not. We therefore use the following positive and negative cue phrases for detecting survey articles. We double (w_{hub_5}) hub scores of research papers if they contain positive cue phrases, and multiply authority scores by 0.5 (w_{auth_5}) in the opposite case.

- Positive cues: "this survey," "this review," "this overview," "(honronbun | honkou)dewa…(gaikan | gaisetsu)suru ("In this survey, we overview")"
- Negative cues: "this thesis," "this dissertation," "we propose," "teiansuru ("we propose")"

3.2 Improvement of HITS Algorithm

Using the five features explained in Section 3.1, we improve the HITS algorithm. These features are taken into account by multiplying both the hub and authority scores of the HITS algorithm by the respective weights. The authority and hub scores of each paper are calculated by the following equations.

$$x_p = \prod_{j=1}^{5} f(w_{auth_j}, L) \times \sum_{q \quad such \quad that \quad q \to p} y_q \tag{4}$$

$$y_p = \prod_{j=1}^{5} f(w_{hub_j}, L) \times \sum_{q \quad such \quad that \quad p \to q} x_q \tag{5}$$

$$f(w, L) = \begin{cases} w \times L & (if\ w > 1) \\ w / L & (if\ w < 1) \\ 1 & (if\ w = 1) \end{cases} \tag{6}$$

where w_{auth_j} and w_{hub_j} indicate the five weights for authorities and hubs, respectively. Both authority and hub scores are normalized in each iteration by $\sqrt{\sum x_p^2}$ and $\sqrt{\sum y_p^2}$, respectively, in the same way as the HITS algorithm. $f(w,L)$ is a function to change the relative importance of each weight among all weights. Changing the values L of each feature and combination of five features, we identify the best combination and optimal weights.

4. Experiments

To investigate the effectiveness of our method, we conducted an experiment. In this section, we first describe the multilingual database used in our examination. Second, we explain the experimental method, and we then report the results.

4.1 Construction of a Bilingual Database

Recently, we have been able to obtain many full-text research papers on the WWW. In this study, we construct a multi-lingual database by collecting Postscript and PDF files on the WWW. We will briefly explain the method as follows;

(1) Collecting Research Papers on the WWW:
We collected Web pages using the Web search engines Google[3] and goo[4] with the combination of five key words ((“gyoseki (“work”)” or “kenkyu (“study”)” or “publications”) and (“postscript” or “pdf”)). Then we collected all Postscript and PDF files within depth two from each collected page.

(2) Conversion of Postscript and PDF Files into Plain Texts:
We convert Postscript and PDF files into plain texts using prescript[5] and pdftotext[6], respectively. A patch for prescript for Japanese was provided by Dr. Noriyuki Katayama of the National Institute of Informatics.

(3) Analysing the Structure of Research Papers:
We remove lists of references at the ends of files using cue words, such as “sanko bunken (“references”)”, “References,” and “Bibliography.” Next, we detect the positions of citations using patterns of citation (e.g., 1), (1), [1]). We also extract bibliographic information (a title and authors) within the first five sentences in each paper.

(4) Identification of Citation Relationships between Papers:
We identify the duplication of bibliographic information extracted in step (3) for analysing whole citation relationships among papers in a database. For each pair of bibliographic records, we compare n-grams in one bibliographic record with those in the other, and count the number of matches. These matches are position-independent. If the number of matches is above a threshold, we consider the pair to be duplicates. We use a value of six for n in English texts, and three in Japanese texts.

(5) Extraction of Citation Information:
Citation types are identified based on several rules using cue phrases [5].

Finally, a bilingual research paper database was constructed. The database includes about 20,000 full-text papers (2,100 Japanese papers and 17,900 English papers) and 296,000 bibliographic references in the domain of computer science, nuclear biophysics, chemistry, astronomy, material science, electrical engineering, and so on.

[3] http://www.google.com
[4] http://www.goo.ne.jp
[5] http://www.nzdl.org/html/prescript.html
[6] http://www.foolabs.com/xpdf/

4.2 Experimental Methods

Alternatives
We conducted experiments using the following nine methods.

- Our methods
  - WORD, CITATION TYPE, SIZE, DEVIATION and CUE: combination of HITS and the named features.
  - COMB: combination of HITS and five features.
- Baseline
  - HITS: original HITS algorithm
  - BASE-WORD: research papers containing particular words, which were used for WORD, in their titles.
  - BASE-CUE: research papers containing particular cue phrases, which were used for CUE.

As we described in Section 3.2, we change L for each feature manually from zero (the feature is not used) to 10^9 in consideration of the very large range of the hub scores, and identify the optimal values and combinations.

Test Collection
In the same way as the original HITS algorithm, we prepare several base sets using some key phrases, and apply our methods to each set. The procedure to select key phrases was as follows:

1. Apply n-gram analysis to a list of bibliographic records;
2. Select 39 key phrase candidates manually, by checking the list of frequently used expressions from step 1;
3. Collect all bibliographic information including key phrases, and make a root set R for each key phrase candidate;
4. Collect all bibliographic information u that has citation relationships with any $r \in R$, and form a base set S by integrating them with R;
5. Eliminate key phrases for which S contains very few full-text papers;
6. Select the remaining candidates as key phrases.

We also took account of the variation of research domains. We finally obtained 20 key phrases, all of them are in English.

We then identified survey articles in a base set S of each key phrase. It is necessary to look through all the papers in S to obtain all the survey articles, but this is impossible if the set is very large. We therefore used a pooling method [6], known as a method for the construction of large-scale test collections. In this method, a pool of possible relevant documents is created by taking a sample of documents selected by various IR systems. Then human assessors judge the relevance of each document in the pool. We examined the top-ranked 100 documents (full-text papers) from each of eight methods.

We show the 20 key phrases, the size of each base set *S*, the number of full-text papers in each *S*, and the number of survey articles in Table 1.

Table 1. Key phrases, size of *S*, and the number of survey articles

Topics (Key phrases)	Number of bibliographic items (*S*)	Number of full-text papers	Number of survey articles
applied mathematics	2988	893	16
astronomy	1068	127	6
computer architecture	3280	1097	13
computer graphics	3324	656	14
constraint programming	689	217	5
database systems	3902	1077	33
data mining	2183	403	16
discrete mathematics	1353	299	12
distributed systems	4260	1279	24
high energy	1173	263	11
knowledge engineering	625	186	14
logic programming	4195	1081	30
mathematical physics	1268	199	9
operating systems	4509	1378	21
parallel processing	2776	1118	25
pattern recognition	3319	1054	34
robotics	6374	1246	30
spectroscopy	400	91	3
symbolic computation	753	330	9
wavelet	2641	457	10
Averages	2443.0	646.3	16.2

Evaluation Measure

We believe that when more survey articles in a domain are detected, it becomes more efficient for users to grasp the outline of the domain, because the survey articles may be written from different viewpoints, and comparison of such viewpoints is useful for deep understanding and taking a broad view of the domain. We therefore detect as many survey articles as possible.

Eleven-point Recall-Precision is the most typical evaluation measure in the IR community. We evaluate our systems by 11-point R/P using Equations (6) and (7). We also evaluate our system by the precisions of top-ranked documents, because survey articles are written for quickly grasping the outline of the domain, and should be detected in higher ranks. For the calculation of recall and precision, we made use of "trec_eval"(ftp://ftp.cs.cornell.edu/pub/smart), which is an evaluation tool developed for Text REtrieval Conference (TREC).

$$\text{Recall} = \frac{\text{The number of survey articles correctly detected by a system}}{\text{The number of survey articles that should be detected}} \quad (7)$$

$$\text{Precision} = \frac{\text{The number of survey articles correctly detected by a system}}{\text{The number of survey articles detected by a system}} \quad (8)$$

4.3 Results

We optimized values of L for each method to give the best precisions of top-ranked documents. The results are shown in Table 2. Using these values, we evaluated our systems by 11-point Recall-Precision, and by the precisions of the top-ranked documents. Both results are shown in Fig. 1 and Table 3, respectively.[7] The most striking result in Fig 1 is that WORD produces results that are remotely useful than those of other methods at recall = 0.2. CUE is second best, and both WORD and CUE improved the HITS algorithm. SIZE could also improve HITS when Recall is more than 0.1. DEVIATION and CITATION TYPE made HITS worse. In the evaluation by precisions of top-ranked documents (Table 3), both COMB and CUE are much superior to the others. We can also confirm that CUE and WORD could improve HITS significantly.

Table 2. The optimal values L of each method

Method	Values of L
CUE	15000-20000
DEVIATION	10^8
SIZE	1000
CITATION TYPE	10^5
WORD	10^7
COMB	CUE: 18000 DEVIATION: 0 SIZE: 0 CITATION TYPE: 10 WORD: 0

[7] As BASE-WORD and BASE-CUE collect all papers containing particular words (or cue phrases), and do not rank the results, we randomly ranked each result, and calculated the precision scores of both methods in Table 3.

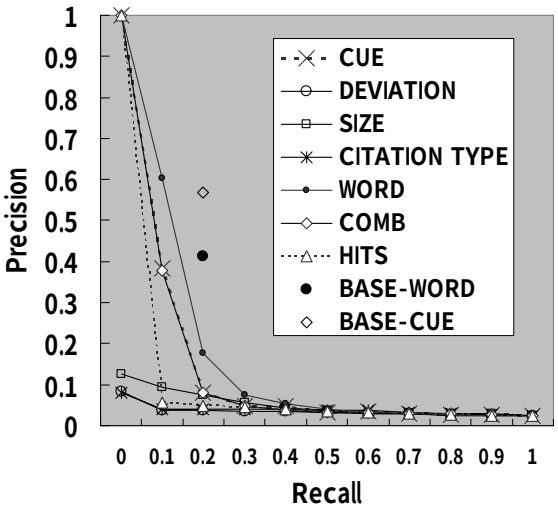

Fig. 1. Evaluation by 11-point Recall/Precision

Table 3. Evaluation by precisions of top-ranked documents

Top -n	Our methods						Baseline		
	Single feature					Multiple features	HITS	BASE-WORD	BASE-CUE
	CUE	DEVI-ATION	SIZE	CITE TYPE	WORD	COMB			
5	1.000	0.000	0.000	0.000	0.800	1.000	0.400	0.414	0.568
10	1.000	0.000	0.100	0.000	0.500	1.000	0.200	0.414	0.568
15	1.000	0.000	0.067	0.000	0.533	1.000	0.133	0.414	0.568
20	0.950	0.000	0.050	0.000	0.550	1.000	0.100	0.414	0.568
30	0.700	0.033	0.100	0.000	0.633	0.700	0.100	-	0.568
100	0.350	0.070	0.050	0.050	0.540	0.350	0.100	-	0.568

4.4 Discussion

Effectiveness of the HITS algorithm

From the results in Fig. 1, we can find that BASE-CUE has an outstanding abilitiy to detect survey articles by itself. However, CUE could never obtain precision scores of 1.0 at top-5, 10, and 15 without the HITS algorithm, because BASE-CUE detected non-relevant papers at rates up to 43.2%. In other words, HITS could exclude non-relevant documents from the result of BASE-CUE. We therefore conclude that the HITS algorithm is effective in detecting survey articles.

A list of cue phrases
As we could not prepare enough survey articles to apply statistical methods (e.g., n-gram) for the selection of cue phrases, we could only make a list of cue phrases. Fortunately, we found that our list of cue phrases was effective, although it may not be exhaustive. In future, we can add other cue phrases by applying statistical methods to survey articles that are collected automatically using our proposed method "COMB."

Parameter tuning
We could not confirm the effects of SIZE in the evaluation by precisions of top-ranked documents, though the precision scores of SIZE are superior to HITS, when the recall score is more than 0.1 in Fig. 1. We could not tune parameters of five features finely, because of the processing time. If we spent the time to examine the parameters more closely, we may confirm the effectiveness of SIZE.

5. Conclusions

In this paper, we proposed a method for detecting survey articles from a multilingual research paper database. We considered HITS, which is an algorithm to retrieve Web pages using the notions of authority and hub. It is considered that important papers and survey articles correspond to authorities and hubs, respectively. It is therefore possible to detect survey articles by applying the HITS algorithm to research paper databases, and selecting papers with outstanding hub scores. However, as HITS does not take account of the contents of each paper, the algorithm might detect papers citing many principal papers in mistake for survey articles. We therefore improved HITS by incorporating five features of survey articles. To investigate the effectiveness of our method, we conducted an experiment. We found that the HITS algorithm was useful for the detection of survey articles. We also found that cue phrases (CUE) could improve the HITS algorithm, and performed better than other methods.

6. Future Work

As the next step of this study, we need to measure the qualities of survey articles and to select the best one among detected candidates. Although it is confirmed in the experiment that our method is useful to detect comprehensive survey articles, the method does not guarantee that there are good-quality comments about the referring papers, and the task to measure the quality of such comments automatically seems very difficult. However, this problem may be resolved without analyzing survey articles by using NLP techniques. The limited resolution of this issue is to take account of the number of citations from other papers. Good-quality survey articles that contain good-quality comments are considered to be cited from many papers in the subject domain, and to have high authority scores. In our future work, we will

investigate with the relations between qualities of survey articles and their authority scores.

7. Acknowledgements

The authors would like to express our gratitude to anonymous reviewers for their suggestions to improve our paper. This work was supported in part by JSPS (Japan Society for the Promotion of Science) under the grant for Postdoctoral Fellowship.

References

[1] Cohn, D. and Chang, H. *Learning to probabilistically identify authoritative documents*. In Proceedings of the 17th International Conference on Machine Learning, pp.167–174, 2000.

[2] Kleinberg, J.M. *Authoritative sources in a hyperlinked environment*. In Proceedings of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 668–677, 1998.

[3] Lawrence, S., Giles, L., and Bollacker, K. *Digital libraries and autonomous citation indexing*. IEEE Computer, 32(6), pp. 67–71, 1999.

[4] McCallum, A., Nigam, K., Rennie, J. and Seymore, K. *Building domain-specific search engines with machine learning techniques*. In Proceedings of AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace. 1999.

[5] Nanba, H. and Okumura, M. *Towards multi-paper summarization using reference information*. In Proceedings of the 16th International Joint Conferences on Artificial Intelligence, pp. 926–931, 1999.

[6] Sparck Jones, K. and Van Rijsbergen, C.J. *Report on the need for and provision of 'ideal' test collections*. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.