*Technical Report of Knowledge Grid Research Center*, KGRC-2007-08, Sept, 2007.
www.knowledgegrid.net/TR.

# Automatically Discovering Semantic Links among Documents and Applications*

Hai Zhuge[1] and Junsheng Zhang[1,2]

*China Knowledge Grid Research Group, Key Lab of Intelligent Information Processing*

*Institute of Computing Technology, Chinese Academy of Sciences*[1]

*Graduate School of Chinese Academy of Sciences*[2]

*Beijing 100080, P. R. China*

**ABSTRACT**

Automatically discovering semantic links among documents is the basis of developing advanced applications on large-scale documentary resources. This paper proposes an approach to automatically discover semantic links in a given document set. It has the following advantages: (1) It does not rely on any predefined ontology. (2) The semantic link networks and relevant rules automatically evolve. (3) It can adapt to the update of the adopted techniques. Experiments on document sets of different types (scientific papers and Web pages on Dunhuang culture) and different scales show the proposed approach feasible. The approach can be used to automatically construct semantic overlays on large document sets to support advanced applications like various relation queries on documents.

**Keywords**: Semantic Link, Discovery, Semantic Web, Automation

## 1. INTRODUCTION

### 1.1 Motivation

Rethinking the success of the World Wide Web indicates the way to develop the Semantic Web by inheriting the features of the Web — the simple hyperlink mechanism and the easy utility mode.

The Semantic Link Network (SLN) model extends the hyperlink Web by attaching semantics to hyperlinks. A typical SLN consists of semantic nodes, semantic links between nodes, and semantic linking rules. A semantic node can be any type of resource, abstract concept or a SLN. Potential semantic links can be derived from an existing SLN according to a set of semantic linking rules. Adding a semantic link to the network could derive new semantic links. The major advantages of the SLN are its simplicity, the ability of relational reasoning and the nature of semantic self-organization: *any node can link to any semantically relevant node*.

* Supported by National Basic Research Program of China (2003CB317001).
Corresponding author: Hai Zhuge's email; zhuge@ict.ac.cn

The early SLN model was proposed in [16, 17]. A SLN builder was developed to help users build the SLN, and the SLN browser was developed to enable users to see several steps ahead when browsing the semantically linked document network [15]. An autonomous SLN model integrating logic reasoning, inductive reasoning and analogical reasoning was suggested in [14]. SLN was also used to improve the routing efficiency of peer-to-peer content network [18].

The SLN aims at an autonomous Semantic Web. An easy construction approach can promote the adoptability of the SLN and facilitate its application. This paper focuses on the approach to automatically discover semantic links among documentary resources.

### 1.2. Related Work

The semantic association between elements of RDF graphs was discussed for using alternative ways of specifying the context using ontology [1]. The approach supports capturing more precise user interests and better qualified results in relevance ranking. The approach for searching semantic link path and the ranking approach in a RDF graph was proposed in [2]. These works are based on the existing ontology in form of RDF.

A technique was proposed for automatically evaluating strategies of using Web hierarchies to replace user feedback [6]. The document similarity is calculated by using text, anchor–text and hyperlinks. An algorithm based on indexing and optimization strategies that solves document pair similarity in high dimensional spaces without relying on approximation methods or extensive parameter tuning was proposed [4].

An approach that answers complex questions with random walk model was proposed for answering complex questions relying on question decomposition and gives answers by a multi-document summarization system [5]. Graph model was used to expand the query for answering relation query on entities [8]. The method uses connecting terms to reconstruct the query, and finds more relations. Relation query on the Web was discussed for finding the document pairs that contain the probable semantic links between two entities and ranking them according to the probability [7].

An approach to the automatic categorization of documents was proposed for exploiting contextual information extracted from an analysis of the HTML structure of Web documents as well as the topology of the Web [3]. Human-created metadata for Web directories was used to estimate semantic similarity and semantic maps to visualize relationships between content and link cues and what these cues suggest about page meaning [9].

## 2. THE DOCUMENTARY SEMANTIC LINK NETWORKS

A documentary SLN consists of the following components:

(1) Document set $D = \{d_1, d_2, ..., d_n\}$ $(n \geq 1)$.

(2) Document cluster set $C = \{c_1, c_2, ..., c_n\}$ $(n \geq 1)$, where $c_i = \{d_1, d_2, ..., d_m\}$ $(1 \leq i \leq n, m \geq 1)$ is a document cluster.

(3) Semantic link set $SL = \{(s_1, \omega_1), (s_2, \omega_2), ..., (s_p, \omega_p)\}$ where $s_i (1 \leq i \leq p)$ is a semantic link between documents, between document and cluster, or between clusters, and $\omega_i \in [0, 1]$ is the probability of $s_i$. Semantic links can be assigned by users, inferred by inference rules, or derived by semantic linking rules.

(4) Keyword set of document set $TD = \{t_1, t_2, ..., t_q\}$, where $q \geq 1$, and $t_i (1 \leq i \leq q)$ is a keyword found by TF-IDF approach. Each document or document cluster has its corresponding keyword set.

(5) Rule set $RULES = \{LR, IR, CR\}$,

- *LR* is a set of semantic linking rules, and each linking rule takes the following form $(\alpha, \omega_1) \times (\beta, \omega_2) \rightarrow (\gamma, w_1 w_2)$, where $\omega_1$ and $\omega_2$ are the probability values of semantic links $\alpha$ and $\beta$ respectively. The SLN model is equipped with a basic set of linking rules. Users can define more linking rules for domain-specific semantic links.
- Inference rule set *IR* consists of rules of two types: statistical inference rules and rules for judging the semantic links according to metadata on documents and citations among documents.
- Classification rule set *CR* consists of rules for classifying documents according to the probable relation between keywords, documents and document clusters.

Table 1 shows the difference between the typical SLN [14, 15] and the documentary SLN. The document classification rules are for classifying documents and discovering semantic links. The semantic link inference rules are for determining the semantic links.

**Table 1.** Typical SLN and Documentary SLN.

| Components | Typical SLN | Documentary SLN | Explanation |
|---|---|---|---|
| semantic nodes | $A$, $B$, $C$ | $A$, $B$, $C$ | A node can be any type of resource, a concepts or a SLN |
| semantic links | $A \xrightarrow{\alpha} B$, $B \xrightarrow{\beta} C$ | $A \xrightarrow{(\alpha,\omega_1)} B$, $B \xrightarrow{(\beta,\omega_2)} C$ | $\omega_1$ and $\omega_2$ are probabilities of semantic links $\alpha$ and $\beta$ |
| linking rules | $\alpha \times \beta \rightarrow \gamma$ | $(\alpha,\omega_1)\times(\beta,\omega_2)\rightarrow(\gamma,\omega_1\omega_2)$ | $\alpha$, $\beta$ and $\gamma$ are semantic links, $\omega_1$ and $\omega_2$ are probabilities of semantic links $\alpha$ and $\beta$ respectively |
| classification rules | | $\{p(c \mid t_i) \in [0, 1]\}$ | The probability of a keyword $t_i$ in a cluster $c$ |
| inference rules | | $\{p(r_i, c_1, c_2) \in [0, 1]\}$ | The probability of the existence of a semantic link $r_i$ between clusters $c_1$ and $c_2$ |
| cluster and keywords | | $\{c_i(t_1, t_2, \ldots, t_n)\}$ | $c_i$ is a cluster, and $t_i$ is a keyword |

A documentary SLN consists of the following parts:

- *Cluster–document network* consists of document clusters and document entities. The *instanceOf* link exists between a document and a cluster. The *equal*, *similar*, and *subCluster/partOf* links exist between clusters and between documents.
- *Citation network* mainly consists of *refer* link, which can be refined as *cocite*, *cocited, sequential, improve* and *similar* links.
- *Metadata network* consists of relations between document attributes such as *sequential* and *equal* links. The equal link can be specialized such relations as *sameAuthor*, *sameJournal* and *sameConference*. The document metadata and citation relations can be obtained from digital libraries of related document sources.
- *Keyword–document network* consists of the *occur* link between keywords and documents. The probability of the *occur* link is the weight of the keyword in the document. The *co-occur* link exists between a pair of keywords if they appeared in the same document.

- *Document classification rule* classifies new documents and inserts them into the documentary SLN. This type of rules can be obtained from the cluster networks and from keyword–document networks by Bayes formula.

- *Semantic link inference rule* infers semantic links between documents according to existing classifications on documents. This type of rules is acquired by statistical method from existing SLN.

- *Semantic linking rule* reflects the relationship between semantic links. A general linking rule can be specialized in different domains. For example, the *equal* link can be explained as the *sameTopic* and the *partOf* link can be explained as the *subSite* link. Given a set of domain-specific rules and the semantic links, the semantic link network supports relational reasoning.

Figure 1 shows the documentary SLN model and its construction mechanism. It does not require any background knowledge nor ontology. When new documents are inserted into the documentary SLN, semantic links between the new document and the existing nodes are inferred by the inference rules and the linking rules. Then, the statistical inference rules of semantic links are recalculated, the keywords of document clusters change, and the document classification rules are recalculated.

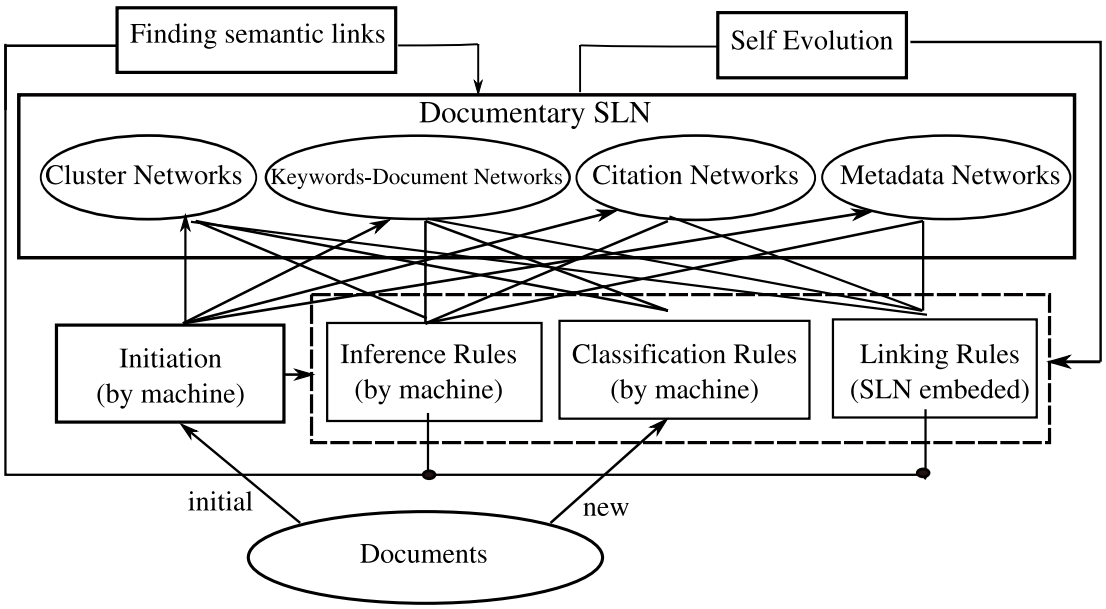



Figure 1: The documentary SLN mechanism.

## 3. INITIATION

### 3.1 Step 1: Document Analysis

This step is to determine the document source, input documents, carry out text analysis, and build keyword-document networks.

Each document cluster has a representative keyword set, which can be used to classify the new documents. Documents, classifications, keywords and attributes make up of a coarse documentary SLN. Figure 2 is a documentary SLN segment that contains two documents $d_1$ and $d_2$ and relevant keywords.

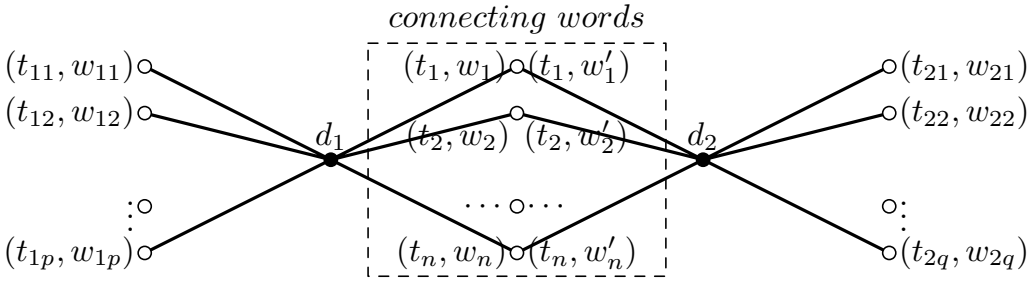



**Figure 2.** A segment of the keyword–document network. The *occur* link exists between keywords and documents. The *co-occur* link also exists between keywords of the same document.

### 3.2 Step 2: Document Pair Similarity

The document vector and keyword set can both represent a document. Document vectors are used to calculate the document pair similarity by the *vector cosine* formula, while the keyword sets are used to calculate the document similarity with *Jaccard Coefficient* formula. According to the comparison of different similarity measures [12], we adopt the *cosine* and *Jaccard* similarities and compare different similarities and their combinations.

According to TF-IDF [10, 11], a documents can be represented as the vectors $d = \{(t_1, \omega_1), (t_2, \omega_2), \ldots, (t_m, \omega_m)\}$, where $t_i (1 \le i \le m)$ is a keyword, and $\omega_i (1 \le i \le m)$ is its weight. The similarity between documents $d_1$ and $d_2$ can be measured as follows.

- The similarity *sim_vec* can be calculated by

  $$sim\_vec = \frac{\sum_{i=1}^{n} \omega_i \omega_i^{'}}{\sqrt{\sum_{i=1}^{n} (\omega_i)^2 \sum_{i=1}^{n} (\omega_1^{'})^2}}$$, where $n$ is the number of common keywords between $d_1$ and $d_2$.

- The similarity *sim_content* can be calculated according to the keyword sets of documents. Suppose keyword sets of $d_1$ and $d_2$ are $S(d_1)$ and $S(d_2)$ respectively, $S(d_1) = (t_1, t_2, \ldots, t_m)$, and $S(d_2) = (t_1, t_2, \ldots, t_n)$, then

  $$sim\_content = \frac{|S(d_1) \bigcap S(d_2)|}{|S(d_1) \bigcup S(d_2)|}$$, where $m \geq 0$ and $n \geq 0$.

- Combine the similarity *sim_vec* and *sim_content*, the similarity between $d_1$ and $d_2$ can be calculated by

  $$sim(d_1, d_2) = \alpha \bullet sim\_vec + (1-\alpha) \bullet sim\_content$$, where $\alpha \geq 0$.

The document pair similarity is used to cluster documents.

### 3.3 Step 3: Document Clustering

To control the clustering results, one way is to set the similarity threshold, the other is to sort the edges by similarity and control the edge number. Three clustering approaches are as follows.

- *Comparing every document pair.* Algorithm 1 clusters documents by setting the similarity threshold. The similarity between documents in the same cluster is higher than the threshold. The changes of the similarity threshold influence the clustering results.
- *Adding edge*. Construct a graph made up of all isolated nodes of documents. Then, sort the document pair similarities in the descending order and insert the edges from top to bottom. With adding edges to the network, the number of connected components (clusters) reduces step by step. The change of the number of edges influences the clustering results. The clustering condition is looser than Algorithm 1.
- *Deleting edge*. Documents can be clustered by deleting edges. Firstly, the clique graph is constructed by all the document pair similarities. Then, sort the edges according to the similarities in the ascending order. Finally, deleting the edges from top to bottom. With deleting edges, the number of connected components increases. The changes of the number of edges influence the clustering results.

  For a large document set, Algorithm 1 is preferred to get a more precise clustering result.

**Algorithm 1**: Document Cluster Algorithm
**Input** : document pair and their similarity ($d_1$, $d_2$, $sim(d_1, d_2)$) , ($d_1$, $d_3$, $sim(d_1, d_3)$), …, ($d_1$, $d_n$, $sim(d_1, d_n)$), ($d_2$, $d_3$, $sim(d_2, d_3)$), …, ($d_{n-1}$, $d_n$, $sim(d_{n-1}, d_n)$)
**Output**: Document Clusters

Let *nodes* = {$d_1$, $d_2$,..., $d_n$};
Let *clusters* = ∅;
**while** *nodes* ≠∅ **do**
  Get *node* from *nodes*;
  clustered = false;
  *nodes* = *nodes*–{*node*};
  **if** (*clusters* = = ∅)
    **then**
      Add *cluster* = {*node*};
      *clusters* = *clusters* + {*cluser*};
    **else**
      *checkcluster* = *clusters*;
      **while** (*not clustered*) *and* ( *checkcluster* ≠ ∅)) **do**
        Get a cluster *c* from *checkcluster*;
        *checkcluster* = *checkcluster* –{*c*};
        **if** ( *similarity between any page p in c and node sim* ≥ *threshold* )
          **then**
            *c* = *c* + {*node*};
            clustered = true;
      **if** (*not clustered*)
        **then**
          New a cluster $c_1$ = {*node*};
          clustered = true;
          *clusters* = *clusters* + {$c_1$};
return *clusters*;

### 3.4 Step 4: Creating Classification Rules

After the keyword–document networks and the document–cluster networks are built, the association rules between keywords, documents and clusters can be calculated by Bayes formula. The document classification rules can be used to infer document classifications with keywords in documents.

Documents are clustered according to keywords and their weights. Figure 3 shows the relations between keywords, documents and classes. An arrow between document $d_1$ and class $c_1$ means that $d_1$ belongs to $c_1$, and $p(c_1 \mid d_1)$ means the probability of $d_1$ belonging to $c_1$. An arrow between keyword $t_1$ and document $d_1$ means that $t_1$ occurs in $d_1$, and $p(d_1 \mid t_1)$ means the probability that $t_1$ occurs in $d_1$.

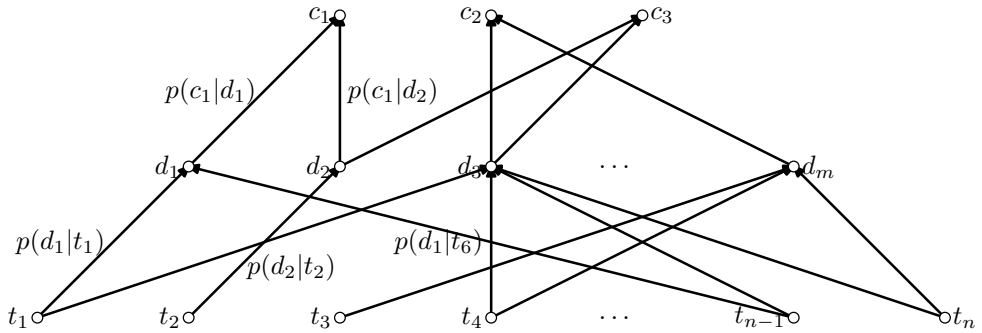



**Figure 3.** Classifying documents according to the probable relations between keywords, documents and clusters.

After the initial documents are clustered, each document cluster is assigned by an automatically generated class name. Each cluster has the representative keyword set chosen from the intersection of document keyword sets or chosen according to the weights of keywords in the documents of the cluster.

The inference rules between keywords and classes are acquired by statistic method. A keyword $t$ occurs in documents $d_1$, $d_2$, ..., and $d_n$ of cluster $c_1$, then the association between keyword $t$ and cluster $c_1$ can be calculated by

$$P(c_1 \mid t) = \sum_{i=1}^{n} P(c_1 \mid d_i) P(d_i \mid t),$$

where $P(c_1 \mid d_i)$ is the probability that document $d_i$ belongs to class $c_1$, and $P(d_i \mid t)$ is the probability that word $t$ occurs in document $d_i$. $P(c_1 \mid d_i)$ is calculated by the cluster algorithms while $P(d_i \mid t)$ are calculated by Bayes formula as follows:

$$P(d_i \mid t) = \frac{P(t \mid d_i)}{\sum_{t \in d} P(t \mid d)},$$

where $P(t \mid d_i)$ means the probability that word $t$ occurs in document $d_i$.

### 3.5 Step 5: Building Cluster Networks

The document cluster vector is the mean vector of all the document vectors in the cluster. The elements of the keyword set are chosen from the intersection of the document keyword sets in the cluster. With the document cluster representative vectors, the similarity *sim_vec* between cluster pairs can be calculated with the *vector cosine* formula. With the cluster keyword sets, the similarity *sim_content* between cluster pairs can be calculated by *Jaccard Coefficient*

formula. As each cluster corresponds to a document classification, the similarity between document classifications can be calculated with *sim_vec*, *sim_content*, or their combination, just like the similarity between documents.

The document clustering process is iterative, that is, the document clusters can be clustered iteratively until all the documents are in the same cluster or the iteration times have come to the maximum iteration time predefined. With the iterative clustering, document clusters formulate a hierarchical SLN. Each pair of clusters at the same level has semantic associations, and the association degree is their similarity values. The *subCluster* link may exist between clusters of different levels.

Figure 4 shows a SLN segment of document clusters, where $d_x$ represents document, $c_{1x}$ and $c_{2x}$ represent the clusters at level 1 and level 2 respectively, and $c_h$ represents the cluster networks that are iteratively clustered $h$ times into one cluster. The *subCluster* links exist between clusters at different levels, while the *ins* (*instanceOf*) links are between documents and clusters. Clusters at the same level have semantic associations, and the association values are the similarity values between clusters such as $sim(c_{1x},c_{1x})$ and $sim(c_{2x},c_{2x})$.

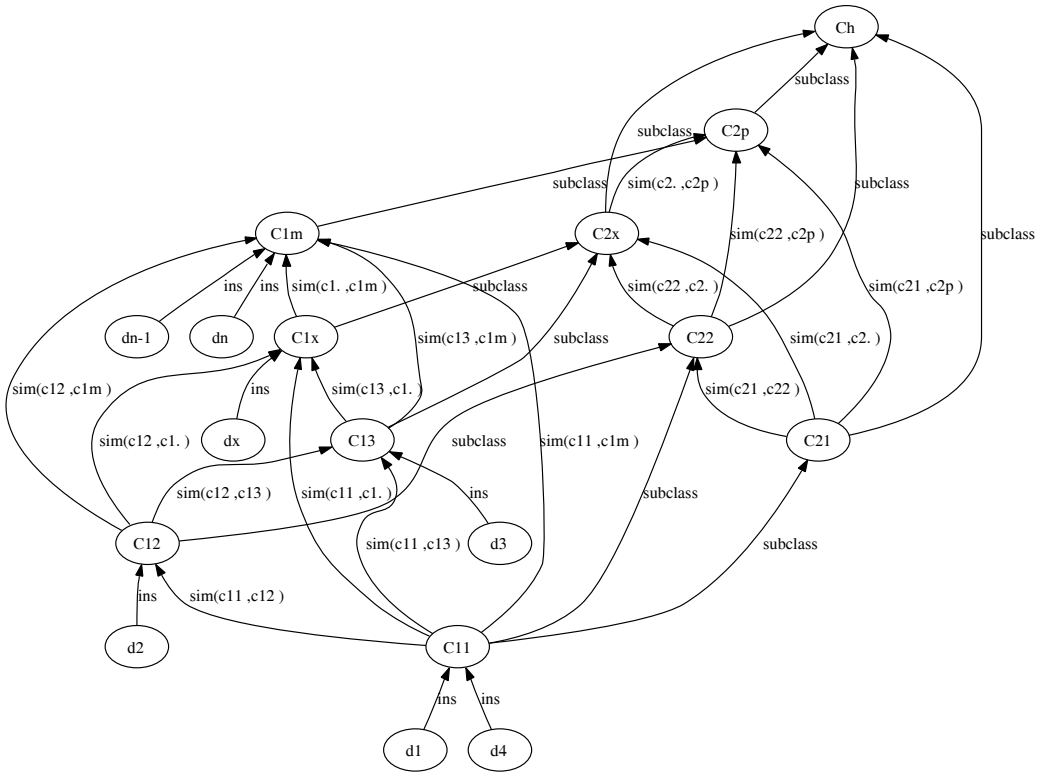



**Figure 4**. The hierarchical document cluster network consists of the *instanceOf* link between documents and clusters, the *association* link between clusters of the same level, and the *subCluster* link between clusters of different levels.

## 4. DISCOVERING SEMANTIC LINKS

After documents are clustered, the semantic links between documents, semantic associations between clusters, and semantic links between documents and clusters can be discovered and inserted into the documentary SLN.

### 4.1 Discover Semantic Links from Contents

If $T_1$ and $T_2$ are keyword sets of documents $d_1$ and $d_2$ in the same cluster, the semantic links between them such as *similar*, *partOf* and *equal* can be discovered according to the rules listed in Table 2.

**Table 2.** Discovering semantic links according to keyword sets.

| Semantic link | Characteristics |
|---|---|
| *irrelevant* | $\|T_1 \cap T_2\| = 0$ |
| *similar* | $0 < \|T_1 \cap T_2\| < min(\|T_1\|, \|T_2\|)$, the similarity can be defined as $\|T_1 \cap T_2\| / \|T_1 \cup T_2\|$ |
| *partOf* | $\|T_1 \cap T_2\| = min(\|T_1\|, \|T_2\|) < max(\|T_1\|, \|T_2\|)$ |
| *equal* | $\|T_1\| = \|T_2\|$ |

Documents sharing keywords can form a *cluster*. The union of keyword sets of documents can represent the cluster. The mean of document vectors can be as the *cluster vector*. Let $T(c_1)$ and $T(c_2)$ be the keyword sets of cluster $c_1$ and $c_2$ respectively. The following semantic links between clusters can be discovered:

- $c_1$ is the *subCluster* of $c_2$ (denoted as $c_1$ —*subCluster*→ $c_2$) if $T(c_1) \subset T(c_2)$.
- $c_1$ is equivalent to $c_2$ (denoted as $c_1$ —*equal*→ $c_2$) if $T(c_1) = T(c_2)$.
- $c_1$ is similar to $c_2$ (denoted as $c_1$—*similar*→ $c_2$) if $T(c_1) \cap T(c_2) = \varnothing$, $T(c_1) \cap T(c_2) \neq T_1$ and $T(c_1) \cap T(c_2) \neq T(c_2)$.

A document $d$ can be regarded as an instance of $c$ (denoted as $d$—*instanceOf*→$c$) if a document $d$ belongs to cluster $c$. Table 3 lists some linking rules of semantic links discovered from document content. These rules enable the semantic link network to carry out relation reasoning.

**Table 3.** Some semantic linking rules for documentary semantic link network.

| Semantic linking rules | Characteristics | Explanation |
|---|---|---|
| *subCluster*×*subCluster* → *subCluster* | $T(c_1) \subset T(c_2)$, $T(c_2) \subset T(c_3) \Rightarrow T(c_1) \subset T(c_3)$ | Relationship between clusters |
| *partOf*×*irrelevant*→ *irrelevant* | $T(d_1) \subset T(d_2)$, $T(d_2) \cap T(d_3) = \varnothing \Rightarrow T(d_1) \cap T(d_3) = \varnothing$ | Relationship between documents |
| *partOf*×*partOf*→*partOf* | $T(d_1) \subset T(d_2)$, $T(d_2) \subset T(d_3) \Rightarrow T(d_1) \subset T(d_3)$ | Relationship between documents |
| *instanceOf*×*subCluster* →*instanceOf* | $d_1 \in c_1$, $c_1 \subset c_2 \Rightarrow d_1 \in c_2$ | Relationship between document and cluster |
| *partOf*×*instanceOf*→ *instanceOf* | $T(d_1) \subset T(d_2)$, $d_2 \in c \Rightarrow d_1 \in c$ | Relationship between document and cluster |

If there are semantic links between two documents, the two documents should share some *connecting words* that imply the semantic links. The larger are the number and weights of *connecting words*, the higher the probability that semantic links exist between them. The initial inference rules depend on the initial document set. After initiation, the semantic links between documents can be discovered automatically without human intervene.

### 4.2 Discover Semantic Links from Metadata and Citations

Document metadata includes *authors*, *editors*, *publish time*, *journal/conference*, *paper types* like regular/short paper, *page number*, *author address* etc. The metadata of scientific papers in computer science is often stored in XML or BibTeX files which can be obtained from DBLP (http://dblp.uni-trier.de/), IEEE digital library (http://www.ieeexplore.ieee.org/) and ACM portal (http://portal.acm.org/). If the semantic linking rules related to the document attributes are defined, the corresponding semantic links are easy to find. For example, comparing authors can find the *sameAuthor* link, and comparing the *publishTime* can find *sequential* link. Table 4 shows the semantic links related to the metadata.

Figure 5 is a SLN segment which contains some document metadata, including *author*, *editor*, *type*, *journal/conference* and *publishTime*. Besides, there are some optional information such as *page number*, *project*, *institution* and *language*. From the document metadata, some semantic links including *sameAuthor*, *sameEditor*, *sameJournal*, *sameConf*, *sameYear*, *sameProject*, *sameAffiliation* and *sameLanguage* can be found.

**Table 4.** Metadata and related semantic links.

| Field name | Semantic links | Explanations |
|---|---|---|
| title, abstract, keywords | *similar*, *sameTopic*, *partOf*, *sameArea* | The *sameTopic* link and the *sameArea* link are the specialization of the *equal* link. The *partOf* link implies the *similar* link. |
| author | *sameAuthor*, *coauthor*, *reviewerOf*, *recommenderOf*, *commenterOf* | The *sameAuthor* is the specialization of the *equal* link. The *reviewerOf* link exists in scientific paper publication activities but it is usually anonymous. In Web 2.0, the *commenterOf* link is open and widely used. |
| institution | *sameInstitution*, *subInstitution* | The *sameInstitution* link is the specialization of the *equal* link. The *subInstitution* link is the specialization of the *subCluster* link. It can be got by analyzing the composition of name. |
| journal/conf. | *sameJournal*, *sameConf*, *subCluster* | The *sameJournal* and the *sameConf* link are the specialization of the *equal* link. The *subCluster* link establishes the ties between journals/confs to their ranks (impacts). |
| date | *sequential*, *sameDate* | The *sequential* link implies the *earlierThan* or *laterThan* link. The *sameDate* link is the specialization of the *equal* link. |
| length | *longerThan*, *shortThan* | Related reasoning: If paper A is a regular paper, and paper B is *longerThan* A, then B should be a regular paper. If paper A is a short/concise paper, and paper B is shorter than A, then B should be a short/concise paper. |
| language | *sameLanguage* | The *sameLanguage* link is the specialization of the *equal* link. |
| project/grant no. | *sameProject*, *sameTeam* | The *sameProject* link is the specialization of the *equal* link. The authors acknowledge the same project number work for the same project/grant. |

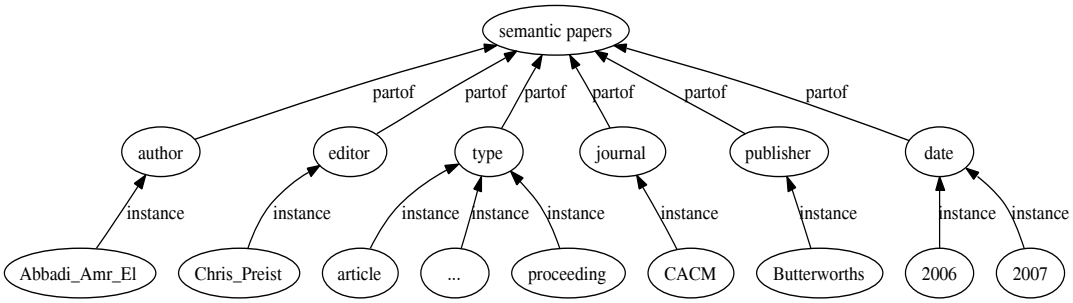



**Figure 5.** SLN segment from document metadata.

Citation networks can be constructed by the citation links from *Citeseer* (http://citeseer.ist.psu.edu/), *Google Scholar* (http://scholar.google.com/) and *Science Citation Index* (SCI). Citation relation leads to more semantic link types such as *sequential*, *sameTopic*, *sameMethod*, *cocite*, *cocited*, and *sameTopic* as shown in Table 5.

Semantic links such as *refer*, *cocite* and *cocited* can be found in the citation networks, while the *sequential* semantic link can be derived from *refer* links. Semantic links such as *sameTopic*, *sameModel* depend on document analysis.

**Table 5.** Semantic links related to the citation link.

| Semantic links | Denotation | Explanation |
|---|---|---|
| *reference* | $A$ —*refer*→ $B$ | $A$ refers to $B$ |
| *sequential* | $A$ —*seq*→ $B$ | $A$ is after $B$ |
| *cocite* | $(A,B)$ —*cocite*→ $C$ | Both $A$ and $B$ refer to $C$ |
| *cocited* | $(A,B)$ —*cocited*→ $C$ | Both $A$ and $B$ are referred by $C$ |
| *sameTopic* | $A$ —*sameTopic*→$B$ | $A$ and $B$ share the same topic |

### 4.3 Building Semantic Link Inference Rules

Comparing with frequent changes of nodes and semantic links in the documentary SLN, the semantic linking rules are relatively stand-alone and static.

Each document may belong to several clusters probably at the same time. Let $r$ be a specific semantic link, *src* be the source cluster, *tgt* be the target cluster and $p$ be the probability of $r$, storing semantic link data ($r$, *src*, *tgt*, $p$) in a relational table and taking ($r$, *src*, *tgt*) as the primary key, the probability of $r$ is calculated as follows:

$$P(r, src, tgt) = \frac{\sum p(r, src, tgt)}{\sum p(R, src, tgt)} \times P(src, tgt) \quad (1)$$

where

- $p(r, src, tgt)$: the probability of a semantic link $r$ between the source cluster $src$ and the target cluster $tgt$.
- $\Sigma p(r, src, tgt)$: the sum of probabilities of semantic links between cluster $src$ and cluster $tgt$;
- $R$: any semantic link between cluster $src$ and cluster $tgt$;
- $P(src, tgt)$: the probability of the existence of semantic links from cluster $src$ to cluster $tgt$ is calculated by

$$P(src, tgt) = \frac{|\{r \mid \text{semantic link } r \text{ is from cluster } src \text{ to cluster } tgt\}|}{|\{R \mid \text{semantic link } R \in \text{the SLN}\}|}.$$

Eq. (1) is used as the semantic link inference rule set. If two documents $d_1$ and $d_2$ are given, their classifications can be found by using the classification algorithm, and the probability of semantic link $r$ between $d_1$ and $d_2$ can be inferred if $d_1$ belongs to class $A$ and $d_2$ belongs to class $B$.

### 4.4 Inferring and Reasoning

Inference rules evolve with the changes of SLN. Given two documents $d_1$ and $d_2$, the semantic links between them are inferred as follows.

1. According to the keyword list captured from the initial document set by TF-IDF approach, we can get the document vectors of $d_1$ and $d_2$, and denote the keyword sets as $T_1$ and $T_2$.

2. Classify documents $d_1$ and $d_2$ by using

   - the document vector. One way uses the document vectors with $k$-NN algorithms. The other way calculates the similarity between the document vectors and the cluster vectors.

   - the document keyword set to compare the similarity of the document keyword sets or the cluster keyword sets by using *Jaccard Coefficient* formula. The most similar clusters are chosen as the document classification.

   - the keyword–cluster association rules among documents, keywords and clusters.

3. If $d_1$ and $d_2$ are in the same cluster, find the semantic links by using document keyword sets according to the rules in Table 3. Semantic links such as *irrelevant*, *similar*, *partOf* or *equal* can be discovered.

4. Infer the semantic links between $d_1$ and $d_2$ with semantic link inference rules built in Sec. 4.3.

After the documentary SLN is constructed, more semantic links can be reasoned according to the linking rules. If the semantic links exist between two nodes in the SLN, there would be one or more semantic link paths between them. The probability values of the derived semantic links can be calculated by the production of all the probability values of the semantic links in the path, in this way, the probability of the derived semantic links reduces with the increase of the semantic link path length.

If there are no corresponding linking rules for two neighboring semantic links, the reasoning result of the semantic link path can be regarded as *null*, which means that the semantic links between the source and the target are *unknown*.

## 5. EVOLUTION

A documentary SLN evolves with the changes of nodes and semantic links. The insertion of documents may lead to the occurrence of new clusters. The vectors and keywords of new documents activate the changes of cluster vectors and keyword sets.

Figure 6 shows the evolution process of the documentary SLN. When a new document comes, the document vector and the keyword set are calculated by the TF-IDF approach. During the initiation, the clustering approaches are preferred. With the increase of documents, document clustering is time consuming [13]. Since the document cluster vectors and keyword sets are obtained, new documents can be classified by using the document classification algorithms. With the *k*-NN algorithm, the most probable document classification could be found. Because a document may belong to several clusters, the keyword–cluster association rules can be used to infer the document classifications. After the document classification is finished, new documents can be inserted into the document cluster networks. The new documents will change the cluster networks and the keyword networks, and the document–cluster–keyword networks will influence the document classification rules according to Bayes formula.

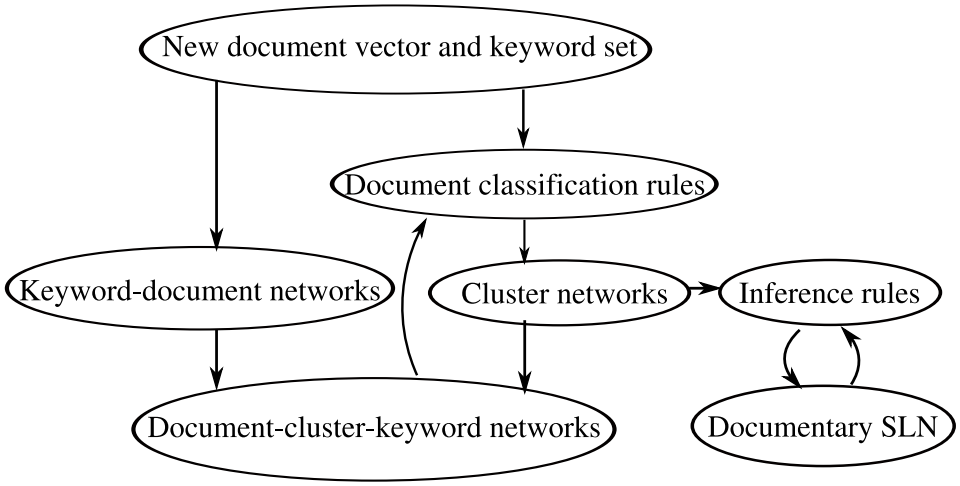



**Figure 6.** Documentary SLN evolution.

### 5.1 Evolution of Cluster Networks

The evolution of cluster networks carries out with

1. new document insertion.
   - When a new document is inserted, semantic links between new document and its clusters, and the semantic links between new document and other documents are discovered.
   - The cluster vector evolves with the changes of the document vectors. The new documents will affect the weights of keywords, which cause the change of the keyword set.
2. new document clusters occurrence.
   - Semantic association degrees between the new cluster and the existing clusters are calculated.
   - New clusters will be clustered into the higher level clusters, and the depth of the cluster networks may increase.

### 5.2 Evolution of Inference Rules

As Eq. (1) shows, an inference rule is influenced by the following factors:

- The source cluster or the target cluster of the semantic links changes. New documents can lead to the changes of the clusters or the occurrence of more clusters.

- The new semantic link types occur. When new documents are inserted, semantic link types may be increased. The change of the number of semantic links leads to the change of inference rules.

- The classification rules change. The association rules between keywords, documents and clusters will evolve with the changes of the cluster vectors and keyword sets. The classification rules change with the probability values of the semantic links between documents and clusters.

Because the inference rules will evolve with the changes of the documentary SLN, the semantic link types and the probability of semantic links between documents are relevant to the document insertion order. Even the same document is inserted into the documentary SLN at different times, the semantic links and the probability values will be different. When duplicate documents are inserted into the documentary SLN, inconsistency will occur.

Inconsistent inference results may occur but they reflect the uncertainty of the documentary SLN. Different inference results are caused by different initial document sets. Documents inserted into the documentary SLN will act as the initial documents which will influence the classification and semantic link inference of the new coming documents. With the evolution of the documentary SLN, the semantic link types will become more plentiful, and the probability values of the semantic links will be more precise.

## 6. EXPERIMENT AND APPLICATION IN CHINA

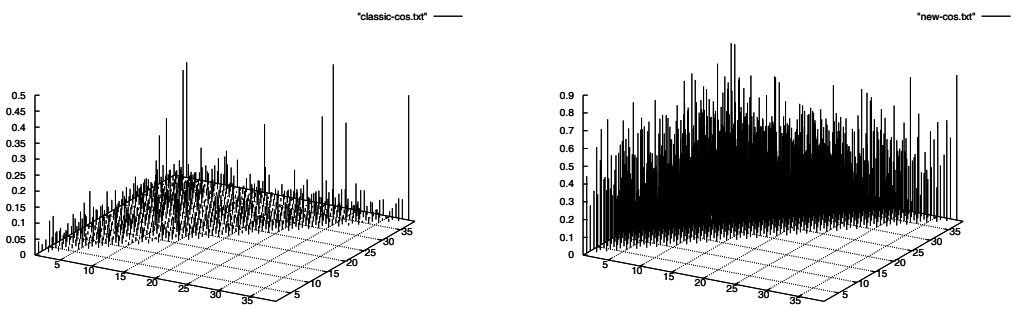
### 6.1 Discover SLN among Scientific Papers

We collect 39 papers including text, metadata and citations in the Semantic Web area for experiment (as shown in Table 15 of Appendix).

When computing document similarity, the classical *cosine vector similarity* formula considers all the keywords although document pairs do not contain all the keywords. It leads to the parse and low values of document similarity, and most of them are 0.

We calculate document pair similarity only considering their common keywords and the corresponding weights. Experiment results show that the document similarity values distribute evenly than classical calculation method. Especially, if two documents have no common keywords, then their similarity is 0.

Figure 7 shows the difference of document pair similarity calculation by classic cosine vector approach and our modified classic cosine vector approach. Since the similarity values of document pairs are symmetric, only half of the result is plotted.

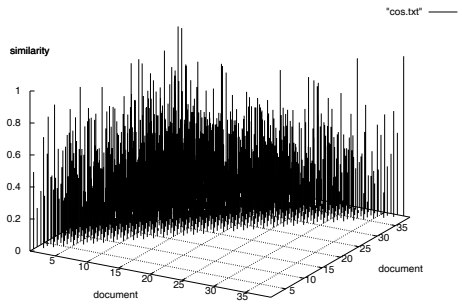



(*a*) *classic-cos* (*b*) *new-cos*

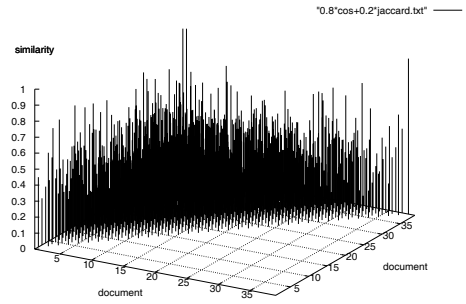
**Figure 7.** Document pair similarity calculated by classic and new cosine vector.

*Jaccard similarity* only considers the keywords, while the *cosine* method considers keywords and their weights. We combine the *cosine* similarity and *Jaccard Coefficient* to calculate document pair similarity. Figure 8 shows parameters of different weights for *cosine* similarity and *Jaccard Coefficient* similarity. To combine the two kinds of similarity, the similarity intervals are mapped onto [0, 1]. We can see that the *cosine* similarity and *Jaccard* similarity are very similar to each other except for the difference in similarity intervals.

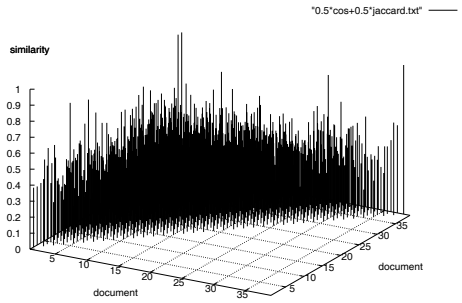
After the similarity values of document pairs are calculated, the document pairs are sorted in descending order according to the similarity values. The similarity values are within [0.0774412, 0.817713]. By controlling the number of top similar document pairs, the document clustering results from *cosine* similarity are listed in Table 6.



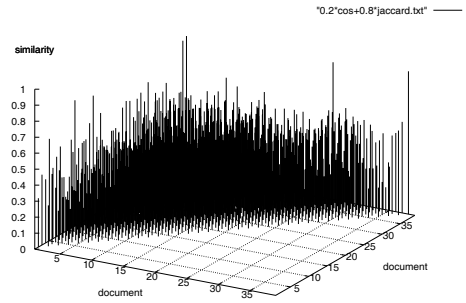

(*a*) *cos*



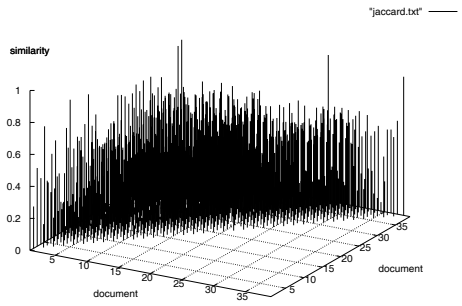

(*b*) 0.8*cos* + 0.2*jaccard*



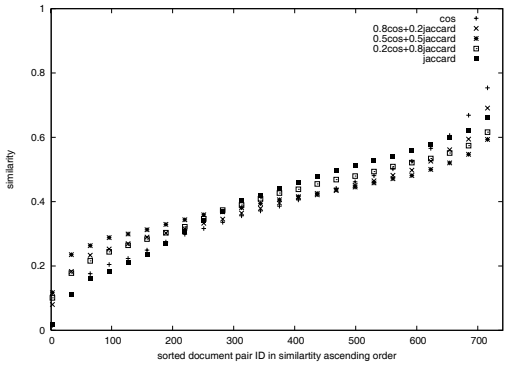

(*c*) 0.5*cos* + 0.5*jaccard*




(*d*) 0.2*cos* + 0.8*jaccard*




(*e*) *jaccard*




(*f*) *distribution comparing*

**Figure 8.** Document similarities with different calculation methods and the similarity distributions (cosine similarity and Jaccard similarity are mapped onto [0, 1]).

**Table 6.** Clustering with different edge numbers.

| Edge | Similarity | Cluster | Edge | Similarity | Cluster |
|---|---|---|---|---|---|
| 15 | 0.689902 | 25 | 30 | 0.623188 | 13 |
| 50 | 0.582651 | 7 | 100 | 0.508107 | 1 |

Figure 10 in Appendix shows different clustering results with top 15 similar edges according to different similarity calculation approaches. Table 16 in Appendix shows top 15 similar document pairs according to different similarity calculation methods. Take Figure 10(*c*) for example, the document clusters and keyword numbers are shown in Table 7.

**Table 7.** Document clusters and their union and intersection keyword set.

| Cluster | Documents | Union | Common | Total |
|---|---|---|---|---|
| $c_1$ | (d1,d12,17) | 2334 | 216 | 12363 |
| $c_2$ | (d3,d38,d39) | 2579 | 153 | 12363 |
| $c_3$ | (d5,d27) | 2157 | 390 | 12363 |
| $c_4$ | (d6,d23,d35) | 838 | 45 | 12363 |
| $c_5$ | (d8,d22) | 1979 | 351 | 12363 |
| $c_6$ | (d11,d36) | 2199 | 450 | 12363 |
| $c_7$ | (d15,d26) | 2261 | 468 | 12363 |
| $c_8$ | (d19,d29,d31,d34) | 2986 | 161 | 12363 |

With the document keyword intersection and union, we use the following two indicators for clustering evaluation:

- *Semantic abstract degree*. The more abstract document cluster owns more documents, and it is more close to the root of taxonomy tree. Semantic abstract degree *sem_abstract* is calculated by

$sem\_abstract(c) = \frac{\bigcap_{d_i \in c} T_i}{\bigcup_{d_i \in c} T_i}$, where $d_i$ is a document, $T_i$ is the keyword set of $d_i$, and $c$ is a document cluster.

- *Semantic cover degree*. It reflects the ratio between the document cluster and all document set. The larger is the *sem_cover*, the more semantics of document cluster has. Semantic cover degree *sem_cover* is calculated by

$sem\_cover(c) = \frac{\bigcup_{d_i \in c} T_i}{\bigcup_{d_i \in \{c_1, c_2, \ldots, c_n\}} T_i}$, where $d_i$ is a document, $T_i$ is keyword set of $d_i$, $c$ is a document cluster, and $\{c_1, c_2, \ldots, c_n\}$ are existing document clusters.

Figure 9 shows the semantic abstract degree and the semantic cover degree with data in Table 7.

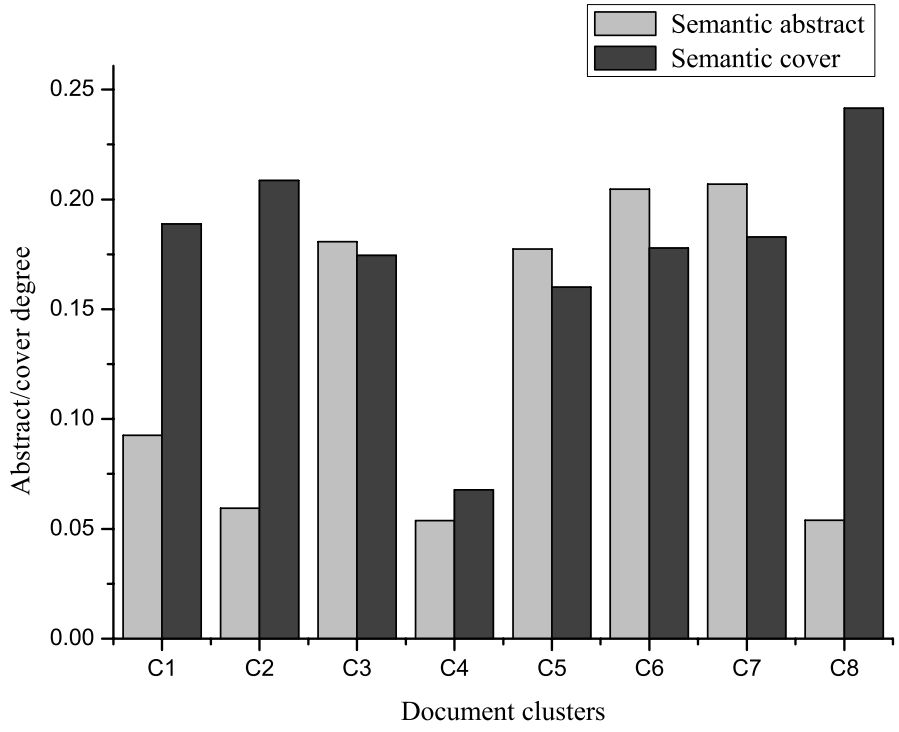



**Figure 9.** Document clustering evaluation.

Table 8 lists several clusters and the related papers in the 4-time interactive clustering process. We cluster papers by the *adding edge* approach, and the edge numbers for iterative clustering are 15, 30, 50 and 100 respectively. Each paper has *instanceOf* link to the corresponding cluster. At the initiation stage, the semantic link degrees are set as 1.

**Table 8.** Paper clusters (each paper belongs to a cluster via the *instanceOf* link), $d_i$ denotes a paper and $c_i$ denotes a document cluster.

| Cluster | Papers in the cluster |
|---|---|
| $c_{11}$ | $\{d_2, d_6, d_{13}, d_{20}, d_{23}, d_{29}, d_{31}\}$ |
| $c_{12}$ | $\{d_3, d_{38}, d_{39}\}$ |
| $c_{13}$ | $\{d_{10}, d_{21}, d_{22}, d_{25}, d_{28}, d_{33}, d_{35}\}$ |
| $c_{21}$ | $\{d_1, d_2, d_3, d_6, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{19}, d_{20}, d_{21}, d_{22}, d_{23}, d_{25}, d_{28}, d_{29}, d_{31}, d_{32}, d_{33}, d_{34}, d_{35}, d_{36}, d_{37}, d_{38}, d_{39}\}$ |
| $c_{22}$ | $\{d_5, d_{27}\}$ |
| $c_{31}$ | $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{17}, d_{19}, d_{20}, d_{21}, d_{22}, d_{23}, d_{25}, d_{26}, d_{27}, d_{28}, d_{29}, d_{31}, d_{32}, d_{33}, d_{34}, d_{35}, d_{36}, d_{37}, d_{38}, d_{39}\}$ |
| $c_{41}$ | $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{14}, d_{15}, d_{16}, d_{17}, d_{18}, d_{19}, d_{20}, d_{21}, d_{22}, d_{23}, d_{243}, d_{25}, d_{26}, d_{27}, d_{28}, d_{29}, d_{30}, d_{31}, d_{32}, d_{33}, d_{34}, d_{35}, d_{36}, d_{37}, d_{38}, d_{39}\}$ |

Table 9 shows some discovered semantic links among paper clusters. *Similar* links may exist between clusters at the same clustering level, and the similarity between clusters can be calculated by *Jaccard Coefficient* approach. While clusters at different clustering levels may have semantic links *partOf*, *subCluster* or *equal*, and the probability of these semantic links are set as 1. During the evolution of documentary SLN, conversions among *similar*, *partOf*, *subCluster* and *equal* links would occur due to the changes of the cluster keywords.

**Table 9.** Semantic links discovered between paper clusters.

| Semantic Links | Cluster Pairs |
|---|---|
| *similar* | $c_{11}$—*similar*→$c_{12}$, $c_{11}$—*similar*→$c_{13}$, ... |
| *partOf* | $c_{11}$—*partOf*→$c_{21}$, $c_{12}$—*partOf*→$c_{21}$, ... |
| *subCluster* | $c_{11}$—*subCluster*→$c_{21}$, $c_{11}$—*subCluster*→$c_{31}$, ... |

Table 10 shows some discovered semantic links. The *partOf* link implies the *similar* link. By comparing keyword sets, the *partOf* link can be discovered.

**Table 10.** Semantic links discovered between papers.

| Semantic Links | Paper pairs |
|---|---|
| *similar* | ($d_3$, $d_{38}$), ($d_3$, $d_{39}$), ($d_{38}$, $d_{39}$), ... |
| *sameTopic* | ($d_3$, $d_{38}$) belongs to $c_{12}$, ($d_{38}$, $d_{39}$) belongs to $c_{12}$ , ... |
| *sameAuthor* | ($d_3$, $d_{39}$), ... |
| *refer* | ($d_{38}$, $d_3$), ($d_{38}$, $d_{39}$), ... |
| *cocited* | ($d_{38}$, $d_{39}$) —*cocited*→$d_3$, ... |
| *sequential* | ($d_{38}$, $d_3$), ($d_{38}$, $d_{39}$), ... |

### 6.2 Discover SLN among Web Pages on Dunhuang Culture

The proposed approach is suitable for discovering and describing semantic links on Web pages. The possible semantic links among Web pages are listed in Table 11.

**Table 11.** Semantic links between Web pages.

| Field Name | Semantic Links |
|---|---|
| page body text | *similar*, *partOf*, *equal* |
| author/creator | *sameAuthor*, *coAuthor* |
| URL | *sameSite*, *subSite* |
| date | *sequential*, *sameDate* |
| Tag | *sameTopic*, *similarTopic* |
| language | *sameLanguage* |
| anchor | *refer*, *cocite*, *cocited* |

Constructing SLN for Web pages needs to clean Web pages because the Web pages including noise data such as HTML tags, Javascript code and even Ads. After the Web pages are turned into regular documents, the SLN for the Web can be constructed in the similar way as scientific papers.

Comparing with the *citation* link between papers, semantic links derived from the Web page anchors have more types. The *refer* link becomes a more general hyperlink when it links to a concrete resource such as an image, a HTML page, or a piece of multimedia. Citations among documents closely relate to content, while anchors of Web pages are just URLs.

To discover SLN from Web pages on Dunhuang cave culture, we collect Web pages according to URLs returned from Google search engine with the keywords 'dunhuang China', 'dunhuang caves' and 'dunhung Mogao'. Two hundred URLs are crawled. After cleaning noise Web pages, we get 116 Web pages.

Each Web page pair similarity values are calculated, and the number of *similar* edges is $C_{116}^2 = 6670$. We cluster the Web pages by *adding edges*. The times of clustering are 3 with the edge numbers 30, 60 and 90.

Table 12 lists several clusters and their Web pages in this 3-time interactive clustering process. $p_i$ denotes a Web page, and $c_i$ denotes a Web page cluster. Each Web page links to the corresponding cluster with semantic link *instanceOf*. At the initiation stage, the probability values of the *instanceOf* links are set as 1. Thereafter, if new documents come, the semantic link probability can be calculated by document classification rules. Table 13 shows some semantic links among Web page clusters.

Table 14 lists some semantic links among Web pages. From the Web page body text, the *similar* and *equal* links are found. The *partOf* link implies the *similar* link, and the *partOf* link can be discovered by comparing keyword sets. From the URLs, semantic link *sameSite* can be found. According to the clustering results, *sameTopic* links can be discovered. Considering the link structure of Web pages, the *co-link* and *co-linked* links can be discovered.

The proposed approach can be used to establish a semantic overlay on Web2.0 resources to support more advanced Web applications.

**Table 12.** Web page clusters (each page belongs to a cluster via the *instanceOf* link).

| Cluster | Web pages in the cluster |
|---|---|
| $c_{11}$ | $\{p_{88}, p_{89}, p_{90}\}$ |
| $c_{12}$ | $\{p_5, p_{14}, p_{31}, p_{29}, p_{43}, p_{44}, p_{45}, p_{17}, p_{25}, p_2\}$ |
| $c_{13}$ | $\{p_1, p_{80}, p_{81}, p_{82}, p_{99}, p_{109}\}$ |
| $c_{21}$ | $\{p_{88}, p_{89}, p_{90}\}$ |
| $c_{22}$ | $\{p_{21}, p_{85}, p_{93}\}$ |
| $c_{23}$ | $\{p_{40}, p_{51}, p_{62}, p_{73}\}$ |
| $c_{24}$ | $\{p_{51}, p_{74}, p_{43}, p_{29}, p_{41}, p_{17}, p_{18}, p_{19}, p_{14}, p_{45}, p_{68}, p_2, p_{25}, p_1, p_8, p_{82}, p_{81}\}$ |
| $c_{31}$ | $\{p_{88}, p_{89}, p_{90}\}$ |
| $c_{32}$ | $\{p_{40}, p_{51}, p_{62}, p_{73}\}$ |
| $c_{33}$ | $\{p_{59}, p_{48}, p_{41}, p_1, p_{66}, p_{43}, p_1, p_{21}, p_{70}, p_{93}, p_{85}, p_1, p_{31}, p_{29}, p_{57}, p_2, p_{34}, p_{71}, p_{25}, p_{35}, p_{65}, p_{19}, p_{44}, p_5, p_{74}, p_{18}, p_{13}, p_{45}, p_{79}, p_{17}, p_{14}, p_{68}, p_5, p_1, p_{99}, p_1, p_{80}, p_{81}, p_{92}, p_{82}\}$ |

**Table 13.** Semantic links discovered between Web page clusters.

| Semantic links | Cluster Pairs |
|---|---|
| *similar* | $c_{12}$—*similar*→$c_{13}$ |
| *partOf* | $c_{12}$—*partOf*→$c_{24}$, $c_{13}$—*partOf*→$c_{24}$, … |
| *subCluster* | $c_{12}$—*subCluster*→$c_{24}$, $c_{13}$—*subCluster*→$c_{24}$, … |
| *equal* | $c_{11}$—*equal*→$c_{21}$, $c_{21}$—*equal*→$c_{31}$, $c_{23}$—*equal*→$c_{32}$, … |

**Table 14.** Semantic links discovered between Web pages.

| Semantic links | Web page pairs |
|---|---|
| *similar* | $\{p_{41}, p_{48}\}$, $\{p_{21}, p_{85}\}$, $\{p_1, p_{19}\}$, ... |
| *equal* | $\{p_4, p_7\}$, $\{p_{10}, p_{11}\}$, $\{p_{40}, p_{51}\}$, ... |
| *sameTopic* | $\{p_{25}, p_{45}\}$ belongs to $c_{12}$, $\{p_{31}, p_{45}\}$ belongs to $c_{24}$ , ... |
| *sameSite* | $\{p_{88}, p_{89}\}$, $\{p_{88}, p_{90}\}$, $\{p_{89}, p_{90}\}$, ... |

## 7. CONCLUSION

This paper proposes an approach to automatically discover Semantic Link Networks in a given document set. It has the following distinguished advantages: (1) It does not rely on any pre-defined ontology. (2) It has the evolution ability. New semantic links can be derived from the evolving Semantic Link Network. The document classification rules, inference rules and cluster association networks automatically evolve with the change of the network. (3) The approach itself can adapt to the update of the adopted techniques. Experiments on document sets of different type and different scale show the proposed approach feasible. The approach can be used to automatically construct a semantic overlay on a given document set to support advanced applications like various relation queries on documents [14].

## REFERENCES

[1] B. Aleman-Meza, C. Halaschek-Wiener, I. B. Arpinar, and A. P. Sheth. Context-aware semantic association ranking. In *Semantic Web and Databases Workshop Proceedings*, pages 33–50, 2003.

[2] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web* (*WWW 2005*), pages 117–127, New York, NY, USA, 2005. ACM Press.

[3] G. Attardi, A. Gulli, and F. Sebastiani. Automatic Web Page Categorization by Link and Context Analysis. In *Proceedings of European Symposium on Telematics, Hypermedia and Artificial Intelligence* (*THAI*), pages 105–119, 1999.

[4] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web* (*WWW 2007*), pages 131–140, New York, NY, USA, 2007. ACM Press.

[5] S. Harabagiu, F. Lacatusu, and A. Hickl. Answering complex questions with random walk models. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (*SIGIR 2006*), pages 220–227, New York, NY, USA, 2006. ACM Press.

[6] T. H. Haveliwala, A. Gionis, D. Klein, and P. Indyk. Evaluating strategies for similarity search on the web. In *Proceedings of the 11th international conference on World Wide Web* (*WWW 2002*), pages 432–442, New York, NY, USA, 2002. ACM Press.

[7] G. Luo, C. Tang, and Y. li Tian. Answering relationship queries on the web. In *Proceedings of the 16th international conference on World Wide Web* (*WWW 2007*), pages 561–570, New York, NY, USA, 2007. ACM Press.

[8] D. Mahler. Holistic query expansion using graphical models. In *New Directions in Question Answering*, pages 203–214, 2004.

[9] F. Menczer. Mapping the semantics of Web text and links. *IEEE Internet Computing*, 9(3):27–36, 2005.

[10] S. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive. *Proceedings of the 7th text retrieval conference* (*TREC-7*)*, NIST special publication*, pages 500–242, 1998.

[11] A. Singhal. Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.

[12] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. *Proc. AAAI Workshop on AI for Web Search* (*AAAI 2000*)*, Austin*, pages 58–64, 2000.

[13] H. Zhao. Semantic matching across heterogeneous data sources. *Commun. ACM*, 50(1):45–50, 2007.

[14] H. Zhuge. Autonomous semantic link networking model for the knowledge grid. *Concurrency and Computation: Practice and Experience*, 7(19):1065–1085, 2007.

[15] H. Zhuge, R. Jia, and J. Liu. Semantic link network builder and intelligent browser. *Concurrency and Computation: Practice and Experience*, 16(14):1453–1476, 2004.

[16] H. Zhuge and L. Zheng. Ranking semantic-linked network. In *Proceedings of the 12th international conference on World Wide Web* (*WWW 2003*), Budapest, May 2003.

[17] H.Zhuge, Active e-Document Framework ADF: Model and Platform, *Information and Management*, 41(1): 87-97, 2003.

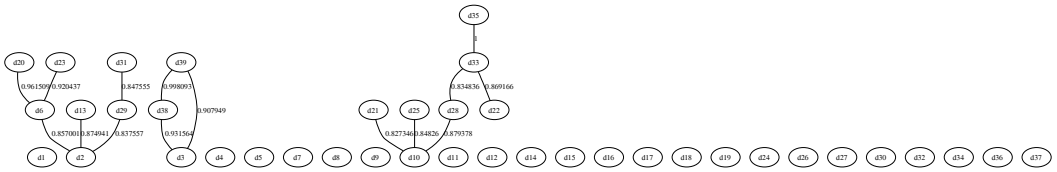
[18] H.Zhuge and X.Li, Peer-to-Peer in Metric Space and Semantic Space, *IEEE Transactions on Knowledge and Data Engineering*, 6(19) (2007) 759-771.

## APPENDIX

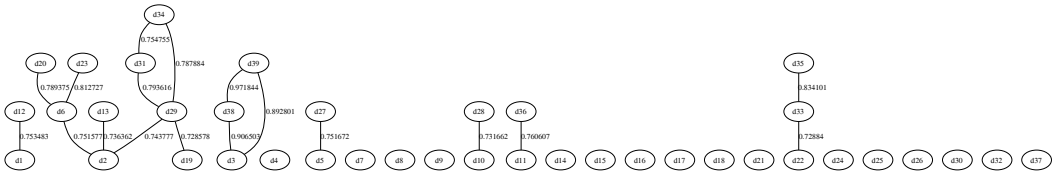
**Table 15.** 39 papers on Semantic Web.

| Paper ID | Title and URL |
|---|---|
| d1 | A comparison of implicit and explicit links for web page classification |
| d2 | A link-based ranking scheme for focused search |
| d3 | Context-aware semantic association ranking |
| d4 | A clustering method for web data with multi-type interrelated components |
| d5 | A large-scale evaluation and analysis of personalized search strategies |
| d6 | A link classification based approach to website topic hierarchy generation |
| d7 | Answering bounded continuous search queries in the world wide web |
| d8 | Answering relationship queries on the web |
| d9 | Bridging the gap between OWL and relational databases |
| d10 | Building and managing personalized semantic portals |
| d11 | Compare&Contrast: using the web to discover comparable cases for news stories |
| d12 | Demographic prediction based on user's browsing behavior |
| d13 | Dynamic personalized pagerank in entity-relation graphs |
| d14 | Efficient search engine measurements |
| d15 | Efficient search in large textual collections with redundancy |
| d16 | Evaluating strategies for similarity search on the Web |

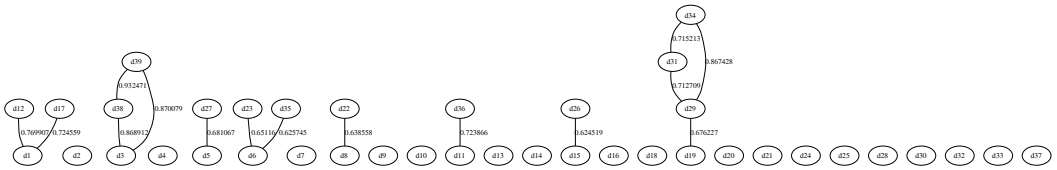
| | |
|---|---|
| d17 | Exploring in the weblog space by detecting informative and affective |
| d18 | Extraction and classification of dense communities in the Web |
| d19 | Finding related pages on the link structure of the WWW |
| d20 | From SPARQL to rules (and back) |
| d21 | Graphical models for probabilistic and causal reasoning |
| d22 | Holistic query expansion using graphical models |
| d23 | Identifying ambiguous queries in Web search |
| d24 | Internet-scale collection of human-reviewed data |
| d25 | Measuring semantic similarity between words using web search engines |
| d26 | Navigating the Intranet with high precision |
| d27 | Navigation-aided retrieval |
| d28 | Ontology summarization based on RDF sentence Graph |
| d29 | Optimizing web search using social annotations |
| d30 | Privacy-enhancing personalized web search |
| d31 | P-TAG: large scale automatic generation of personalized annotation tags for the Web |
| d32 | Supervised rank aggregation |
| d33 | Tag clouds for summarizing web search results |
| d34 | Towards effective browsing of large scale social annotations |
| d35 | Web page classification with heterogeneous data fusion |
| d36 | Why we search: visualizing and predicting user behavior |
| d37 | Autonomous semantic link networking model for the Knowledge Grid |
| d38 | SemRank: ranking complex relationship search results on the Semantic Web |
| d39 | p-Queries: enabling querying for semantic associations on the Semantic Web |



*cos*

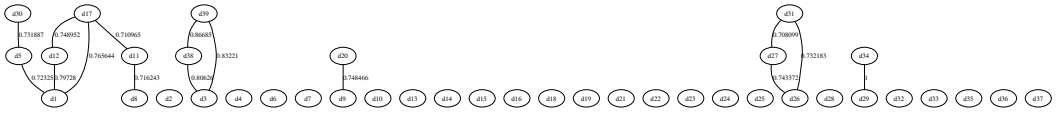


$0.8cos + 0.2jaccard$



$0.5cos + 0.5jaccard$

$0.2cos + 0.8jaccard$



*jaccard*

**Figure 10**. Top 15 high similar edges for cluster with different similarity calculation methods.

**Table 16**. Top 15 most similar document pairs.

| *cos* | 0.8*cos*+0.2*jaccard* | 0.5*cos*+0.5*jaccard* | 0.2*cos*+ 0.8*jaccard* | *jaccard* |
|---|---|---|---|---|
| (d33,d35) | (d38,d39) | (d38,d39) | (d29,d34) | (d29,d34) |
| (d38,d39) | (d3,d38) | (d3,d39) | (d38,d39) | (d38,d39) |
| (d6,d20) | (d3,d39) | (d3,d38) | (d3,d39) | (d3,d39) |
| (d3,d38) | (d33,d35) | (d29,d34) | (d3,d38) | (d3,d38) |
| (d6,d23) | (d6,d23) | (d1,d12) | (d1,d12) | (d1,d12) |
| (d3,d39) | (d29,d31) | (d1,d17) | (d1,d17) | (d1,d17) |
| (d10,d28) | (d6,d20) | (d11,d36) | (d5,d30) | (d12,d17) |
| (d2,d13) | (d29,d34) | (d31,d34) | (d11,d36) | (d9,d20) |
| (d22,d33) | (d11,d36) | (d29,d31) | (d8,d11) | (d26,d27) |
| (d2,d6) | (d31,d34) | (d5,d27) | (d31,d34) | (d26,d31) |
| (d10,d25) | (d1,d12) | (d19,d29) | (d9,d20) | (d5,d30) |
| (d29,d31) | (d5,d27) | (d6,d23) | (d11,d17) | (d1,d5) |
| (d2,d29) | (d2,d6) | (d8,d22) | (d26,d31) | (d8,d11) |
| (d28,d33) | (d2,d29) | (d6,d35) | (d15,d26) | (d11,d17) |
| (d10,d21) | (d2,d13) | (d15,d26) | (d26,d27) | (d27,d31) |

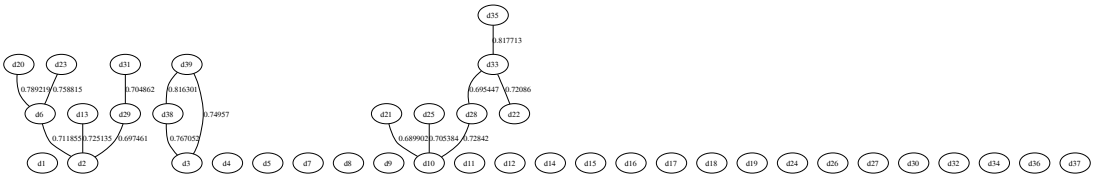



**Figure 11**. Top 15 high similar edges for cluster.

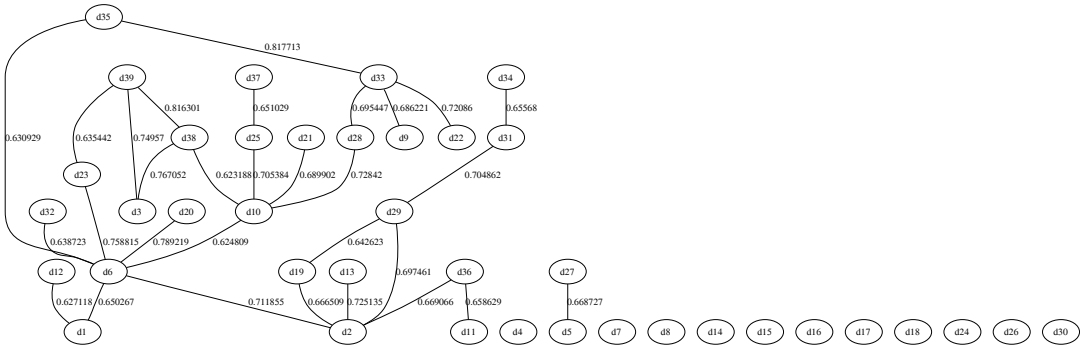



**Figure 12**. Top 30 high similar edges for cluster.

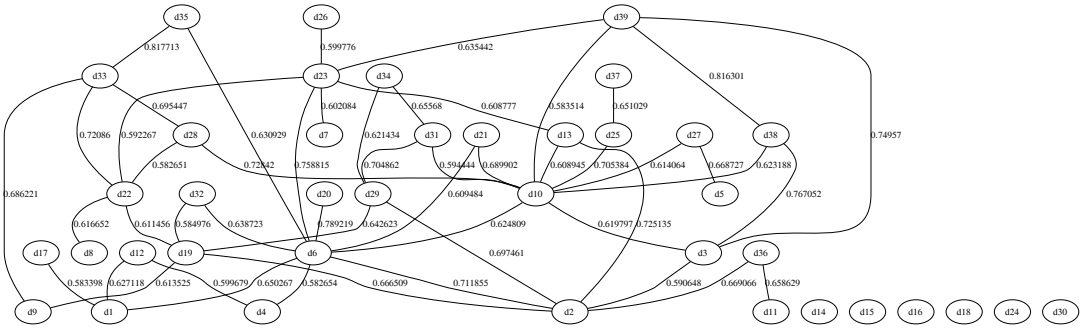



**Figure 13**. Top 50 high similar edges for cluster.

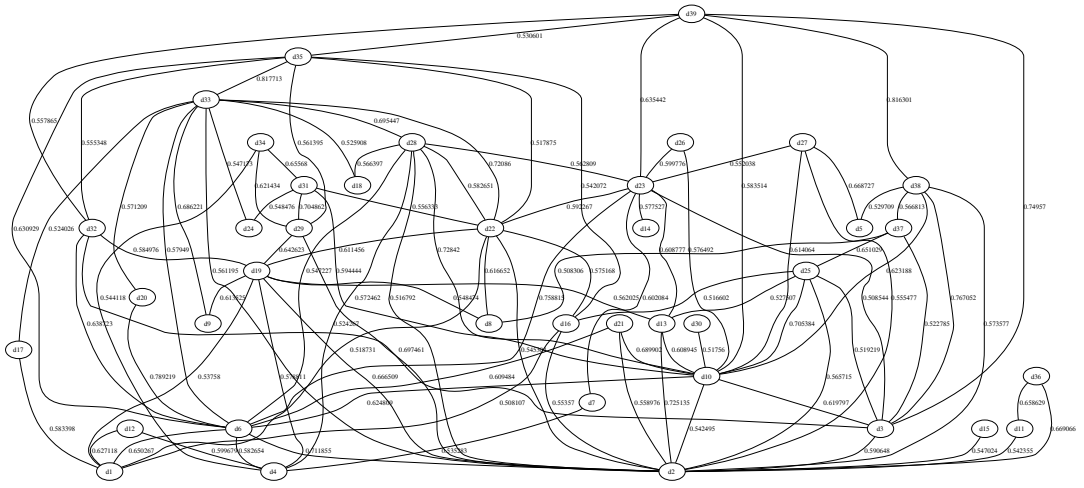


**Figure 14**. Top 100 high similar edges for cluster.

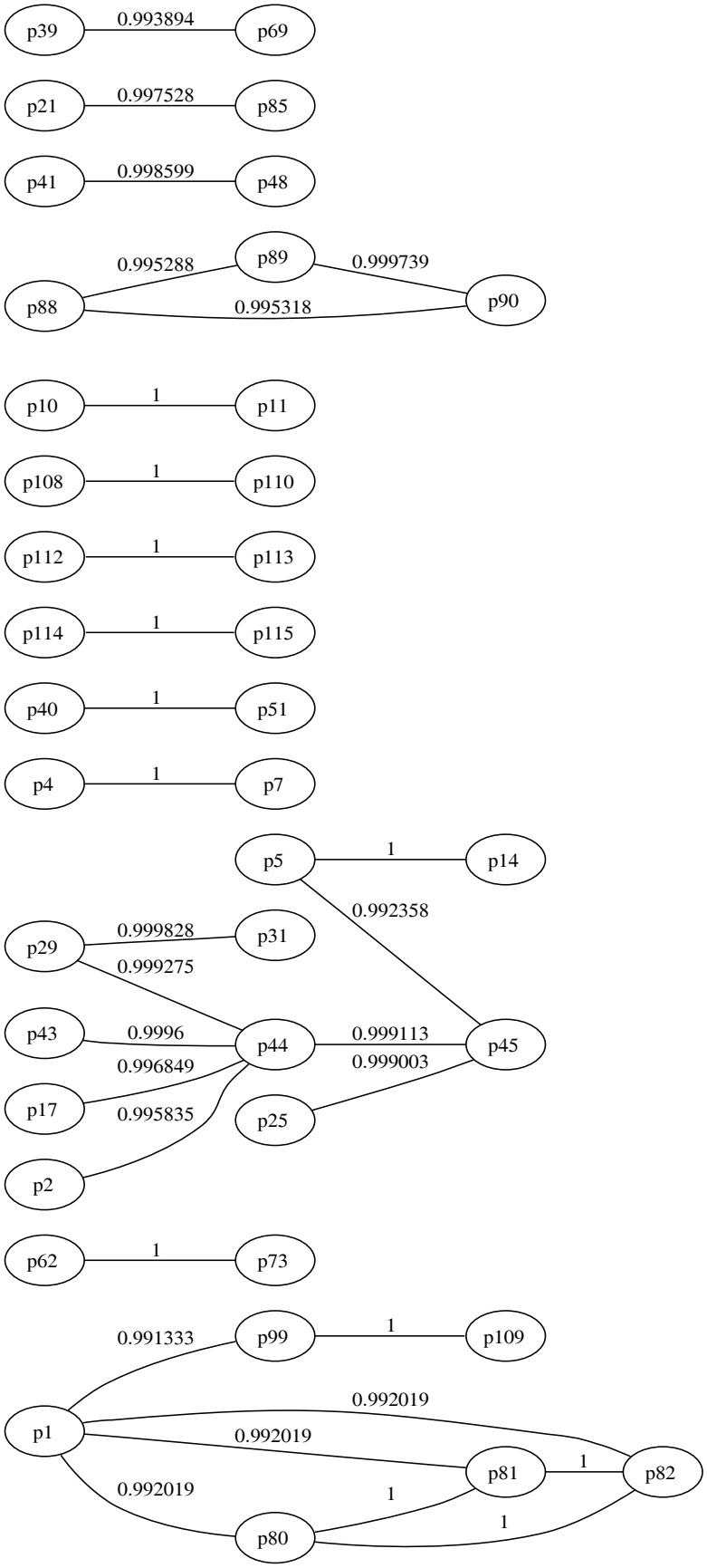



**Figure 15**. Top 30 high similar edges for Web pages clustering.

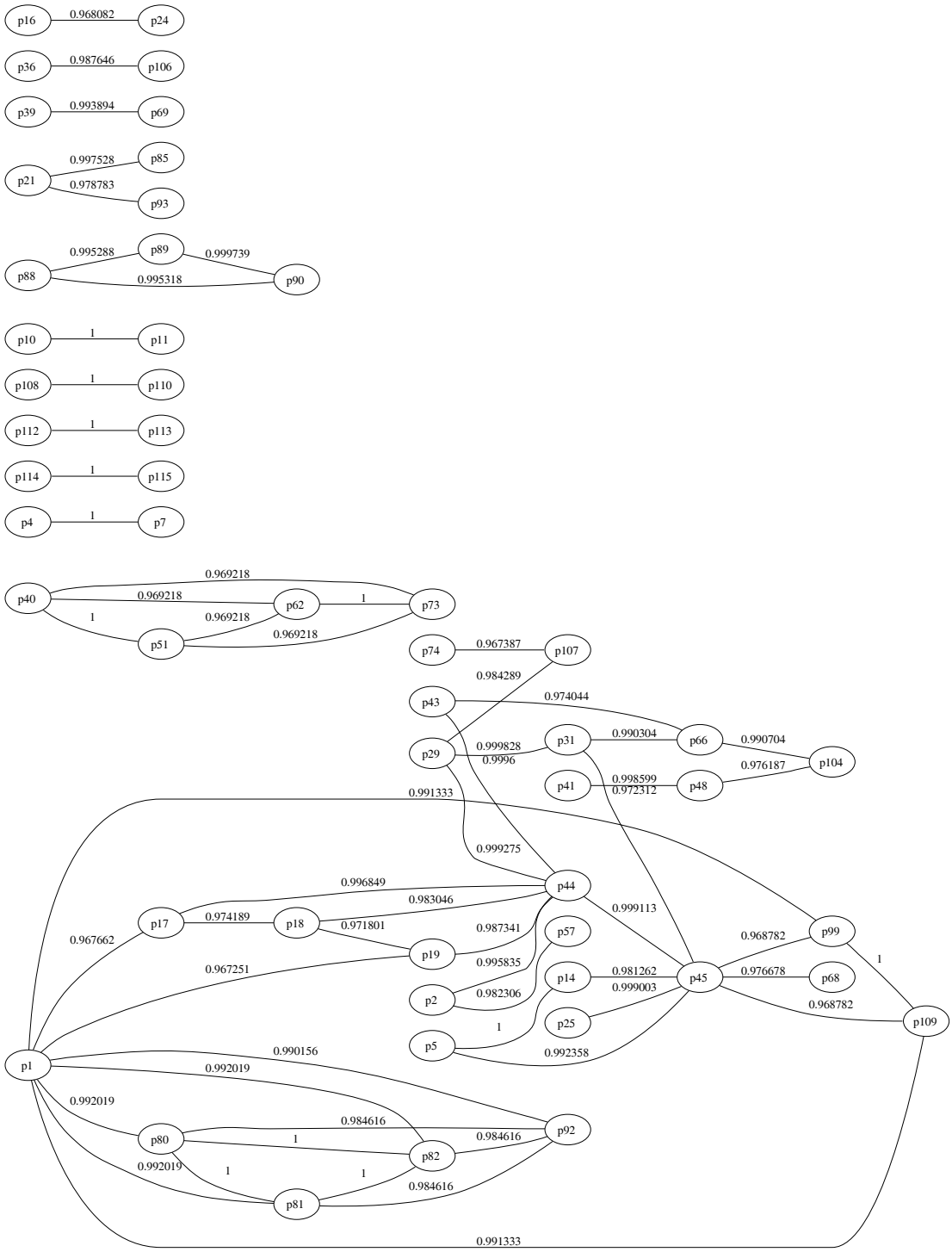



**Figure 16**. Top 60 high similar edges for Web pages clustering.

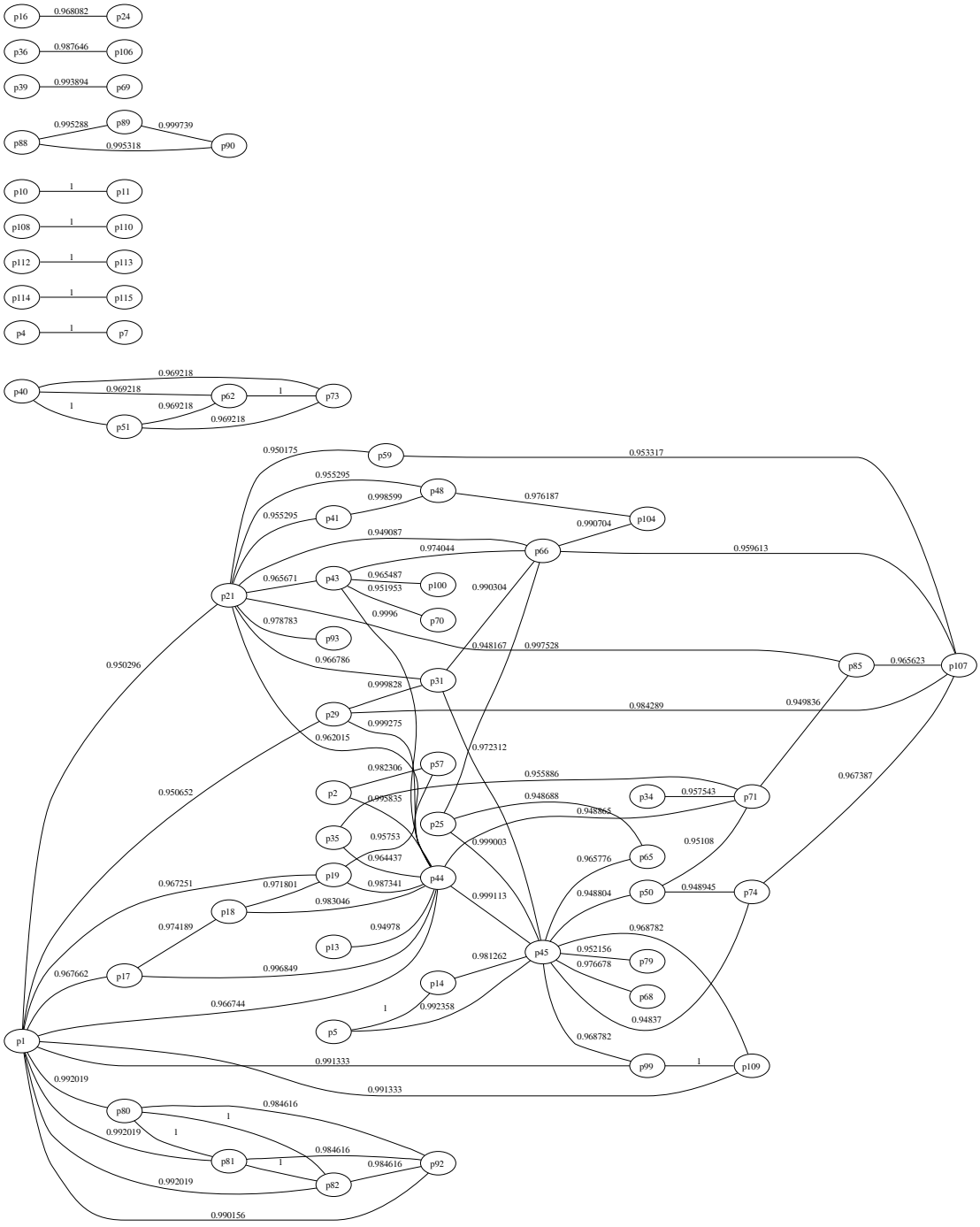



**Figure 17**. Top 90 high similar edges for Web pages clustering.