Using Link-Based Content Analysis to Measure Document Similarity Effectively

Pei Li[1,2], Zhixu Li[1,2], Hongyan Liu[3], Jun He[1,2], and Xiaoyong Du[1,2]

[1] Key Labs of Data Engineering and Knowledge Engineering, Ministry of Education, China
[2] School of Information, Renmin University of China, Beijing, China
{lp,lizx,hejun,duyong}@ruc.edu.cn
[3] Department of Management Science and Engineering, Tsinghua University, Beijing, China
hyliu@tsinghua.edu.cn

Abstract. Along with a massive amount of information being placed online, it is a challenge to exploit the internal and external information of documents when assessing similarity between them. A variety of approaches have been proposed to model the document similarity based on different foundations, but usually they are not applicable for combining internal and external information. In this paper, we introduce a link-based method into content analysis, which is based on random walk on graphs. By defining similarity as the meeting probability of two random surfers, we propose a computational model for content analysis, which can also be integrated with external information of documents. Empirical study shows that our method achieves good accuracy, acceptable performance and fast convergent rate in multi-relational document similarity measuring.

Keywords: link graph, content analysis, document similarity.

1 Introduction

Document similarity needs to be measured in a variety of applications for clustering, filtering, sorting and retrieving, etc. For example, in a personalized digital library, the computing of document similarity is the foundation of collaborative filtering [1]. But along with astonishing amount of information being placed online, the computation of document similarity encounters great challenges in two aspects: (1) the complexity of internal and external information of a document; (2) the large scale of document amount. For this reason, the ability to measure similarity between documents in an accurate and efficient way is a key determinant for many applications.

A variety of approaches have been proposed to model document similarity based on different foundations. Some traditional approaches calculate similarity according to document contents (especially document-term relationship), such as Vector Space Model [2], *n*-gram measures [3] and Latent Semantic Analysis [4], etc. Recently, by exploiting link structure of objects, some methods focusing on link-based object ranking are proposed by researchers [5, 6, 7]. If viewing documents as nodes and relationships among documents as edges, document similarity can be measured by these link-based object ranking methods with the contents of documents ignored.

Q. Li et al. (Eds.): APWeb/WAIM 2009, LNCS 5446, pp. 455–467, 2009.

In this paper, we propose a new approach by using link-based content analysis to measure document similarity effectively. This approach takes advantage of document-term relationship and builds a link graph among documents. Then link analysis is imported to assess the similarity between documents. There are plenty of link analysis methods introduced in [12]. Our approach propagates the similarity between documents with a certain transition probability, and has a theoretical foundation based on random walk theory [9]. Moreover, internal and external information of documents can be combined effectively using our method, which is not applicable for most similarity measuring methods mentioned above.

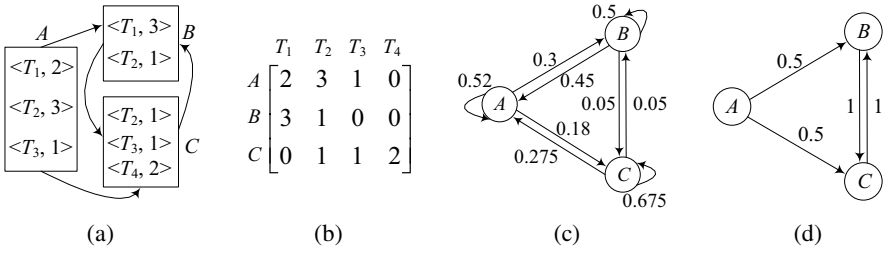


Fig. 1. (a) Documents and citation relationship. (b) Document-term relationships. (c) Transition probability based on contents. (d) Transition probability based on references.

As Figure 1(a) shows, the content of a document can be modeled as a collection of <*Term*, *TF*> pairs, where *TF* is term frequency in the corresponding document. Using some simple statistics, we obtain a matrix describing relationship between documents and terms as seen in Figure 1(b). Transition probability based on contents between documents is derived from relationship matrix, via normalization step and matrices product step (explained in Section 3.2 and shown in Figure 1(c)). External information is also considered (discussed in Section 4). From the citation relationship shown in Figure 1(a), we can get transition probability shown in Figure 1(d). Given a ratio of importance between contents and citation relationship in measuring the similarity, we can combine these two kinds of transition probabilities as a whole, which is the input of similarity computation. Based on random walk theory, we define similarity as the meeting probability of two surfers. Given a document-pair (*a, b*), its similarity is determined by similarities of all its direct-connected document-pairs and transition probabilities from (*a, b*) to these pairs (details can be found in Section 3.3).

The internal information (that is content) and external information (namely, outside links) of documents can be combined effectively using the same computational model. Experiments on real datasets are conducted to test the accuracy and performance of our link-based content analysis. Observation on convergence indicates that similarity result of the first iteration is acceptable for most cases.

The rest of the paper is organized as follows. Section 2 surveys related work. In Section 3, we introduce link-based content analysis. Afterwards, Section 4 describes how to combine internal and external information. The results of experiments are shown in Section 5 and this study is concluded in Section 6.

2 Related Works

Measuring pair-wise document similarity has been extensively studied for decades, with lots of methods proposed. These methods can be roughly divided into several types according to the document information they focus on. Generally, related works of document similarity measuring can be considered from three views: content analysis, link analysis and their combination.

(1) Content analysis

Analysis of content (or internal information) is a traditional information retrieval task. Early methods treat a document as a bag of words and calculate cosine similarity according to the *tf-idf* weight [13], such as Vector Space Model and its variations [2, 14]. Considering the sequence of terms, *n*-grams [3] are introduced to gauge the similarity. A more complicated approach is Latent Semantic Analysis [4], which maps each document and term vector into a lower dimensional space associated with concepts. Aslam et al. [15] propose an information-theoretic measure for document similarity in an axiomatic manner, which is a different research route from others.

(2) Mining links of documents

Some documents (especially web pages) have rich outside links (or called external information). Viewing documents as nodes and external relationships as edges, the corpus of documents can be modeled as a graph, and exploiting this link structure may be one of the best ways to measure document similarity. There are multiple link-based object ranking methods. *SimRank* [5] provides a wonderful definition for similarity on a link graph and takes $O(N^2)$ time for each iteration. The intuitive underlying model of *SimRank* is "random surfer-pairs", a concept derived from random walk theory. Xi et al. [6] use a Unified Relationship Matrix (URM) to represent a collection of heterogeneous objects and their relationships, and compute the similarity iteratively over the URM. Besides, Yin et al. [7] proposed a hierarchical structure called *SimTree* to represent similarities between objects in a compact way. These link-based object ranking methods usually involve iterative computation and ignore the inside attributes of an object.

(3) Combining content and outside links

Researchers have introduced techniques for combining link-based and content-based methods to improve the accuracy of web document classification [16]. Multiple models are developed. Jin et al. [17] introduce a probabilistic model that integrates content matching and link information in a single unified framework to improve retrieval. Zhu et al. [18] design an algorithm that carries out a joint factorization on both the linkage adjacency matrix and the document-term matrix, and represents web pages in a low-dimensional space.

The intuitive model of our method is based on random walk on graphs, which is a special Markov Chain [10]. We apply link-based idea to content analysis, and integrate it with outside links naturally, thus making it different from other methods.

3 Link-Based Content Analysis

In this section we introduce link-based content analysis method.

3.1 Modeling the Content of a Document

Before modeling the content of a document, a cleaning step is performed to remove stop-words, and stem leftover words using the Porter Stemmer algorithm [19]. The stop-words list is taken from SMART Retrieval System [20], and a complete manifest can be found in [21].

We model the content of a document as a set of <*Term*, *TF*> pairs, where *TF* is the frequency of this term occurring in this document. Formally, let *D* denote the content of a document. Supposing that *D* contains *n* different terms, we describe *D* by

$$D = \{< Term_1, TF_1 >, \cdots, < Term_n, TF_n >\} \tag{1}$$

For a corpus of documents, we give simple definitions for some terminologies which will be used in the rest of this paper.

Definition 1. (**Document Vector**) The document vector of a corpus is an ordered array of all document objects in this corpus. Let *DV* denote it.

Definition 2. (**Term Vector**) The term vector of a corpus is an ordered array of all *different* terms in this corpus. We signify it by *TV*.

Obviously, the relationship between document vector and term vector can be modeled as a weighted bipartite graph with two disjoint sets corresponding to documents and terms. A relationship matrix can be deduced from this bipartite graph, with |*DV*| rows and |*TV*| columns, where |*DV*| denotes the number of documents in *DV* and analogously for |*TV*|. Let *R* denote relationship matrix.

We take a corpus example of documents shown in Figure 2(a) for the convenience of following analysis. Note that term T_4 only occurs in document *C*. We present the bipartite graph corresponding to corpus example in Figure 2(b). The relationship matrix between documents and terms is easy to obtain in Figure 2(c).

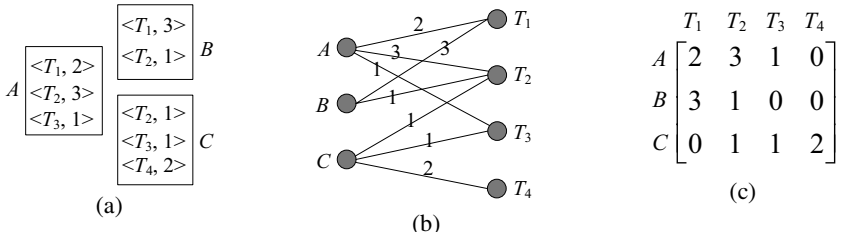


Fig. 2. (a) A corpus example of documents. (b) Weighted bipartite graph. (c) Corresponding relationship matrix *R*.

3.2 Transition Probability between Documents

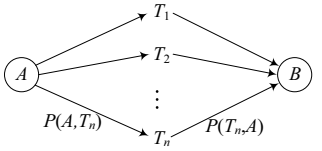
From the viewpoint of Random Walk Theory, given a directed unweighted graph *G*(*V, E*), supposing there is a random surfer standing on node *a*, he has identical possibility to visit each node at which he can arrive on next step, and *zero* possibility to

other nodes. If it is a weighted graph, the possibility will be adjusted according to edge weight. This possibility is entitled "transition probability" by researchers.

Treating every document in *DV* as a node and transition probability of each pair of documents as an edge, the corpus of documents can be viewed as a graph. However, the transition probability between two documents can't be ascertained directly. The reason is, from the view of a random surfer in document *A*, he do not know any internal information (the content) of another document *B*. He only knows the terms in *A*. So our solution is using terms as the bridge between two documents.

A normalization step is performed before computing of transition probability. From the probability theory, transition probabilities of a node to all other nodes should sum to 1. For a matrix *M*, we use M_N to denote the normalized matrix. The terminology "*transition matrix*" is defined by researchers to represent the normalized probability matrix from object sets O_1 to O_2, and we use $T(O_1, O_2)$ to denote it. Moreover, given the *i*-th element *a* in O_1 and the *j*-th element *b* in O_2, let $P(a, b)$ denote the element in the *i*-th row and the *j*-th column of $T(O_1, O_2)$.

Base on the above discussions, the transition matrix from documents to terms can be described as $T(DV, TV) = R_N$, and the transition matrix from terms to documents can be described as $T(TV, DV) = (R^T)_N$, where *R* is given in Section 3.1 and R^T is the transpose of *R*.



(a)

	A	B	C
A	0.52	0.3	0.18
B	0.45	0.5	0.05
C	0.275	0.05	0.675

(b)

Fig. 3. (a) The computation model of $P(A, B)$. (b) Transition matrix of the corpus example.

We model the computation of transition probability from document *A* to *B* as two steps (shown in Figure 3(a)). The first step is a random walk from document *A* to some term T_n with transition probability $P(A, T_n)$, and the second step is another random walk from term T_n to document *B* with transition probability $P(T_n, B)$. Serial steps mean a multiplication. Considering all terms including T_n , we take

$$P(A,B) = \sum_{n=1}^{|TV|} P(A,T_n) \cdot P(T_n,B) \tag{2}$$

to represent the transition probability from document *A* to *B*, where $|TV|$ is the number of different terms in term vector *TV*.

Considering the transition matrix of documents, supposing *A* and *B* are the *i*-th and the *j*-th documents in *DV* respectively, we get $P(A, B) = T_{ij}(DV, DV)$, $P(A, T_n) = T_{in}(DV, TV)$ and $P(T_n, B) = T_{nj}(TV, DV)$. Thus we have the following theorem.

Theorem 1. The transition matrix of documents, $T(DV, DV)$, can be computed by

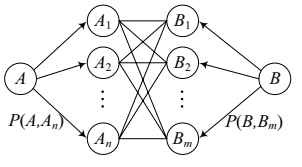
$$T(DV,DV) = T(DV,TV) \cdot T(TV,DV) \tag{3}$$

Proof. From Equation (3) we can get $T_{ij}(DV,DV)=\sum_{n=1}^{|TV|}T_{in}(DV,TV)\cdot T_{nj}(TV,DV)$, which can be also obtained from Equation (2).

Taking the corpus shown in Figure 2(a) as an example, we can obtain *T*(*DV*, *TV*) and *T*(*TV*, *DV*) using relationship matrix *R* shown in Figure 2(c). The transition matrix of documents is computed by Equation (3) and shown in Figure 3(b) with another view in Figure 1(c). Note that the transition probability from a document to itself is not equal to *zero*, which means the random surfer has possibility to stay in his current position on next step.

3.3 Assessing Similarity

Let *Sim*(*A*, *B*) denote the similarity between object *A* and *B* in a link graph. If *A* = *B*, we define *Sim*(*A*, *B*) = 1, otherwise we define *Sim*(*A*, *B*) as the meeting probability of two random surfers starting from *A* and *B* respectively (similar definition is found in [5]). It is easy to get *Sim*(*A*, *B*) = *Sim*(*B*, *A*) according to this definition.



(a)

	A	*B*	*C*
A	1.0	0.585	0.524
B	0.585	1.0	0.486
C	0.524	0.486	1.0

(b)

Fig. 4. (a) The computation model of *Sim*(*A*, *B*). (b) *S*(*DV*) of the corpus example.

In our method, the meeting probability (or similarity) is reinforced step by step. Each step of random surfers means a re-distribution of meeting probabilities, and results of the (*k*+1)th step are based on results of the *k*-th step. Assessing similarity *Sim*(*A*, *B*) is an iterative process. We model the (*k*+1)th iteration of *Sim*(*A*, *B*) (we denote it by $Sim_{k+1}(A,B)$) in Figure 4(a). Supposing document *A* has *n* outgoing edges and *B* has *m* outgoing edges, there are n×m document-pairs needing to be considered. Let *d* be a decay factor (usually *d* = 0.8) and we represent $Sim_{k+1}(A,B)$ by

$$Sim_{k+1}(A,B)=d\cdot\sum_{i=1}^{n}\sum_{j=1}^{m}P(A,A_i)\cdot Sim_k(A_i,B_j)\cdot P(B,B_j),\ \text{where } A\neq B \tag{4}$$

For the convenience of derivation, we take the objects not in $\{A_1, A_2, ..., A_n\}$ or $\{B_1, B_2, ..., B_m\}$ into consideration. Since transition probabilities from *A* or *B* to these objects are zero, they make no contribution to $Sim_{k+1}(A,B)$. Denoting similarity matrix of the *k*-th iteration by $S_k(DV)$ and supposing *A* and *B* are the *p*-th and the *q*-th documents in *DV* respectively ($p \neq q$), we get

$$(S_{k+1}(DV))_{pq}=d\cdot\sum_{i=1}^{|DV||DV|}\sum_{j=1}T_{pi}(DV,DV)\cdot(S_k(DV))_{ij}\cdot T_{jq}(DV,DV) \tag{5}$$

Based on Equation 5, we give a theorem as follows.

Theorem 2. The (k+1)th iteration of similarity matrix $S_{k+1}(DV)$ can be computed by

$$S_{k+1}(DV) = d \cdot T(DV, DV) \cdot S_k(DV) \cdot (T(DV, DV))^T + M \quad (6)$$

where $(T(DV, DV))^T$ is the transpose of transition matrix $T(DV, DV)$, and M is a correction matrix making every element on diagonal of $S_{k+1}(DV)$ to be 1.

Proof. In fact, Equation (6) is the matrix form of Equation (5).

A notice here is the initialization of $S_k(DV)$ when $k = 0$. Since we can't foreknow the similarity between two objects before iteration, it is reasonable to simply define $(S_0(DV))_{ij} = 1$ for $i = j$, and $(S_0(DV))_{ij} = 0$ for $i \neq j$. Hence, $S_0(DV)$ is an identity matrix and is symmetrical, which ensures $Sim(A, B) = Sim(B, A)$ on all iterations.

Our similarity computation method can be viewed as an extension of *SimRank* [5] on directed weighted graph. The naive method of *SimRank* is only applicable for undirected unweighted graph. Besides, the starting points of our method and *SimRank* are different. *SimRank* measures the Expected-*f* Meeting Distance on a graph, while our method evaluates the meeting probability of two random surfers. *SimRank* gets an iterative formula similar to Equation (4), and the mathematic proof of convergence given in the Appendix of [5] is also adaptable for our method with a few changes.

Let's consider the example shown in Figure 2(a). Using the transition matrix shown in Figure 3(b), we can compute its similarity matrices easily by Equation (6), and the convergent result $S(DV)$ (via 10 iterations) is presented in Figure 4(b).

We summarize the major steps of link-based content analysis method as follows.

1. Preprocessing. Remove stop-words from document contents and stem the left words (Section 3.1).
2. Obtain relationship matrix R between documents and terms using statistical technologies (Section 3.1).
3. Compute transition matrix of documents $T(DV, DV)$ by Equation (3) (Section 3.2).
4. Initialization. Set $k = 0$ and $S_0(DV)$ to be a $|DV|$-by-$|DV|$ identity matrix.
5. For the (k+1)th iteration, compute $S_{k+1}(DV)$ using Equation (6) (Section 3.3).
6. If $S_{k+1}(DV)$ is not convergent when compared with $S_k(DV)$, let $k = k+1$ and then jump to step 5; else return $S_{k+1}(DV)$.

3.4 Complexity Analysis

For simplicity, we suppose $|DV| = n$ and $average(|D|) = m$ in a corpus of documents, and the average outgoing edge number of a node is d. The time cost on our link-based content analysis can be roughly divided into two parts: (1) Computation of transition matrix. According to Equation (2), the time consumed by transition matrix computation is $O(mn^2)$. (2) Iterative computation of similarity matrix. According to Equation (4), supposing the number of iterations is k, time complexity for iterative computation is $O(kd^2n^2)$. In real datasets, m and d are usually constants, and $k < 10$ in most cases.

There are some methods for improving performance of *SimRank*, such as pruning *SimRank* [5], fingerprint *SimRank* [11], etc. These methods are also suitable for

improving the performance of our link-based content analysis, but it is not the key point of this paper. The aim of our work is to introduce a link-based method into content analysis, and combine internal and external information (introduced in Section 4) for more accurate similarity measuring. In addition, the convergence feature studied in Section 5.3 indicates that the result is acceptable when $k = 1$.

4 Combining Internal and External Information

Relationships between documents can be explored from different aspects, for instance references, publication date, authors, and so on. The best way to describe external information of documents is using a link graph. Extensive studies have been performed on exploiting the link structure such as web graph. In this section, we focus on how to integrate external information into link-based content analysis.

4.1 Integration of Transition Matrices

Usually, there is more than one kind of relationships between documents, and we only consider the ones independent with each other. For example, references and authors are independent, while references and "cited-by" are not. A relationship matrix R can be obtained according to each relationship. Similar with link-based content analysis in Section 3.2, the transition matrix between documents can be described as $T(DV, DV) = R_N$. Taking documents shown in Figure 1(a) as an example, transition probabilities of citation relationship are shown in Figure 1(d).

For a corpus of documents, supposing there are K different relationships between documents, we can get K different transition matrices $T_1(DV, DV)$, $T_2(DV, DV)$, ..., $T_K(DV, DV)$. Considering the transition matrix computed by link-based content analysis (let $T_0(DV, DV)$ denote it), we give different weights to these K+1 transition matrices and compute a weighted mean matrix. That means the integrated transition matrix $T(DV, DV)$ can be calculated by the following formula.

$$T(DV, DV) = \sum_{i=0}^{K} w_i \cdot T_i(DV, DV) \tag{7}$$

where w_i is the weight of $T_i(DV, DV)$ and $\sum_{i=0}^{K} w_i = 1$. Then, similarity matrix $S(DV)$ is computed iteratively using Equation (6).

4.2 Estimating Weights

In Equation 7, the weights are usually determined by experts. That is not a perfect solution, because different people have different evaluations of importance. Inspired by machine learning methods such as decision tree and neural networks, we design an approach to learn weights from training dataset.

Usually we take a portion of classified documents (e.g. 10%) as training dataset. For a transition matrix $T_i(DV, DV)$, we get corresponding similarity matrix $S(DV)$ by Equation 6 and cluster documents based on $S(DV)$. Correct classified documents are

counted (we denote the sum by C_i) and let $accuracy(T_i)$ be the ratio between C_i and $|DV|$. The weight of $T_i(DV, DV)$ can be estimated by

$$w_i = accuracy\ (T_i) \Big/ \sum_{x=0}^{K} accuracy\ (T_x) \tag{8}$$

5 Empirical Study

We have described a method for link-based content analysis and a solution to combine internal and external information. In this section, the accuracy and iterative process of similarity propagation will be tested, compared with (1) *VSM* [2], a traditional content-based method. We implement it strictly following this equation $Sim(D_1, D_2) = \cos\theta = \sum_{k=1}^{|TV|} W_{1k} \times W_{2k} \Big/ \sqrt{(\sum_{k=1}^{|TV|} W_{1k}^2)(\sum_{k=1}^{|TV|} W_{2k}^2)}$, where W is *tf-idf* weight of a term in a document. (2) *SimRank* [5], a link-based approach on graph.

The similarity score between two objects is hard to ascertain without performing extensively user studies. ACM Computing Classification System [22] (CCS) is a credible subject classification system for Computer Science, which provides an identification of similar papers by organizing these papers in the same category. Note that CCS is only a rough evaluation of similarity. Our dataset is crawled from three categories in ACM CCS, and contains 5469 documents with the ineffectual papers removed. There are three kinds of information in this dataset: (1) Document-term information. Terms are extracted from ABSTRACT of each paper. This relationship is used for content analysis. (2) Reference information. If document *A* refers to document *B*, there is an edge from *A* to *B*, so we can construct a citation relationship which is a directed weighted graph. (3) Author information. If *A* and *B* share the same author, an edge between *A* and *B* exists. Thus author relationship can be described as an undirected weighted graph. More details are listed in Table 1.

All experiments are performed on a PC with a 1.86G Intel Core 2 processor, 2 GB memory, and Windows XP Professional. All algorithms are implemented using Java.

Table 1. Details of each relationship

Information	Details (5469 documents in total)
document-term	11,312 different terms; each document contains 43.9 terms on average, and a term exists in 21.2 documents on average
reference	29,304 references in total; 5.4 references per paper
author	7238 different authors; each author appears in 2.2 documents and there are 1.7 authors per document on average

5.1 Similarity Evaluating

In this section we report experiments to examine the accuracy of our method, compared with others. For the given ACM dataset, we compute the similarity matrix of

documents in five different ways. To be specific, based on document-term relationship, we obtain two similarity matrices using *VSM* and our link-based content analysis respectively. Another two are obtained by using *SimRank* on reference relationship and author relationship. At last, by combining internal and external information, we obtain a holistic similarity measuring of this dataset.

We use PAM [8] to cluster papers into groups based on similarity matrix. Comparing these groups with CCS categories, we define accuracy as the maximal ratio between the number of correct classified papers and the total number of papers.

Table 2. Accuracy of different approaches

Approach	Max Accuracy
VSM	0.4452
Link-Based Content	0.5072
SimRank (reference)	0.5009
SimRank (author)	0.4876
Combination	0.5681

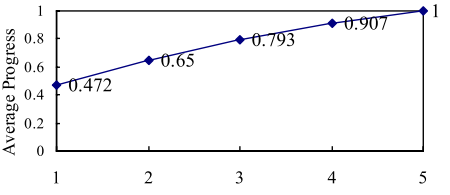


Fig. 5. Average progress of each iteration

Table 2 shows the accuracy of different approaches. The accuracy of *VSM* is not very good compared with other methods, mainly because *VSM* behaves weak in large amount of short documents (e.g. web pages). Anyway, our link-based content analysis gets comparable accuracy with *SimRank*. Then, we compute weights via accuracy according to Equation 8, and obtain a combination of internal and external information by Equation 7. The accuracy of this combination is 56.81% and higher than others, which means a more accurate similarity measuring.

To observe the reinforcement of similarity in iterative process of our link-based content analysis, we take $average(Sim_k(A,B)/Sim(A,B))$ to denote the average progress of each document-pair on the k-th iteration. In Figure 5, the similarity score increases iteration by iteration, and the rising amplitude of each iteration indicates the effect of this iteration to final convergent similarity.

5.2 Performances

We have discussed issues about time complexity in Section 3.4. Supposing there are n documents and a document has m terms on average, *VSM* takes $O(mn^2)$ time and for the worst case, time complexity is $O(|TV|n^2)$. In comparison, our link-based content analysis takes $O(kd^2n^2)$ time, where d is the average number of outgoing edges.

Table 3. Performances of different approaches

Approach	Time/Iteration (sec)	Iteration Num	Total (sec)
VSM	1665	1	1665
Link-Based Content	5858	5	29289
SimRank (reference)	1101	9	9908
SimRank (author)	1030	9	9270
Combination	5519	5	27593

Table 3 lists performances of different approaches. We can see link-based content analysis takes longer time than *VSM* and other link-based methods. That is because viewing terms as bridges between documents, outgoing edge number of a document is usually big. As we have said before, there are some technologies to improve the performance of our method. In next subsection, convergence feature indicates that we only need to perform the first iteration when using link-based content analysis. Moreover, our link-based content analysis can be combined with external information such as references and authors. Transition matrix of this combination is dense too, which results in a performance similar to link-based content analysis.

5.3 Convergence Feature

Iterative process is common for link-based methods based on random walk on graphs. In this set of experiments, the convergence rate will be measured from two aspects: maximum difference and accuracy. If let $Sim_k(A, B)$ denote the similarity score on the k-th iteration, maximum difference M_k can be described as $max(|Sim_{k+1}(A,B) - Sim_k(A,B)|)$, where (A, B) is an arbitrary pair of documents. When M_k is less than the tolerance factor of convergence (e.g. 0.001), iterative process will stop.

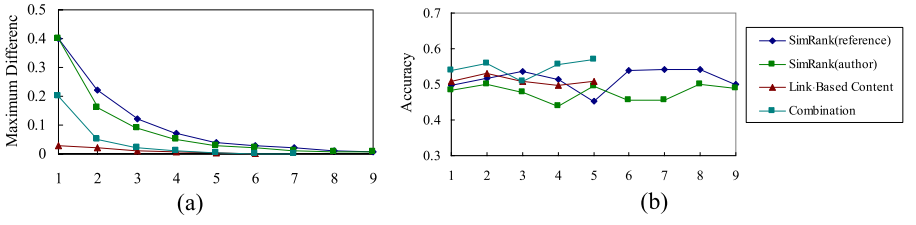


Fig. 6. (a) Maximum differences of different approaches. (b) Accuracy in iterative process.

We get maximum difference and the accuracy of each link-based method on each iteration shown in Figure 6(a) and Figure 6(b) respectively. Comparing with other methods, link-based content analysis has remarkable advantage on the first iteration. In Figure 6(a) the maximum difference M_1 of link-based content analysis is 0.03, which means the increase of similarity score on latter iterations is very limited. The reason is that dense links between documents can accelerate similarity propagation and result in faster convergence rate (but also consume more time on each iteration).

Based on this feature, we can draw a conclusion that in most cases, similarity computed by the first iteration of link-based content analysis is acceptable for most applications. That usually means $Sim_1(A,B) \approx Sim(A,B)$ and we can replace $Sim(A,B)$ by $Sim_1(A,B)$ to avoid expensive computing cost of iterative process.

6 Conclusion

In this paper, we introduce a link-based method to content analysis, and by exploiting document-term relationship, we propose a link-based content analysis to measure

document similarity iteratively. Moreover, our link-based content analysis can be combined with external information to obtain a more accurate similarity measuring.

The contributions of our work are summarized as follows.

- We introduce a link-based method into content analysis research. Traditionally, link analysis and content analysis are mutual noninterference, and they differ in research routes and theory models. Our method is a link-based content analysis based on random walk on graphs.
- The internal and external information of documents can be combined effectively using the same computational model. That means our method not only suits for content analysis (utilizing internal information), but also is applicable for utilizing external information such as references, authors and publication date, etc.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 70871068, 70621061, 70890083, 60873017, 60573092 and 60496325.

References

1. Renda, M.E., Straccia, U.: A Personalized Collaborative Digital Library Environment: a model and an application. Information Processing and Management 41(1), 5–21 (2005)
2. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18, 613–620 (1975)
3. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science 267, 843–848
4. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)
5. Jeh, G., Widom, J.: SimRank: A measure of structural-context similarity. In: SIGKDD 2002, pp. 538–543 (2002)
6. Xi, W., Fox, E.A., Fan, W., Zhang, B., Chen, Z., Yan, J., Zhuang, D.: SimFusion: measuring similarity using unified relationship matrix. In: SIGIR 2005, pp. 130–137 (2005)
7. Yin, X., Han, J., Yu, P.S.: Linkclus: Efficient clustering via heterogeneous semantic links. In: VLDB 2006, pp. 427–438 (2006)
8. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, Chichester (1990)
9. Lovasz, L.: Random walks on graphs: a survey. In: Combinatorics, Paul Erdos is Eighty, vol. 2, pp. 1–46, Keszthely, Hungary (1993)
10. Kallenberg, O.: Foundations of Modern Probability. Springer, New York (1997)
11. Fogaras, D., Racz, B.: Scaling Link-Based Similarity Search. In: WWW 2005, pp. 641–650 (2005)
12. Getoor, L., Diehl, C.P.: Link mining: A survey. In: SIGKDD 2005 Explorations, vol. 7(2), pp. 3–12.
13. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5), 513–523 (1988)

14. Hammouda, K.M., Kamel, M.S.: Phrase-based Document Similarity Based on an Index Graph Model. In: ICDM 2002, pp. 203–210 (2002)
15. Aslam, J.A., Frost, M.: An Information-theoretic Measure for Document Similarity. In: SIGIR 2003, pp. 449–450 (2003)
16. Calado, P., Cristo, M., Moura, E.S., Ziviani, N., Ribeiro-Neto, B.A., Goncalves, M.A.: Combining link-based and content-based methods for web document classification. In: CIKM 2003, pp. 394–401 (2003)
17. Jin, R., Dumais, S.: Probabilistic Combination of Content and Links. In: SIGIR 2001, pp. 402–403 (2001)
18. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. In: SIGIR 2007, pp. 487–494 (2007)
19. Porter, M.: An algorithm for suffix stripping. Program, vol. 14(3), pp. 130–137 (1980), http://www.tartarus.org/~martin/PorterStemmer
20. Salton, G.: The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall, Inc., Upper Saddle River (1971)
21. The stop-words list, http://members.unine.ch/jacques.savoy/clef/englishST.txt
22. ACM Computing Classification System, http://portal.acm.org/ccs.cfm