Web Recommender System for Identifying Semantic related Scientific Papers

Manabu Ohta[1], Toshihiro Hachiki[1], Atsuhiro Takasu[2]
[1]Graduate School of Natural Science and Technology
Okayama University
Okayama, 700–8530
Japan
[2]National Institute of Informatics
Tokyo, 101-8430. Japan
{ohta, hachiki}@de.cs.okayama-u.ac.jp, takasu@nii.ac.jp

ABSTRACT: *Identifying the related scientific papers is the unfinished agenda in the scientific research. Keywords are used as signals in the search process where it represents the text and such keys that reflect the content of papers. These keys do not solve the complexities in identifying semantic content and hence we develop a web based recommender system which relies on technical terms. These terms lead to the generation of bipartite graphs that contain the significant words from text corpus and technical terms generated. They are further used to rank the related papers using HITS algorithm for analyzing the bipartite graphs. Experimental results on our method show promising results.*

Keywords: HITS algorithm, Web Search, Scientific papers processing, Semantic content, Key terms, Term Extraction

Received: 3 September 2011, Revised 17 December 2011, Accepted 23 December 2011

1. Introduction

Previous generation digital libraries were formed by the conversion of the print books to digital formats and capturing and converting of images to digital documents. [1]. In the same way scientific papers were converted and currently the digital libraries rely heavily on web platforms. These hybrid libraries stand as the transitions in digital and virtual library world. Although they are accessible on the Web, they still remain a digital version of traditional libraries. The Web contains thus the scientific information and serves a good digital repository. It is important to make a semantic links of all available scientific information so that web can be used a good platform for accessing scientific knowledge. By linking them, books and papers can provide information in more effective ways. For example, research papers contain many unfamiliar technical terms to novice researchers, undergraduate students, and people whose expertise differs from the domain of interest. It is, however, not efficient for them to use a dictionary or search the Web every time they encounter unfamiliar terms. Therefore, we proposed enhancing research papers of DLs with the other resources, i.e., the Web [2], and implemented a prototype online-browsing support system for research papers. Specifically, the proposed system searches the Web for explanatory Web pages of the unfamiliar terms and provides links to the explanatory pages.

In this work we planned to use the extraction techniques for identifying key terms that can be used basicall as seed terms. This work proposes further using the extracted technical terms to recommend research papers to a user that are related to the paper he or she is browsing. We use XML files of research papers with OCR markups to find technical terms because OCRed text

can be cheaply obtained from the scanned research papers in our DL. We first collect research papers related to the extracted technical terms by searching the DL and extract technical terms again from the collected research papers. Then, we generate a bipartite graph by assuming links from the collected papers to the technical terms appearing in them. We apply the HITS algorithm [3] for analyzing the bipartite graph to rank and recommend related papers.

In the related research literature collected using a browsed paper in this way, technical terms frequently appearing in them are considered important. Hence, papers written with a lot of such important technical terms are considered relevant to the browsed paper. In analysis of the proposed bipartite graph by HITS, we can consider relevant papers and important technical terms to correspond to hubs and authorities, respectively. It is, therefore, possible to recommend the most relevant papers by applying HITS to the bipartite graph and by selecting papers with the highest hub scores. As far as we know, the effectiveness of applying HITS to research paper recommendation in such a mode has not yet been examined.

We provide the framework of the paper as below. Section II briefly reviews the related work on research paper recommendation and the HITS algorithm. Section III introduces our paper recommendation method, and section IV gives experimental results to evaluate its performance. Section V summarizes the paper and mentions future work.

2. Related Work

2.1 Recommendation of Research Papers

We present a few significant related papers. Sugiyama et al. proposed scholarly paper recommendation using a user's latent research interests that exist in their publication list [4]. They used not only a researcher's past publications but also their neighboring papers such as citation and reference papers as context to build their research profiles. By experimentation, they verified the effectiveness of their approach for two classes of researchers: junior researchers who had only one recently published paper and senior researchers who had multiple past publications.

Ekstrand et al. presented and empirically tested a large collection of recommender algorithms for the task of generating an introductory reading list for a new researcher [5].

For user-based evaluation, they gave the recommender system a query set of five-to-ten research papers collected using a search tool and received a reading list consisting of five papers that were relevant to the query and important within the research literature. They augmented existing collaborative and content-based filtering algorithms with measures of the importance of a paper within the literature. They measured a node's importance in the citation graph using common algorithms, such as HITS [3] and PageRank [6]. They reported that collaborative filtering that used citation information generated such reading lists well.

Our proposed method, however, uses neither user profiles nor citation information, which differentiates it from the above work.

In addition, Song et al. proposed a learning framework for tag recommendation for scientific and Web documents [7]. They defined tagged training documents as triplets (words, docs, tags), and represented them in two bipartite graphs, which were partitioned into clusters. Tags in each topical cluster were ranked by their ranking algorithm. Their experiments on large-scale tagging datasets of research papers indicated that their framework effectively recommended tags in one second on average.

2.2 HITS Algorithm

HITS proposed by Kleinberg [3] is a major ranking algorithm for Web search results, which is often compared to another major one, PageRank, proposed by Page et al. [6]. HITS is also applied to finding communities on the Web. HITS discovers authority and hub nodes by analyzing links among them on the basis of the notion that the relationship a node has with important nodes affects the importance of the node more than that it has with less important nodes. In the context of Web analysis, authorities are pages having sufficient information on a specific topic, whereas hubs are ones that have sufficient links to such authoritative pages.

According to the HITS algorithm, the hub and authority scores for a node are iteratively calculated by the following equations:

$$a_p = \sum_{q,\ q \to p} h_q, \tag{1}$$

$$h_p = \sum_{q,\ p \to q} a_q . \qquad (2)$$

Note that $p \to q$ means node q is linked by node p . As iteratively calculating hub and authority scores starting with each node having a hub and authority score of 1 leads to diverging values, these scores must be normalized after every iteration. The final hub and authority scores are determined after repetitions of this process. Nodes are ranked in accordance with the final hub or authority scores.

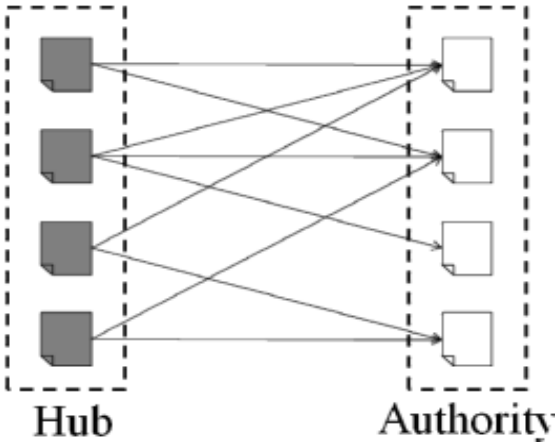

Figure 1. Reletionship between hub and authority

Kleinberg stated authoritative pages do not have links connecting each other, but are connected by hub pages that have links to multiple authoritative pages. Therefore, a good hub points to many good authorities while a good authority is pointed to by many good hubs, as shown in Figure 1.

As for application of the HITS algorithm, Nanba et al. proposed automatic detection of survey papers from a multilingual database using HITS [8]. They paid attention to the fact that important papers and survey papers respectively correspond to "*authority*" and "*hub*" in their citation relationship. They also modified HITS to improve accuracy of detecting survey papers by taking into account the contents of each paper.

3. Research Paper Recommendation

3.1 Outline

This part briefly describes how to recommend research papers relevant to a browsed one by using the technical terms extracted from it. We first collect related papers by searching our DL for every extracted technical term and then further extract technical terms from the related papers. We apply the HITS algorithm to ranking the related papers. HITS was originally used for ranking Web search results or finding Web communities. This paper, however, proposes applying HITS to link analysis of a bipartite graph consisting of related papers and technical terms appearing in them. By assuming each paper has links to every technical term appearing in it, paper nodes with only outlinks and technical term nodes with only inlinks constitute a bipartite graph.

1. The more frequently technical terms appear in the set of related papers, the more important the terms are to a browsed paper.
2. The more such important terms appear in a paper, the more relevant the paper is to the browsed one.

Figure 2. Assumptions on relationship between browsed paper and its related papers

For ranking related papers, we make two assumptions about the relationship between a browsed paper and its related papers as

shown in Figure 2. On the basis of these assumptions, we can consider that highly relevant papers and important technical terms correspond to good hubs and authorities, respectively. Therefore, we rank related papers in accordance with their hub scores and recommend top-ranked papers to a user.

In concrete terms, we define the following procedure:

(i) Extract technical term candidates from a browsed paper p_{target}, rank them using TF-IDF, and select K top-ranked terms as a set of technical terms. We describe this term extraction in detail in subsection *B*.

(ii) Retrieve related papers $p_{ij} \in P_i$ (j =1,..., N) using each extracted technical term $t_i \in T$ (i =1,..., K)as a query for our DL. Note here that our DL ranks papers in descending order of their citation counts. In addition, we collect at most N papers for each term where papers previously retrieved by other terms are not counted among the N papers. Note also that $P = \bigcup_{i=1}^{K} P_i$.

iii) Extract a set of technical terms T_{ij}^{a} from each retrieved paper p_{ij} in the same way as (i). Note here that $T_i^a = \bigcup_{i=1}^{N} T_{ij}^{a}$ and $T^a = \bigcup_{i=1}^{K} T_i^a$.

(iv) A bipartite graph is generated by linking the set of papers collected in (ii), P, to the set of technical terms extracted in (iii), T^a. Applying HITS to the bipartite graph makes it possible to rank the papers in P on the basis of their hub scores. We explain the application procedure in more detail in subsection *C*.

3.2 Technical Term Extraction

We morphologically analyze the text of research papers using a Japanese morphological analyzer Sen [9] to extract technical term candidates as feature terms in accordance with the following rules:

1. Extract all the nouns and unknown terms solely consisting of alphanumerics, *katakana*, or *kanji* as feature terms.
2. Concatenate the continuous feature terms (if any) into one feature term.
3. Remove from the above feature terms those terms i) solely consisting of numerics or *hiragana*, ii) consisting of one character, and iii) in our stopword list.

After extraction, we apply OCR error correction to the extracted feature terms. We utilize the query correction function of the Yahoo!JAPAN search engine [10], i.e., "Did you mean *guessed-corrected-term*". This query correction, however, is not always appropriate, especially for acronyms with various meanings. Therefore, the original feature term is corrected to the suggested query term only if the number of search results for the original term is less than 1,000. Then, we use TF-IDF to score all the extracted feature terms. The $tfidf_i$ of the term t_i is defined as follows:

$$tfidf_i = tf_i \times \log \frac{num}{df_i}, \tag{3}$$

where tf_i is the frequency count of the term t_i in the document from which *i t* is extracted, df_i is the document frequency of the term t_i, and *num* is the total number of documents. We define this df_i to be the number of papers retrieved by t_i, and regard *num* as the total number of papers stored in our DL. When we conducted the experiments described in section IV, the *num* was 13,206,916.

All the feature terms extracted from a research paper are ranked in accordance with this TF-IDF score, and the K topranked terms are selected as the technical terms used for retrieving related papers.

3.3 Applying HITS to Paper Recommendation

We first generate a bipartite graph as follows to apply HITS to related paper recommendation:

- Generate links from each paper to technical terms appearing in the paper.
- Regard papers and technical terms as hubs and authorities, respectively.

In analyzing the Web by HITS, each node can have both inlinks and outlinks and, hence, both hub and authority scores. In the

bipartite graph mentioned above, however, paper nodes have only outlinks and term nodes only inlinks by their definitions. Therefore, we assign only hub scores to paper nodes and only authority scores to technical term nodes.

Figure 3 illustrates a simple image of the bipartite graph generated by the proposed method. In this bipartite graph, we iteratively calculate authority scores of term nodes and hub scores of paper nodes by using equations (1) and (2), respectively. After each iteration, these scores are normalized by the following equations:

$$\sum_{p} a_p^2 = 1, \tag{4}$$

$$\sum_{p} h_p^2 = 1. \tag{5}$$

The iteration continues until the absolute difference in authority or hub scores between iterations becomes sufficiently small. Finally, we obtain lists of related papers and technical terms ranked by their hub scores and authority scores, respectively. Thus, we can recommend highly ranked related papers. If two or more papers have the same hub score, we rank them by evaluating the following metrics in this order:

1. The rank of a related paper in the search result of our DL.
2. The TF-IDF score of the technical term by which a related paper is retrieved.

3.4 Prototype Browser

Figure 4 shows the GUI of the implemented prototype browser for reading research papers. The left window shows major bibliographies such as title, authors, abstract, and keywords. The title and authors are in Japanese and English, but the abstract and keywords are only in Japanese because our system targets only Japanese papers at present.

The right window shows a list of recommended papers ranked by the proposed method. A paper on “Personalized Web Search” is displayed in the left window, and four recommended papers are visible in the right window in Figure 4.

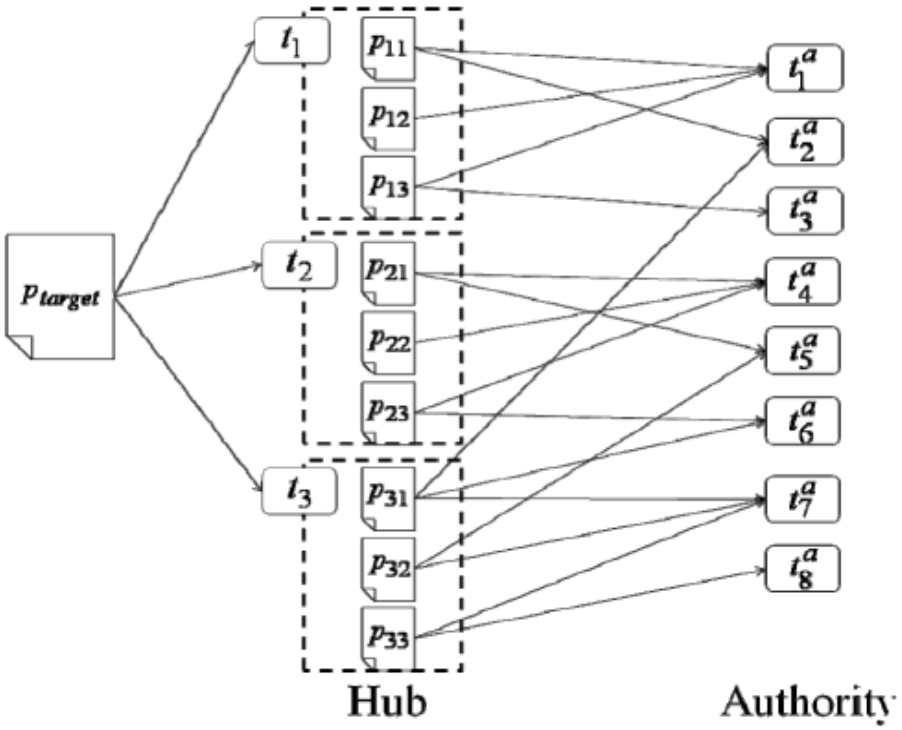

Figure 3. Bipartite graph to be analyzed by HITS

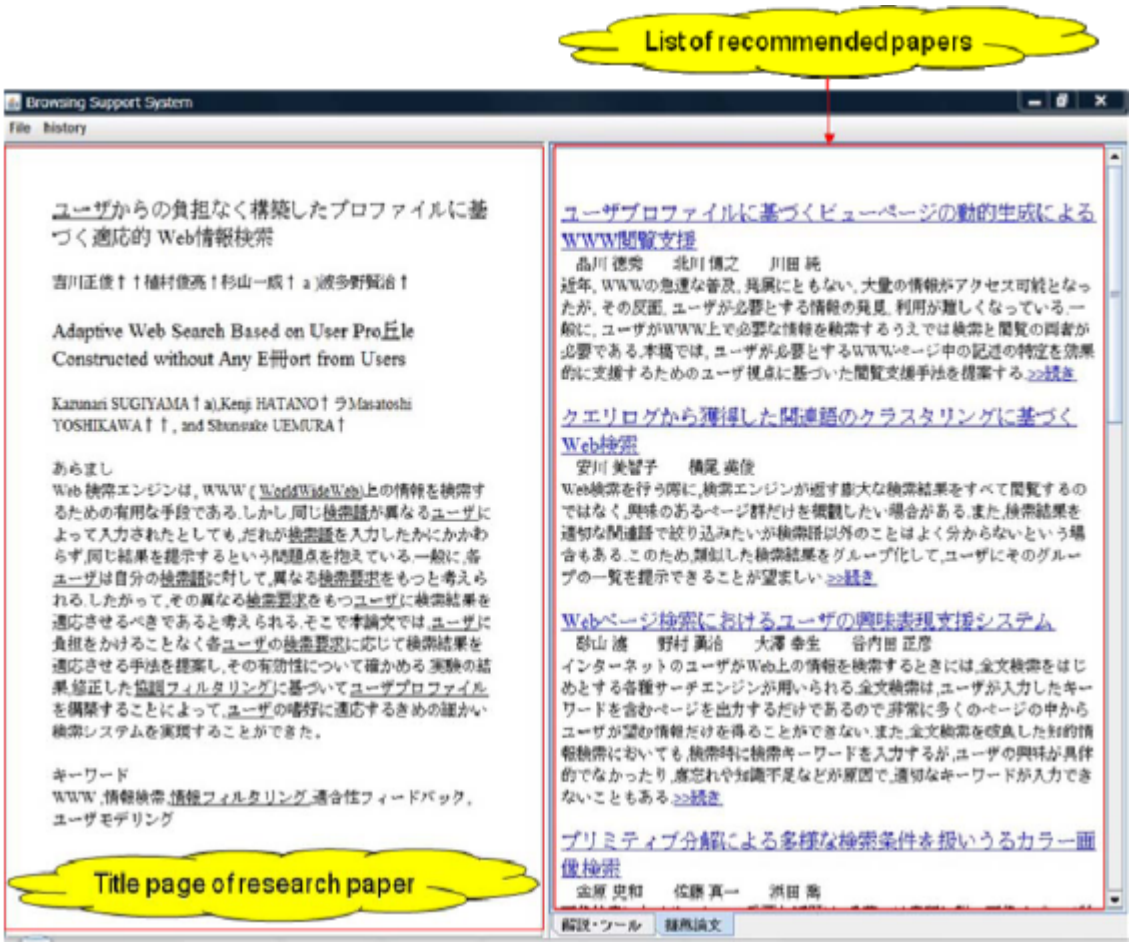

Figure 4. Prototype browsing support system with showing list of recommended papers

4. Experiment

4.1 Experimental Setup

First, we randomly selected 10 papers from six years worth of issues published by the Institute of Electronics, Information and Communication Engineers (from 2000 (Vol. J83-D-I) to 2005 (Vol. J88-D-I)). We evaluated performance on the basis of precision of top-ranked recommended papers such as precision at 10 recommended papers (p@10). Note that the experiments only used the Japanese title and abstract of papers. We, however, could extract technical terms from the whole paper in the same way. Moreover, we set the maximum number of extracted technical terms per paper because the more technical terms in a paper, the larger its hub score. We set this number to 10 on the basis of a preliminary experiment.

Second, our OCR system had good recognition accuracy. In recognizing a different Japanese academic journal, the accuracy was 99.00% for the abstract and 97.01% for the references. Mixtures of Japanese and English characters, as well as various fonts and punctuation symbols often appearing in the references, were difficult to recognize correctly.

In the experiment, one of the authors judged the relevance of the recommended papers to the browsed paper. We adopted rigid and relaxed relevance judgments in accordance with the degree of relevance. In the rigid judgment, only those papers that have the same purpose as the browsed one are relevant. In the relaxed one, papers that use the same technique are relevant in addition to those that have rigid relevance. For example, if a user browses a paper titled “Document classification using support vector machine”, papers on document classification are regarded as relevant in both rigid and relaxed judgments and papers reporting the use of support vector machine for another purpose are considered relevant only in the relaxed judgment.

4.2 Methods for Comparison

We implemented two recommendation methods for comparison with the proposed one.

• Vector space (VS) model-based method: This method is based on a VS model widely used in information retrieval. The VS model represents each paper as a vector of technical terms extracted from it. We define a similarity measure as the cosine of the angle between two paper vectors in order to recommend the papers most similar to a browsed one.

• Baseline method: This method directly reflects the assumptions shown in Figure 2. First, we score technical terms in related papers in accordance with their document frequencies. Then, we score each related paper a sum of the score of the technical terms appearing in the paper. Finally, we rank the related papers by their scores to recommend top-ranked papers to a user. The baseline method calculates scores of both papers and technical terms without iterative calculation, which differentiates it from the proposed HITS-based method.

4.3 Experimental Results

First, we show the number of related papers that can be collected from our DL with varying values of *N*, the maximum number of retrieved papers per technical term, in Table I. As explained in subsection *A*, we extract at most 10 technical terms from a paper. Hence, the maximum number of related papers is calculated as $10 \times N$. Table I shows the actual number of related papers is smaller than this because some papers have nine or fewer extracted terms and some technical terms have fewer than *N* related papers.

Next, we show the average indegrees of technical term nodes in the bipartite graph in Table II. Note here that the average number of technical terms refers to the average size of a set of technical terms, T^a, extracted from the related papers of the 10 papers selected for experiment. The terms with a large indegree affect hub and authority scores because such terms appear in many papers. Table II shows that the larger the *N*, the more technical terms are obtained, and the larger indegrees have the technical term nodes. The increase in indegrees is, however, subtle compared with that in the number of technical terms.

Moreover, we summarize some statistics of the technical terms with two or more indegrees in Table III because such term nodes play an important role in the proposed bipartite graph. The last row in Table III shows that the ratio of such terms to all the technical terms is from 6.1% to 9.2%, which means that more than 90% of extracted technical terms only appear in the one paper from which they are extracted.

Finally, Figure 5 plots the precision at 10 recommended papers of the VS model-based, baseline, and proposed methods with two relevance judgments w.r.t. the number of retrieved papers per term, *N*, of 5, 10, 30, and 50. As we can see from Figure 5, the proposed method achieved the best precision of 0.35 in the rigid judgment with $N = 5$ and of 0.90 in the relaxed one with $N = 10$. We can also see that the precision declines with the increase in *N* in the rigid judgment while the precision marks its highest value at $10 = N$ among 5, 10, 30, and 50 in the relaxed judgment. Compared with the other two methods, the proposed one performed better in the relaxed judgment irrespective of *N*. In the rigid judgment, the proposed method indeed achieved the highest precision of 0.35, but its precision worsens more than the other methods with $N = 30$ and $N = 50$.

# of retrieved papers/term (*N*)	5	10	30	50
Maximum # of related papers	50	100	300	500
Average # of related papers	37.9	69.3	178.7	269.0

Table 1. Average Number Of Related Papers

# of retrieved papers/term (*N*)	5	10	30	50
Average # of technical terms	346	615	1514	2228
Average indegree	1.09	1.12	1.17	1.19

Table 2. Average Indegreesr of Technical Terms t^a

# of retrieved papers/term (*N*)	5	10	30	50
Average # of technical terms	20.0	39.9	124.3	206.4
Average indegree	2.46	2 .74	3.00	3.09
Ratio of terms with two or more indegrees to all the terms (%)	6.05	6.53	8.32	9.21

Table 2. Average Indegreesr of Technical Terms t^a (≥ 2)

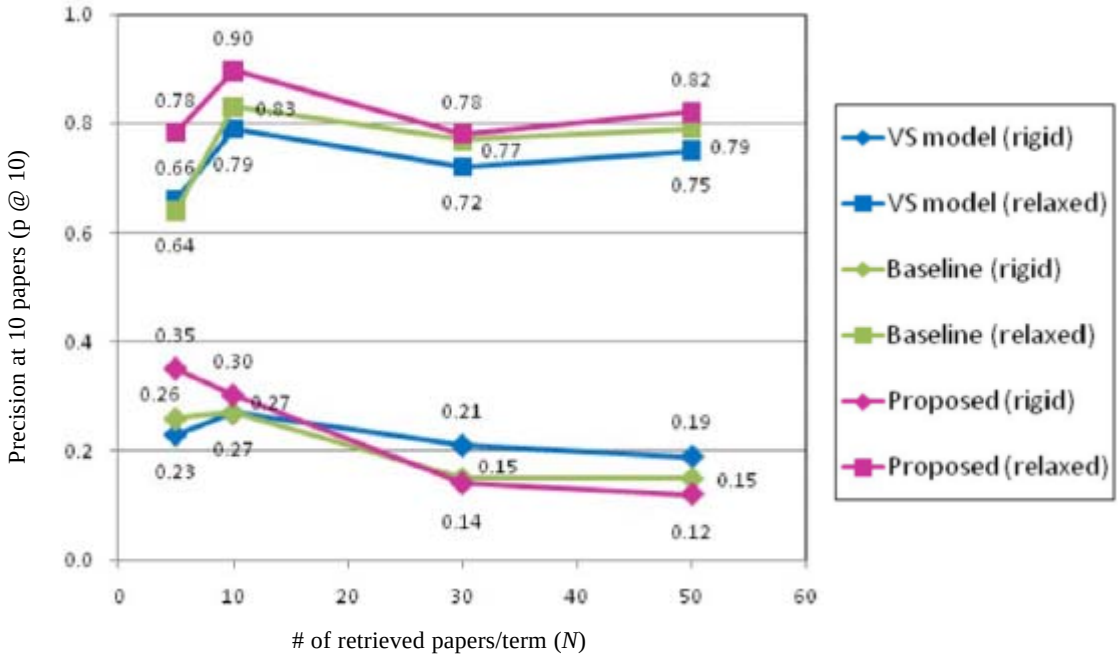

Figure 5. Precision of recommended papers

The VS model-based method achieved only low precision with a small value of *N* such as 5 irrespective of relevance judgment criteria. Although it achieved its best precision with *N* = 10, the values were the lowest among the three methods in both judgments.

On the other hand, the proposed method could recommend more relevant papers by using a small value of *N* such as 5, especially in the rigid relevance judgment. Using larger *N* of 30 or 50 did not improve precision. One of the major reasons for this is that inappropriate terms among the ones with high authority scores increased as *N* increased. The inappropriate terms are the extracted technical terms that are of little or no relevance to the main theme of a browsed paper and often badly affect paper recommendation. One of the remedies for this is considered to be weighting technical terms and related papers retrieved by the terms with the TF-IDF values defined by equation (3) because the terms and papers are respectively handled evenly once extracted or retrieved. We also need to sophisticate our technical term extraction procedure to select more appropriate terms.

When the proposed method was compared with the baseline one, the proposed method outperformed the baseline in most cases. The baseline method first calculates the scores of extracted technical terms on the basis of their document frequencies and then calculates the scores of papers. The proposed method takes into account linkage structure expressed in the bipartite graph in addition to the document frequency, which is considered effective for paper recommendation.

Figure 5 also shows that precisions in the relaxed judgment are more than double those in the rigid judgment irrespective of recommendation methods. In the rigid judgment, only papers with the same research purpose as the browsed one are judged as relevant recommendations. However, we were not always able to collect such papers sufficiently, which leads to this low precision. To improve precision in the rigid judgment, it may be effective to retrieve related papers by using plural technical

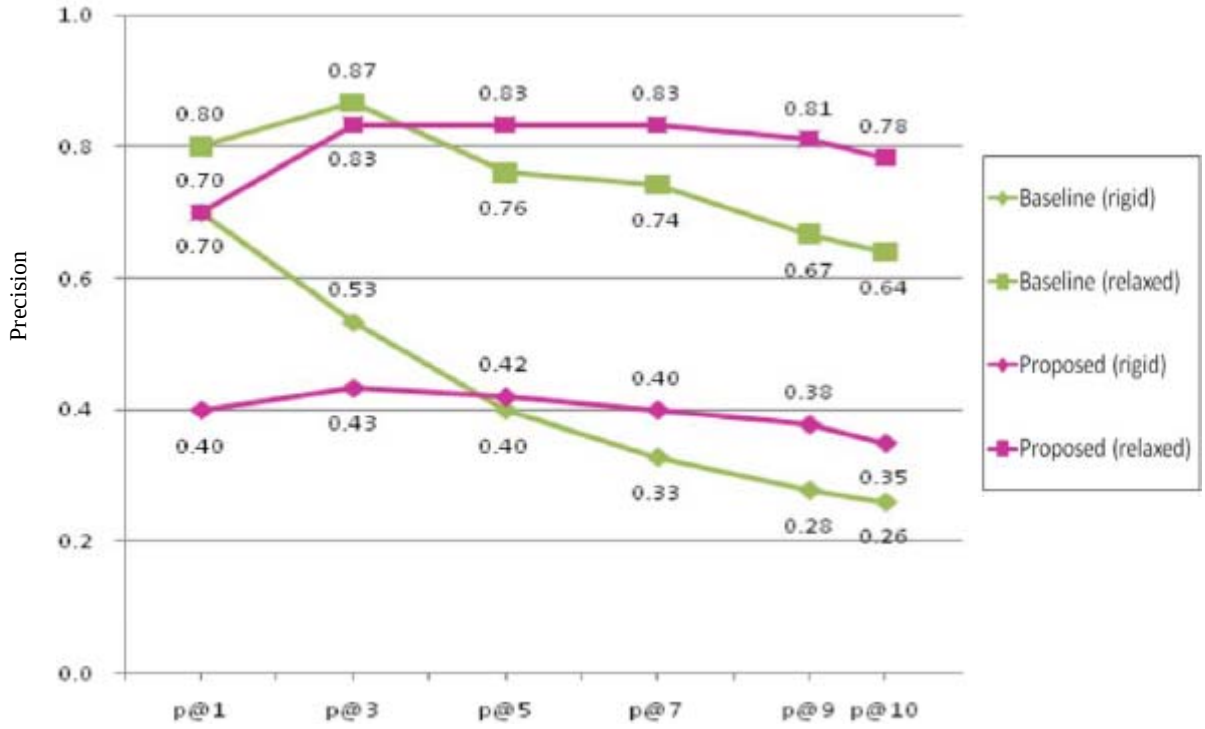

Figure 6. Comparison of proposed and baseline methods (N =5)

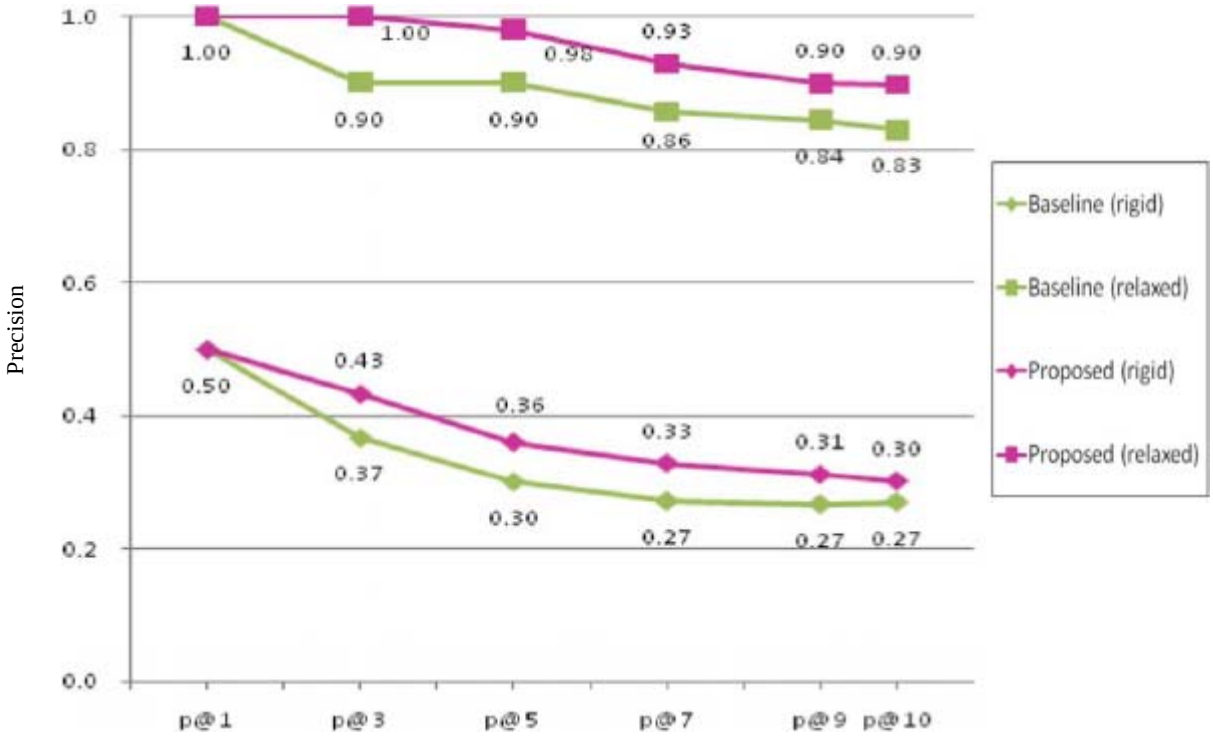

Figure 7. Comparison of proposed and baseline methods (N =10)

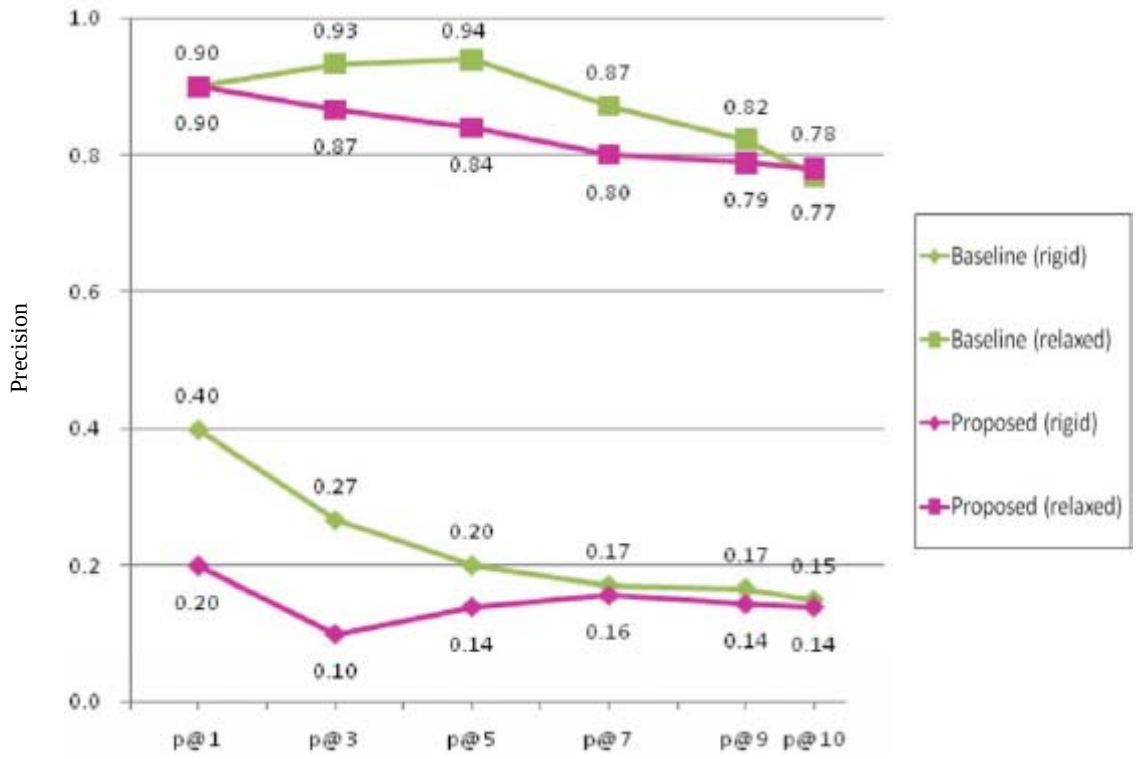

Figure 8. Comparison of proposed and baseline methods (N =30)

terms at the same time for Boolean search, because the proposed method as well as the other two methods collects related papers by using each technical term separately as explained in section III.

4.4 Comparison of the Proposed Method against the Baseline

We evaluated precision of highly ranked recommended papers by using p@1, p@3, p@5, p@7, and p@9 in addition to p@10 to further compare the proposed method against the baseline one. Figures. 6, 7, and 8 respectively show these precisions with N = 5, N = 10, and N = 30. The proposed method shows little differences among p @ X except for Figure 7 where the precision declines as X of p@X increases. The baseline method, on the other hand, tends to show a high precision with small X such as p@1 or p@3, which is especially prominent in Figure 6. That is, the baseline method can rank a few relevant papers highly and recommend them to a user while the proposed one outperforms the baseline when recommending more papers, say, 10 papers.

5. Summary

We have shown that our proposed a method to recommend related papers to support online-browsing of research papers is effective. The proposed method generates a bipartite graph by assuming linkage between related papers retrieved by the technical terms extracted from abrowsed paper, and technical terms appearing in the set of the related papers. It applies the HITS algorithm to analysis of the bipartite graph, then ranks the related papers in accordance with the hub scores assigned to the papers, and recommends top-ranked papers. We evaluated the precision of recommended papers in an experiment and showed that the proposed method could recommend relevant papers selected from a relatively small set of related papers more precisely than the other recommendation methods.

Although we only used titles and abstracts of papers in the experiment, we plan to use their whole contents to extract useful information such as bibliography. In such a case, citation information listed in the references is considered to be especially useful for paper recommendation [11]. As another venue for future work, we aim to embed the proposed functions into existent document browsers on the basis of the findings for the developed prototype browser for OCRed research papers.

6. Acknowledgements

We thank Wesco Scientific Promotion Foundation for its financial support. We also thank the Japan Society for the Promotion of Science (JSPS) for their Grant-in-Aid for Young Scientists (B), No. 23700119.

Note: The current version is the modified version of the paper presented at the Fourth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2011)

References

[1] Bunke, H., Wang ed, P. S. P. (1997). Handbook of character recognition and document image analysis, World Scientific.
[2] Ohta, M., Hachiki, T., Takasu, A. (2009). Using Web resources for support of online-browsing of research papers, *In:* Proc. of IRI, p. 348– 353.
[3] J. M. Kleinberg (1999). Authoritative sources in a hyperlinked environment, *Journal of the ACM*, 46 (5) 604–632.
[4] Sugiyama, K., Kan, M, Y. (2010). Scholarly paper recommendation via user's recent research interests, *In:* Proc. of the 10th annual joint conference on Digital libraries, p. 29–38.
[5] Ekstrand, M. D., Kannan, P., Stemper, J. A., Butler, J. T., Konstan, J. A., Riedl, J.(2010). Automatically building research reading lists, *In:* Proc. of RecSys 2010, p. 159–166.
[6] Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank citation ranking: bringing order to the Web, Technical Report, Stanford InfoLab.
[7] Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W., Giles, C. L. (2008). Real-time automatic tag recommendation, *In:* Proc. of SIGIR, p. 515–522.[8] Nanba, H., Okumura, M. (2005). Automatic detection of survey articles, *In:* Proc. of ECDL, p. 391–401.
[9] Sen Project, http://ultimania.org/sen/
[10] Yahoo!JAPAN, http://search.yahoo.co.jp/
[11] Shi, X., Leskovec, J., McFarland, D. A. (2010). Citing for high impact, *In:* Proc. of the 10th annual joint conference on Digital libraries, p. 49–58, 2010.