IEICE TRANS. INF. & SYST., VOL.E95–D, NO.7 JULY 2012

PAPER

Mining and Explaining Relationships in Wikipedia

Xinpeng ZHANG[†*a)], ***Nonmember***, **Yasuhito ASANO**[†], ***and*** **Masatoshi YOSHIKAWA**[†], ***Members***

SUMMARY Mining and explaining relationships between concepts are challenging tasks in the field of knowledge search. We propose a new approach for the tasks using disjoint paths formed by links in Wikipedia. Disjoint paths are easy to understand and do not contain redundant information. To achieve this approach, we propose a naive method, as well as a generalized flow based method, and a technique for mining more disjoint paths using the generalized flow based method. We also apply the approach to classification of relationships. Our experiments reveal that the generalized flow based method can mine many disjoint paths important for understanding a relationship, and the classification is effective for explaining relationships.

key words: *link analysis, generalized max-flow, Wikipedia mining, relationship*

1. Introduction

Wikipedia is widely used for searching knowledge of concepts, such as humans, places or events. In Wikipedia, the knowledge of a concept is gathered in a single page updated constantly by a number of volunteers. Wikipedia covers concepts in numerous categories, such as people, science, geography, politics, and history. Therefore, Wikipedia is usually a better choice than typical keyword search engines for searching knowledge of a single concept.

A user might desire to search not only knowledge about a single concept, but also knowledge about a relationship between two concepts. For example, a user may desire to know the relationship between petroleum and a certain country, or to know a relationship between two politicians. Typical keyword search engines offer an easy way for searching web pages related to two concepts. For example, we could search web pages related to "Japan" and "Russia" using a query "Japan Russia." 11 of the top-20 pages searched using the query contain some information about the relationship between "Japan" and "Russia." However, it is often difficult for a user to find and organize the information about a relationship from numerous search result web pages. This is because some pages contain no knowledge about the relationship, such as a blog page where "japan" and "Russia" appear independently in different diaries; some pages are very long; and some pages contain many duplicate information. In Sect. 5.2.1, we conduct experiments to confirm that most of the top-20 web pages searched by a keyword search engine using a query "*s t*" are inadequate for understanding the relationship between *s* and *t*.

The main issue for analyzing relationships arises from the fact that two kinds of relationships exist: "explicit relationships" and "implicit relationships." In Wikipedia, an explicit relationship is represented as a link. A user could understand an explicit relationship easily by reading text surrounding the anchor text of the link. For example, an explicit relationship between petroleum and plastic might be represented by a link from page "Plastic" to page "Petroleum." A user could understand its meaning by reading the text "plastic is mainly produced from *petroleum*" surrounding the anchor text "petroleum" on page "Plastic." An implicit relationship is represented by multiple links and pages in Wikipedia. For example, the Gulf of Mexico is a major oil producer in the USA. This fact could be an implicit relationship represented by two links in Wikipedia: one between "Petroleum" and "Gulf of Mexico" and the other one between "Gulf of Mexico" and the "USA." It is difficult for a user to discover or understand an implicit relationship without investigating a number of pages and links. Keyword search engines are also unable to search such an implicit relationship between *s* and *t* described in multiple Wikipedia pages using a query "*s t*." Therefore, it is an interesting problem to mine and explain an implicit relationship in Wikipedia.

Measuring the strength of an implicit relationship is one approach for explaining the relationship. Several methods [1], [2] have been proposed for measuring relationships on an *information network* (V, E), a directed graph where V is a set of concepts; edges in E represents explicit relationships between concepts. We can define a *Wikipedia information network* whose vertices are pages of Wikipedia and edges are links between pages. Zhang et al. [2] model a relationship between two concepts in a Wikipedia information network using a generalized max-flow. The model reflects all three concepts important for measuring relationships: distance, connectivity and co-citation. Zhang et al. [2] ascertained a method using the model that can measure the strength of an implicit relationship in Wikipedia more correctly than previous methods [1], [3], [4] can do. In addition to quantitatively measuring the strength of a relationship, it is also very important to inquire about the quality of information about relationship, as the following problems. (P1) Can a relationship be displayed to users in a easy-to-understand manner? (P2) Can we explain the basis of the

Manuscript received August 9, 2011.
Manuscript revised January 23, 2012.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
[*]Presently, with National Institute of Information and Communications Technology.
a) E-mail: xp.zhang@nict.go.jp

DOI: 10.1587/transinf.E95.D.1918

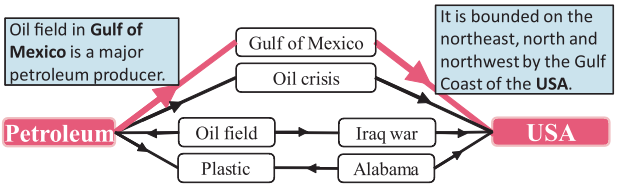

Fig. 1 Explaining the relationship between Petroleum and the USA.

strength of a relationship? And, what constitute a relationship? Such problems have not been well studied so far. Therefore, in this paper, we propose solutions to solve the problems.

Regarding problem (P1), we performed a user questionnaire which will be discussed in Sect. 2 and found that, in order to explain a relationship between two concepts s and t on an information network, it is more useful to display disjoint s-t paths than to present a subgraph containing the two concepts. Disjoint s-t paths are paths connecting s and t sharing no vertices except s and t with each other. For example, four disjoint paths linking "Petroleum" to "USA" depicted in Fig. 1, explain the implicit relationship between petroleum and the USA. A user could understand the meaning of a path easily by tracing the links in the path from s to t. Tracing each link can be done by understanding the meaning of an explicit relationship represented by the link. For example, if users read the snippets shown in Fig. 1 from left to right, then they can understand the top path containing "Gulf of Mexico". They can understand why the Gulf of Mexico is important to the relationship between petroleum and the USA. An idea of displaying disjoint paths is that the same or similar paths should not appear multiple times in the paths shown for explaining a relationship. A similar idea is widely accepted in the field of document retrieval: a search result should not contain same or similar documents [5]–[7]. However, to the best of our knowledge, there has been no study that mines disjoint paths for explaining a relationship on an information network.

The paths displayed for explaining a relationship should satisfy the following two requirements, (1) the paths are useful for understanding the relationship, and (2) the middle concepts in the paths play important roles in the relationship. To mine paths meeting the two requirements for a relationship in Wikipedia, we first propose a naive method based on CFEC [1], which is a method for measuring the strength of a relationship. The naive method adopts the scheme for computing the weight of a path of CFEC, although it cannot mine disjoint paths. We then propose a method to mine disjoint paths important for a relationship, based on the generalized max-flow model proposed by Zhang et al. [2]. For a relationship between two concepts s and t, we first compute a generalized max-flow emanating from s to t. We then obtain paths along which a large amount of flow is sent as paths important for understanding the relationship. A generalized flow can be confluent at a vertex except s and t. Therefore, the obtained paths might contain some dependent paths. To force a generalized flow to be sent along disjoint paths as much as possible, we propose a new technique using vertex capacities, in this paper. We confirm through experiments discussed in Sect. 5.2.3 that the new technique is useful for mining more disjoint paths. The experimental results also reveal that the doubled network proposed in [2] is effective in mining more disjoint paths for the naive method. However, the naive method still cannot reach the generalized flow based method for mining disjoint paths.

Concerning problem (P2), we evaluate the mined paths through experiments using human subject, as described in Sect. 5.2. The participants in the experiments judged that the generalized flow based method mines more paths important for a relationship than the method based on CFEC does. We also ascertained that many concepts in the paths mined by the generalized flow based method play important roles in constituting a relationship.

As an application of our approach, we propose a method for classifying relationships between a common source concept and different destination concepts, e.g. relationships between petroleum and countries, by analyzing mined paths for the relationships. For this example, the method classifies the countries into two groups which could correspond to "petroleum exporting countries" and "petroleum consuming countries."

The rest of this paper is organized as follows. Section 3 reviews related work. Section 4 presents the methods for mining paths important for a relationship in Wikipedia, and the method for classifying relationships. Section 5 reports the experimental results. Section 6 concludes the paper.

A preliminary version of the paper was presented in [8].

2. Mining Disjoint Paths for Explaining Relationships

Users prefer not to read similar documents repeatedly, and they might desire to obtain various kinds of knowledge by reading small number of documents. Therefore, recent document information retrieval methods [5]–[7] adopt an idea that redundant information should be minimized in the top-ranked documents by removing documents similar to a higher ranked documents. For example, given a query "foreign relationships of the USA," a set of the top-ranked documents should cover relationships between the USA and various countries, the set should not contain a number of similar documents explaining the relationship between the USA and a certain country.

Applying the idea to the problem of mining paths important for a relationship on an information network, we should avoid outputting redundant concepts in the mined paths. Disjoint paths connecting two concept s and t are paths sharing no vertices except s and t with each other. If we could mine disjoint paths connecting s and t, we then could prevent a concept except s and t from appearing multiple times in the mined paths.

Figure 2 (A) and (B) depict graphs constituted by three dependent paths and three disjoint paths, respectively. Both

graphs explain the relationship about the territorial problem between Japan and Russia. All the three dependent paths depicted in Fig. 2 (A) contain the same concept "Northern Territories dispute" which represents the sovereignty dispute between Japan and Russia on the South Kuril Islands, including Shikotan, Kunashiri and Habomai rocks. On the other hand, the three disjoint paths depicted in Fig. 2 (B) contains no redundant concept. We conducted a questionnaire on 1000 participants to compare the two graphs through several questions. As one of these questions, we asked the participants to select a better one of the two graphs for understanding the relationship between Japan and Russia, and to explain reasons for their selection. As presented in Fig. 3, 52.0% of the 1000 participants thought the graph depicted in Fig. 2 (B) is better. Most of them gave the reason that the graph depicted in Fig. 2 (B) contains more knowledge about the relationship between Japan and Russian than the graph depicted in Fig. 2 (A) does. If a user knows about the Northern Territories dispute, then the user could not get any new knowledge from the dependent paths depicted in Fig. 2 (A). By reading the graph depicted in Fig. 2 (B), a user could obtain knowledge other than the Northern Territories dispute, such as knowledge about the "Soviet-Japanese Joint Decalaration of 1956" and the "Treaty of Shimoda." On the other hand, only 24.8% of the 1000 participants thought the dependent paths depicted in Fig. 2 (A) is better for understanding the relationship. Most of the 24.8% participants selected the graph depicted in Fig. 2 (A) because they thought the graph is simpler. We also compared another pair of graphs in the questionnaire, and similar results were obtained. The results of the questionnaire show a large possibility that users prefer disjoint paths for understanding relationships because disjoint paths contain more diverse knowledge than dependent paths. The possibility become one of our motivations for mining disjoint paths.

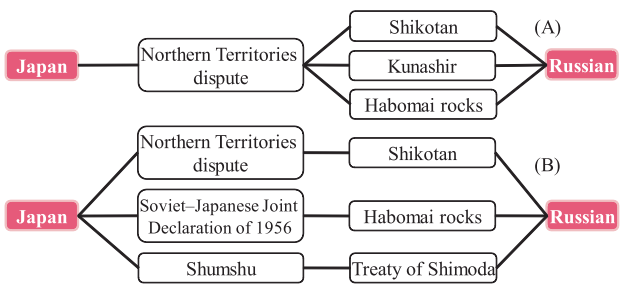

Fig. 2 Dependent paths (A) and disjoint paths (B) explaining the relationship between "Japan" and "Russian."

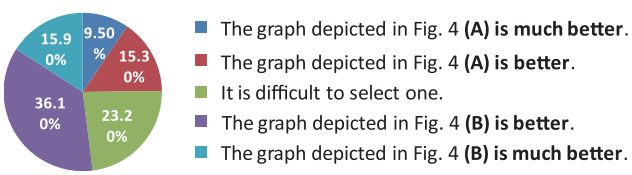

Fig. 3 Rate of participants according to their selection.

As another motivation for mining disjoint paths, it is easy to understand the meaning of each disjoint path by tracing the links in the path from left to right, as discussed in Sect. 1. In contrast, dependent paths make a graph be too complicated to find out the order of tracing links in some cases, such as those in the graph depicted in Fig. 4. Therefore, in this paper, we aim to mine disjoint paths important for a relationship on an Wikipedia information network.

3. Related Work

Measuring the strength of an implicit relationship is one approach for explaining the relationship. Zhang et al. [2] model a relationship between two concepts in a Wikipedia information network using a generalized max-flow. They ascertained a method using the model that can measure the strength of an implicit relationship more correctly than previous methods can do [1], [3], [4]. Several kinds of questions about relationships can be answered by measuring relationships. For example, a user could know which one of two specified countries has a stronger relationship to petroleum. However, measuring strength alone is insufficient for understanding relationships. A user would desire to know what concepts constitute a relationship or what roles they play in the relationship.

Another approach to explain relationships might be extracting a "connection subgraph" [1], [9]–[11]. Faloutsos et al. [9] model an information network as an electric network [12], and model the weight of a path as the current delivered by the path. Given two query vertices s and t and an undirected graph G, they extract a connected subgraph H containing s and t and limited number of other vertices that maximize the weight of H, the sum of the weights of all the paths in H. Extending the problem into more than two query vertices, Tong and Faloutsos [10] proposed *CEPS* problem. Koren et al. [1] proposed CFEC to outputs a small subgraph on which the strength of the relationship measured approximates to that measured on the original graph. The method proposed by Cheng et al. [11] first partitions a network into a set of communities by considering the domains of concepts. The method then focusing on extracting a connected subgraph in each community to which query vertices belongs.

Figure 4 is an example of a connection subgraph presented by Faloutsos et al. [9]. Vertices in a connection subgraph represent concepts that are considered important for a relationship. Therefore, a user could know what concepts constitute a relationship. However, it is still difficult to know what roles the concepts play in the relationship using the connection subgraph. In a connection subgraph that is constituted by dependent paths, several paths

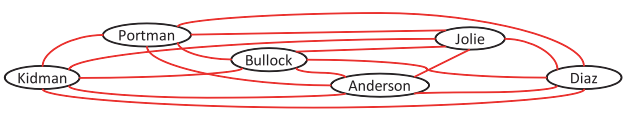

Fig. 4 A snapshot of a connection subgraph for the relationship between Kidman and Diaz.

go through a common concept. For the above example, over 20 paths connecting "Kidman" and "Diaz" go through "Anderson," such as (*Kidman*, *Bullock*, *Anderson*, *Diaz*), (*Kidman*, *Anderson*, *Jolie*, *Diaz*). "Anderson" plays different roles of constituting the relationship in different paths. However, a connection subgraph can not present how to read the subgraph and how to trace each path contained in it. It is difficult for a user to find and understand the paths contained in such a complicated subgraph oneself. Consequently, a connection subgraph is inadequate for understanding a relationship.

To create a connection subgraph for a relationship, the methods [1], [9], [11] discussed above first compute the weights of paths using random walk [12]. They define the weight of a path fundamentally as the product of the weights of the edges composing the path divided by the product of the weights of the edges incident to every vertex in the path. Therefore, random walk based methods have a property that they compute the weight of a path extremely small if a popular concept—a concept linked by or from many other concepts—exists in the path. However, a popular concept might play important role in a relationship in Wikipedia. We claim that this property is unsuitable for mining paths important for a relationship through experiments discussed in Sect. 5.2.

Several studies have been made on giving semantics to explicit relationships. The DBpedia project [13] extracts semantic relationships between concepts from the infobox in Wikipedia articles. For example, the relationship that "Tokyo" is the "*captial*" of "Japan", could be obtained from the infobox of article "Japan." Lehmann et al. [14] then developed a tool for exploring relationships extracted by DBpedia. NAGA [15] is also a semantic search engine which searches semantic relationships, using a semantic knowledge base YAGO [16] extracted from Wikipedia and WordNet. Several methods [17], [18] were proposed for extracting paths between two concepts on an RDF graph whose edges represent explicit relationships with semantics. Both DBpedia and YAGO do not extract knowledge represented by the links existing in the body text of Wikipedia articles. Many explicit relationships existing in the Wikipedia information network are not included in DBpedia or YAGO. Therefore, we do not use these semantic knowledge base as the information source for mining relationships. We propose methods for mining relationships on the Wikipedia information network, in this paper. However, it is able to construct an information network using a semantic relationship database, and to apply our method proposed in Sect. 4.4 on the information network.

EntityCube [19] and SPYSEE [20] extract explicit relationships between pairs of people from the Web. They then search people having explicit relationship to a given people. Both EntityCube and SPYSEE do not explain an implicit relationship between two people.

4. Methods for Mining Disjoint Paths in Wikipedia

We now present our methods for mining disjoint paths important for explaining a relationship. To the best of our knowledge, no such method was proposed. We first propose a naive method based on CFEC [1] in Sect. 4.1. The weight of a path defined in [9], [11] are similar to that of CFEC. Therefore, we only apply CFEC to the naive method.

4.1 Naive Method Based on CFEC

Given a graph G, a source vertex s and a destination vertex t, CFEC first finds the n shortest paths between s and t. It then computes the strength of the relationship between s and t using random walk on the paths [12]. In CFEC, the weight of a path $p = (s = v_1, v_2, \ldots, v_\ell = t)$ from s to t is defined as $w(v_1, v_2) \cdot \prod_{i=2}^{\ell-1} \frac{w(v_i, v_{i+1})}{w_{sum}(v_i)}$, where $w(u, v)$ is the weight of edge (u, v) and $w_{sum}(v)$ is the sum of the weights of the edges going from vertex v. For example, Fig. 5 depicts two paths between "Rice" and "Koizumi." The number shown beside a vertex o_i is the number of links going from the vertex, which equals to $w_{sum}(o_i)$ if the weight of every edge is 1. The weights of path (A) and path (B) become 1/289 and 1/1265, respectively.

We now propose a naive method for mining paths important for a relationship between a source concept s and a destination concept t in Wikipedia based on CFEC [1].

(1) Construct a network $G = (V, E)$ using pages and links within at most m hop links from s or t in Wikipedia. (2) Set the weight of every edge $e \in E$ to 1, compute the largest-k paths in decreasing order of the path weight. (3) For each edge (u, v) in the largest-k paths, extract an explanatory snippet, i.e., text surrounding the anchor text of link v on page u, using a KWIC concordance tool [21].

The largest-k paths mined by the method probably contain some dependent paths. Although we can select some disjoint paths among the mined paths, we cannot determine in advance how many paths should be mined to obtain a specified number of disjoint paths. Moreover, this method has a problem because of popular concepts, as discussed in Sect. 3. For example, in Fig. 5, the weight of path (B) is significantly smaller than that of path (A), because "Bush" is more popular, i.e., linked from or to more concepts, than "Olmert." Consequently, important paths containing a popular concept seldom appear in the largest-k paths mined by the method.

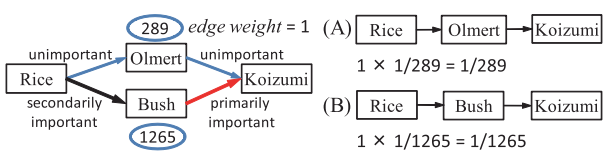

Fig. 5 An example of the relationship between "Rice" and "Koizumi" for CFEC.

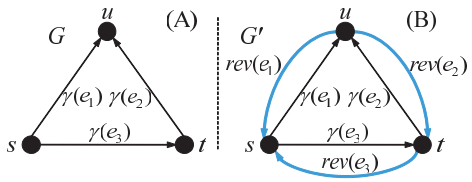

Fig. 6 A doubled network.

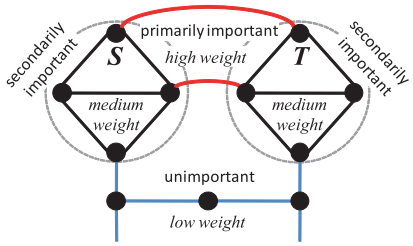

Fig. 7 Assignment of weights for edges.

4.2 Improvements Using Doubled Network and Domain-based Weight

We now discuss two improvements for mining important paths: a doubled network and an edge weight function using the category information on Wikipedia. Both improvements were originally proposed for measuring a relationship [2].

A path constituted by edges of different directions could be important for a relationship in Wikipedia. For example, the path (*Petroleum, Plastic, Alabama, USA*) in Fig. 1 is formed by edges of different directions. The path would correspond to an important fact between "Petroleum" and the "USA" that the Alabama State of the *USA* produces a large quantity of plastic from *petroleum*. To mine such paths, we construct a doubled network [2] by adding to every original edge a reversed edge whose direction is opposite to the original one. For example, Fig. 6 (B) depicts the doubled network G' for G in Fig. 6 (A). It is also possible to construct an undirected network to mine paths constituted of edges of any direction. An undirected network ignores the directions of edges, although a path formed by edges of different directions is usually relatively less important than a path formed by edges of the same direction. Therefore, we construct an doubled network instead of an undirected network. In a doubled network, we set weight $\gamma(e)$ for edge e and set weight $rev(e)$ for the reversed edge of e, where $rev(e) = \lambda \times \gamma(e), 0 \leq \lambda < 1$. λ is used to adjust the importance of a reversed edge. Zhang et al. [2] determine through experiments that $\lambda = 0.8$ is appropriate for analyzing relationships in Wikipedia. Using the doubled network, the naive method was able to mine paths formed by edges of any direction, and estimate paths formed by edges of different directions to be relatively less important.

Paths formed by edges representing important explicit relationships in constituting a relationship, are usually important to the relationship. To mine such paths, let us consider what kinds of explicit relationships are important in constituting an implicit relationship. For the example depicted in Fig. 5, suppose American politician "Rice" is trying to send a message to Japanese politician "Koizumi;" Rice has no explicit relationship to Koizumi, and American politician "Bush" and Israeli politician "Olmert" have respective explicit relationships to Koizumi. Rice could contact Bush easily compared to Olmert because both Rice and Bush belong to the same group "American politician." Therefore, Rice would tend to ask Bush, rather than Olmert, to help transferring the message to Koizumi. The explicit relationship between Bush and Koizumi would be primarily important to the relationship between Rice and Koizumi. Consequently, path (*Rice*, *Bush*, *Koizumi*) containing primarily important edge (*Bush*, *Koizumi*) would be more important than path (*Rice*, *Olmert*, *Koizumi*) to the relationship.

Let a "group" be a set of similar or related concepts, such as American politicians, or Japanese politicians. Zhang et al. [2] observed a number of implicit relationships between concept s in group S and concept t in group T, in Wikipedia, and found the following.

(1) Most of the S-T explicit relationships between a concept in S and a concept in T are primarily important, such as that between "Bush" and "Koizumi" in the above example.

(2) Most of the S-S or T-T explicit relationships between concepts in S or concepts in T are secondarily important, such as that between "Rice" and "Bush" in the above example.

(3) Most of the other explicit relationships connecting concepts in other groups rather than S and T are unimportant, such as that connecting "Rice" and "Olmert" in the above example. However, it does not mean that such explicit relationships are all meaningless in constituting a relationship.

Zhang et al. [2] then adopt the following three assumptions according to the above observations, as illustrated in Fig. 7. (1) S-T explicit relationships are primarily important; (2) S-S or T-T explicit relationships are secondarily important; (3) other explicit relationships are unimportant.

To achieve an edge weight function according to the three assumptions, Zhang et al. [2] first construct groups of concepts in Wikipedia, such as "Japanese politicians" and "baseball players". In Wikipedia, a page corresponding to a concept belongs to at least one category. For example, "George W. Bush" belongs to the category "Presidents of the USA." However, categories cannot be used as groups directly because the category structure of Wikipedia is too fractionalized. Given a category c_i, Zhang et al. [2] construct the group for c_i by grouping c_i and its descendant categories which represent sub concepts of c_i together. S is then the set of concepts belonging to a category in the group for a category of the source. Similarly, we could obtain T for the destination. As illustrated in Fig. 7, Zhang et al. [2] then assign

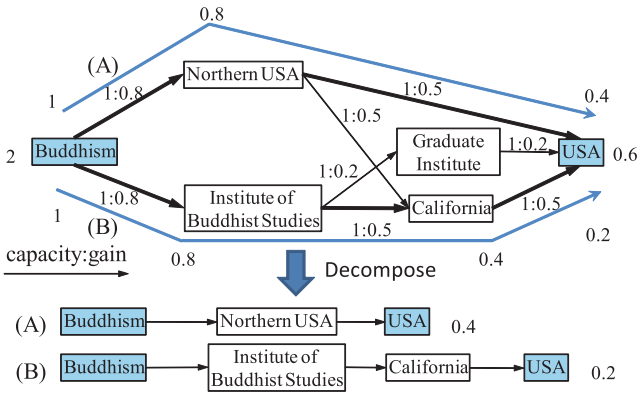

Fig. 8 A generalized max-flow and its decomposition.

a high weight, a medium weight and low weights to edges representing primarily important, secondarily important and unimportant explicit relationships, respectively. Zhang et al. [2] ascertained the appropriateness of the edge weights function for analyzing relationships in Wikipedia through experiments. We omit the details of the assignment of the edge weight here. Please refer to [2] for the details. By applying the edge weight function to the naive method, the weights of paths containing important edges become larger than those of paths containing unimportant edges relatively. Therefore, the weight function would be useful for mining important paths.

4.3 Generalized Max-Flow Model

The naive method was unable to mine disjoint paths, even after using the doubled network and the weight function discussed in Sect. 4.2. We propose a generalized flow based method that could mine disjoint paths in Sect. 4.4. Before introducing the method, we explain its basis: the generalized max-flow model proposed by Zhang et al. [2] for computing the strength of a relationship.

The generalized max-flow problem [22], [23] is identical to the classical max-flow problem except that every edge e has a gain $\gamma(e) > 0$; the value of a flow sent along edge e is multiplied by $\gamma(e)$. Let $f(e) \geq 0$ be the amount of flow f on edge e, and $\mu(e) \geq 0$ be the capacity of edge e. The capacity constraint $f(e) \leq \mu(e)$ must hold for every edge e. The goal of the problem is to send a flow emanating from the source into the destination to the greatest extent possible, subject to the capacity constraints. Let *generalized network* $G = (V, E, s, t, \mu, \gamma)$ be information network (V, E) with the source $s \in V$, the destination $t \in V$, the capacity μ, and the gain γ. Figure 8 depicts an example of a generalized max-flow. 0.4 units and 0.2 units of the flow arrive at "USA" along path (A) and path (B), respectively. As illustrated in Fig. 8, a large amount of flow is usually sent along paths which are short and are formed by edges having high gains.

To use edges of both directions, Zhang et al. [2] construct a *doubled network*, as discussed in Sect. 4.2. The reversed edge e_{rev} for every edge e in G is assigned with $\mu(e_{rev}) = \mu(e)$ and $\gamma(e_{rev}) = rev(e) = \lambda \times \gamma(e), 0 \leq \lambda \leq 1$, as depicted in Fig. 6 (B). Also, a new constraint $f(e)f(e_{rev}) = 0$ for every edge e is introduced to satisfy the capacity constraint on the doubled network. To assign gain for edges, Zhang et al. [2] use the edge weight function introduced in Sect. 4.2.

The model reflects all the three concepts important for measuring a relationship: distance, connectivity and co-citation. Zhang et al. [2] ascertained the model can measure the strength of relationships in Wikipedia more correctly than previous methods [1], [3], [4] can.

4.4 A Generalized Flow Based Method

We propose a generalized flow based method to mine disjoint paths important for a relationship from concept s to concept t in Wikipedia. We use a new technique for avoiding "confluences" of a generalized max-flow by setting vertex capacities.

We first present the method as follows.

(1) Construct a generalized network $G = (V, E, s, t, \mu, \gamma)$ using pages and links within at most m hop links from s or t in Wikipedia.

(2) Construct the doubled network G' for G, determine edge gain γ using the edge weight function discussed in Sect. 4.2, set capacity $\mu = 1$ for every edge, and set vertex capacities discussed later.

(3) Compute a generalized max-flow f emanating from s into t on G'.

(4) Decompose the flow f into flows on a set P of paths. Let $df(p_i)$ denote the value of flow on a path p_i, for $i = 1, 2, \ldots, |P|$. For example, the flow on the network depicted in Fig. 8 is decomposed into flows on two paths (A) and (B). The value of the decomposed flow on path (A) is 0.4; that on path (B) is 0.2.

(5) Output the largest-k paths in decreasing order of $df(p_i)$.

(6) For each edge (u, v) in the largest-k paths, extracts an explanatory snippet, i.e., text surrounding the anchor text of link v on page u, using a KWIC concordance tool [21].

As explained in Sect. 4.3, in the generalized max-flow model, a large amount of flow is usually sent along paths which are short and are formed by edges having high gains, such as, in Fig. 8, the flow is sent along path (A) and path (B), and the value of the flow on path (A) is larger than that on path (B). Consequently, the largest-k paths that hold high values of flows usually are short and are formed by edges having high gains. It is observed that most of the edges assigned with high gains are important in constituting an relationship, as discussed in Sect. 4.2. Therefore, the largest-k paths usually are short and are formed by important edges.

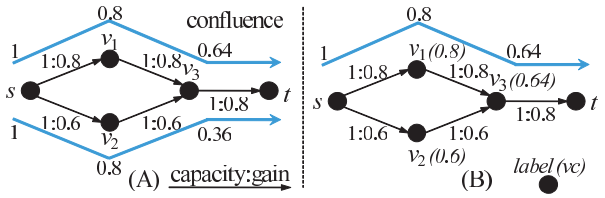

Fig. 9 The flow confluence problem.

Generally, short paths are more important for a relationship than long paths, as discussed in [1], [9]–[11]. Also, there is a high possibility that paths formed by important edges are important. Therefore, we claim that a high possibility exists that the largest-k paths mined by the generalized flow based method are important to a relationship. We ascertain the claim through experiments described in Sect. 5.2.2.

We next discuss the new technique of setting vertex capacity. The generalized max-flow problem is a natural extension of the classical max-flow problem whose flow is always sent along disjoint paths. A problem arises, however, which is attributable to the gain: a flow can be confluent at a vertex except s and t. For example, Fig. 9 (A) depicts a confluence of flow at vertex v_3; the amount of the flow sent along (v_1, v_3) becomes 0.64 at v_3. That along (v_2, v_3) becomes 0.36. The flow can be confluent at v_3 and can be sent along (v_3, t). If a generalized max-flow is confluent at many vertices, then the paths composing the flow become dependent paths. Consequently, the largest-k paths obtained in the step (5) might contain some dependent paths. One idea to solve the problem is to introduce a constraint that a flow must be sent along vertex disjoint paths. Unfortunately, no polynomial-time algorithm exists, to the best of our knowledge, to solve the generalized max-flow with the constraint.

We propose an approach to prevent a flow from being confluent to the greatest extent possible. Concretely, we set the capacity of every edge to one and set the capacity of every vertex v, except s and t, to

$$\mu'(v) = \max_{p \in P} \prod_{e \in p} \gamma(e),$$

the maximum production of the gains of the edges in a path, where P is the set of all paths from s to v. The vertex capacity of s or t is set to ∞. The capacities of all the vertices can be computed easily by solving a single source shortest path problem setting the length of edge e to $-\log(\gamma(e))$. Because the capacity of every edge is setting to 1, the largest value of the flow could be sent to node v along a single path going from s to v is $\mu'(v)$. Therefore, by setting the capacity of vertex v to $\mu'(v)$, most of the time, we could prevent a flow to be sent to v along more than one paths. For example, Fig. 9 (B) depicts the vertex capacity function for the generalized network depicted in Fig. 9 (A). Although a flow is confluent at vertex v_3 in Fig. 9 (A), the confluence does not happen in Fig. 9 (B) because of the vertex capacity. We examine how effective the vertex capacity function is using experiments discussed in Sect. 5.2.

4.5 Theoretical Analysis

In this section we compare the naive method with the generalized flow based method through theoretical analysis.

First, let us consider the qualities of the largest-k paths mined by each method. The naive method always considers popular concepts having high degrees are unimportant, and degrade paths containing popular concepts. For example, in Fig 5, the naive method underestimates path (B) because "Bush" is a popular concept. However, in Wikipedia, pages of famous people, places or events, are linked from and linking to many other pages. Consequently, many popular concepts existing on the Wikipedia information network represent famous people, places, or events. Such popular concepts might be as important as other concepts to some relationships, such as "Bush" depicted in Fig 5. The degree of a concept is essentially independent of its importance to a relationship. Therefore, estimating the importance of paths according to the degrees of the concepts in the paths would be inappropriate.

On the other hand, the generalized flow based method neither overestimates nor underestimates popular concepts, and estimates the importance of a path according to the importance of edges in the path. The importance of edges is determined using the edge weight function introduced in Sect. 4.2.

Second, we consider the quantities of the disjoint paths could be mined by each method. As discussed in Sect. 2, disjoint paths are easy to understand and do not contain redundant information. The naive method outputs paths including dependent paths and independent paths which have high weights as the largest-k paths. On the other hand, as discussed in Sect. 4.4, a flow is usually sent along disjoint paths on a generalized network after setting the vertex capacity. The generalized flow based method outputs the paths along which a large amount of flow is sent as the largest-k paths. Therefore, the largest-k paths obtained by the generalized flow based method are almost all disjoint paths.

We confirm our analysis described above through experiments, in Sect. 5.

4.6 Algorithm

We implement the naive method using the implementation of Yen's ranking K shortest loopless paths algorithm [24], according to the algorithms of CFEC described in [1]. Given a network $G = (V, E)$, the implementation [24] presents $O(K|V|(|E| + |V|log|V|))$ computational complexity order.

We use the rounded primal-dual algorithm [22] to compute an approximately maximum generalized flow for the generalized flow based method. For given approximation parameter $0 < \alpha < 1$, the algorithm outputs a generalized flow on a network $G = (V, E)$ whose value is at least as much α times as the value of a generalized maximum flow. The algorithm iteratively computes maximum flows on G for at most $O\left(|V|^2(1-\alpha)^{-1}\log_2 B\right)$ times, each itera-

Country	Elucidatory Objects
Japan	Oil crisis, Niigata, Nihon Shoki, Kyushu Oil Co., Ltd., added-profit trade, Crude oil, Nippon Oil Corp., Japanese post-war economic miracle, ⋯
Saudi Arabia	Ghawar Field, OAPEC, Crude oil, Oil field, Price of petroleum, Oil-producing Country, Rub' al Khali, Arabian Oil Company, OPEC, ...
Kuwait	Burgan Field, OAPEC, Crude oil, Oil field, Oil-producing Country, OPEC, Asphalt, Gulf War, Middle East War, ⋯

Fig. 10 Elucidatory concepts for the relationships from petroleum to each country.

Group	Countries	Label
0	Saudi Arabia, Kuwait, Iran, Bahrain, Libya	Oil crisis, OAPEC, Oil-producing country, Middle East, Oil field, Price of petroleum, Saudi Aramco, Iran?Iraq War, Asphalt
1	Japan, USA, Russia, China, UK	Crude oil, Middle East, Asphalt, Oil field, Iraq, Iran, Price of petroleum, North Sea oil, Sudan

Fig. 11 Classification for the relationships between petroleum and countries.

tion requires $O\left(|V|^2\sqrt{|E|}\right)$ time, where $\log_2 B$ is the largest number of bits to store each capacity and gain. Totally, in $O\left(|V|^4\sqrt{|E|}(1-\alpha)^{-1}\log_2 B\right)$ time the algorithm computes a maximum generalized flow. We observe that the largest-100 paths for a relationship can be obtained at the first 3–10 iterations of the computation of maximum flows, in the experiments described in Sect. 5. Therefore, we stop the iterations when we could obtain the largest-k paths instead of computing a maximum generalized flow completely. Finally, our implementation for the generalized flow based method computes the largest-100 paths in $O\left(|V|^2\sqrt{|E|}\right)$.

4.7 Classification for Relationships

In this section, given a set of relationships between a common source concept and different destination concepts, we apply our method for mining paths to classify the destination concepts in the relationships. For example, given a set of relationships between petroleum and countries, we classify the countries into groups. We first mine the largest-k paths for each relationship, say $k = 50$. We define *elucidatory concepts* for a relationship as the concepts in the paths, except the source and destination. Intuitively, similar relationships share many common elucidatory concepts. For example, Fig. 10 presents some elucidatory concepts for Japan, Saudi Arabia, and Kuwait. Saudi Arabia and Kuwait, which are both oil-producing countries in the Middle East, share many common elucidatory concepts. On the other hand, Japan shares almost no elucidatory concepts with them.

We apply a frequent itemsets based clustering method named FIHC [25] to our classification. In fact, FIHC is used to classify documents using sets of words appearing together in many documents. Using elucidatory concepts instead of words, we can obtain clusters of relationships. Every cluster is also assigned a label which is a set of elucidatory concepts shared by every relationship in the cluster. In some cases, the clusters obtained by FIHC could be too numerous for a user to understand. Our classification method unifies them into fewer groups in response to a user's request. We first divide a set of the clusters into exactly two sets of clusters. Then, recursively divide one of the sets into two sets, until we obtain the desired number of sets. Each set is regarded as a group of destinations. For dividing a set, we first construct a feature vector for each destination in the set using its frequent items. Let the cluster whose label has items more than that of any other cluster be the *center cluster*. We then compute the similarity $sim(i)$ between the center cluster and each cluster i in the set by the group average method using the feature vector. We define the distance between two sets S_1 and S_2 based on the similarities, $dist(S_1, S_2)$, as $\min_{i\in S_1,\ j\in S_2}|sim(i) - sim(j)|$. We finally determine which one of two sets S_1 and S_2 every cluster belongs to so that the distance $dist(S_1, S_2)$ is maximized. As an example, our method classifies the relationships from petroleum to the top-10 countries strongly related to petroleum into two groups. Figure 11 presents the groups of the countries. By investigating the labels of groups, a user could understand that group 0 and group 1 respectively correspond to "petroleum exporting countries" and "petroleum consuming countries."

5. Experiments and Evaluation

5.1 Dataset and Environment

We perform experiments on a Japanese Wikipedia dataset (2009/05/13 snapshot). We first extract 27,380,916 links that appeared in all pages. We then remove pages that are not corresponding to concepts, such as each day, month, category, person list, and portal. We also remove links to such pages, and obtain 11,504,720 remaining links.

We implemented our program in Java and performed experiments on a PC with four 3.0 GHz CPUs (Xeon), 64 GB of RAM, and a 64-bit operating system (Windows Vista).

5.2 Evaluation of Mined Paths

In this section, we first investigate whether paths mined by our methods are actually important for a relationship. We then examine how many of the mined paths are disjoint.

Let the following six symbols represent our methods below. (o) is the naive method explained in Sect. 4.1. (e), (d), (de) are the naive methods using improvements described in Sect. 4.2: (e) the edge weight function, (d) the doubled network, and (de) the both ones. (g,w/vc) and (g,wo/vc) are the generalized flow based method with the vertex capacity and that without setting the vertex capacity proposed in Sect. 4.4, respectively. We select 200 relationships between two concepts of 16 types: two politicians, two countries, a politician and a country, petroleum and a country, Buddhism and a country, two countries' cuisines, global warming and an industry, headache and a domain of medicine, Apple Inc. and a company, Earthquake and a landform, Japanese and a race, informatics and an academic field, Greenhouse gas and a country, Olympic Games and a country, Human and an animal, Cao Cao and a people in

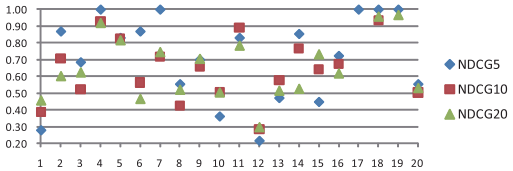

Fig. 12 *NDCG*s of the result pages searched for each relationship.

Chinese history. We evaluate the paths mined for 20 relationships of the 16 types by human subject in Sect. 5.2.2. We select the 16 types because the concepts of the 16 types are familiar to the participants joining the human subjects. The human subjects requires the participants to read a lot of text information related to the concepts. It is hard for them to understand the text information, if they are unfamiliar with the concepts. To mine paths important for a relationship between s and t, we construct a network G using pages and links within at most three hop links from s or t in Wikipedia, for all methods. Careful observation of Wikipedia pages revealed that several 3 hops paths are useful for understanding a relationship, such as the paths depicted in Fig. 1. However, we were able to find few useful 4 hops paths. In preliminary experiments, we also find that paths formed by four links seldom appear in the largest-k paths mined on G constructed using four hop links.

5.2.1 Preliminary Experiment for Web Search Engines

In this section, we quantitatively evaluate how many result pages searched using a query "s t" contain information about the relationship between s and t. We first obtain Google's Top-20 pages using query "s t" for each of the 20 relationships between s and t presented in Table 1. We then use Normalized Discounted Cumulative Gain (NDCG) to measure the top-n pages described at below:

$$NDCG_n = NF_n \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{log_2(1+i)},$$

where $rel_i = 1$ if the i-th page contains any information about the relationship, otherwise $rel_i = 0$, and NF_n is a normalization factor calculated to make $NDCG_n$ equal to 1 when all $rel_i = 1, 1 \leq i \leq n$. $NDCG_n$ varies from 0 to 1. $NDCG_n$ of a ranking of web pages becomes larger if more pages in the ranking contain information about the relationship, and those pages are ranked high.

Figure 12 presents the $NDCG_5$, $NDCG_{10}$, and $NDCG_{20}$ of the result pages for each of the 20 relationships. The indices in the x-axis correspond to the indices of the 20 relationships presented in Table 1. The *NDCG*s vary greatly from 0.21 to 1.00 among the relationships. The average $NDCG_5$, $NDCG_{10}$, and $NDCG_{20}$ for all the 20 relationships are 0.71, 0.68, and 0.66, respectively. On average, 64.5% of the top-20 pages searched for a relationship contain some information about the relationship, without considering the quality and the quantity of the information. That is, averagely, 66% of the top-20 pages searched for a relationship contains information about the relationship. However, Fig. 12 is rather inflated because neither the quality nor the quantity of the information is considered. In fact, we found that some of the top-20 pages contain many duplicate information, and some pages are too long to read. For example, 6 pages searched for the 9th relationship between "George W. Bush" and "Junichiro Koizumi" contain duplicate information; 4 pages searched for the 17th relationship between "Greenhouse Gas" and "German" contain about 70 thousands characters. Therefore, it is too hard to read all the top-20 pages for a relationship and to organize knowledge about the relationship from the top-20 pages manually. Moreover, as discussed in Sect. 1, Web search engines are unable to search information about implicit relationships which is described in multiple pages, such as the one depicted in Fig. 1. Therefore, we propose methods to mine important paths explaining an implicit relationship in Wikipedia.

5.2.2 Path Importance

In this section, we evaluate the importance of the mined paths obtained by every method by human subjects. To the best of our knowledge, there is no benchmark dataset for evaluating the importance of paths for a relationship. We discussed methods [1], [9], [11] which extract a subgraph for explaining a relationship in Sect. 3. The importance or goodness of the subgraphs extracted by these methods are only evaluated by case studies in [1], [9], [11].

We first randomly select 20 from the 200 relationships (one or two from each of the 16 types) explained above. For each relationship, we mine a set of the largest-20 paths by each of our six methods. Let P be the union of these six sets. On average, P contains 50-60 paths, because the sets mined by different methods usually overlap. We then ask 10 participants to evaluate every path p in P and every edge $e(u, v)$ in p. Totally, every participant evaluates 1072 paths and 2714 edges. To each edge $e(u, v)$, every tester assigns an integer score 0, 1, or 2 representing the strength of the explicit relationship between u and v, by reading the explanatory snippet of $e(u, v)$. A higher score was assigned to a stronger explicit relationship. To each path p, every tester assigns an integer score 0, 1, or 2 to judge whether an implicit relationship between the source and the destination could be established through the connection of the explicit relationships represented by the edges in p. A higher score was assigned to a path representing a stronger implicit relationship. Participants are not allowed to evaluate paths and edges using their own knowledge about the concepts appearing in the paths. Instead, explanatory snippets extracted for each edge from Wikipedia are provided for the evaluation. We then compute the average score of every edge and every path for each relationship.

We present some examples of the assignments here. All the testers assigned score 2 to the top path and the two edges in the path depicted in Fig. 1. For the relationship from petroleum to Saudi Arabia, path (*Petroleum, Burgan Field, Saudi Arabia*) is mined. The snippet of edge (*Bur-*

gan Field, Saudi Arabia) is "Burgan Field in Kuwait is still one of the world's easiest production sites now, which differs from the Ghawar Field in Saudi Arabia." Most testers assigned score 0 to the path and the edge. Let us consider another path (*George W. Bush, Yasuo Fukuda, Junichiro Koizumi*). Each of "Yasuo Fukuda" and "Junichiro Koizumi" was a prime minister of Japan during the tenure of "George W. Bush" as the president of the USA. Most testers think that both the two edges in the path represent strong explicit relationships. However, they think that "Yasuo Fukuda" is unimportant in the relationship. Consequently, they assigned score 0 to the path.

Our methods output largest-k important paths ranked in a decreasing order of importance for a relationship. We use Normalized Discounted Cumulative Gain (NDCG) to measure the importance of the largest-k paths based on their ranking. Let $0 \leq score_i \leq 2$ denote the average of the scores given by the participants for the path ranked ith. The $NDCG_n$ at rank position n is computed as follows:

$$NDCG_n = NF_n \sum_{i=1}^{n} \frac{2^{score_i} - 1}{log_2(1+i)},$$

where NF_n is a normalization factor calculated to make $NDCG_n$ equal to 1 when all $score_i = 2, 1 \leq i \leq n$. $NDCG_n$ varies from 0 to 1. $NDCG_n$ of a ranking of paths becomes larger if more paths in the ranking having large *score*, and the paths having large *score* are ranked high. Similarly, we compute $NDCG_n$ for the edges in the largest-n paths.

Table 1 presents the *NDCG*s of paths obtained by every of the six methods (g,w/vc), (g,wo/vc), (d), (de), (o), and (e) for the 20 relationships. The shaded cells emphasize the maximum *NDCG* of each relationship. The generalized flow based method with vertex capacity (g,w/vc) yields most of the highest *NDCG*s. The *NDCG*s obtained by the methods without the double network, (o) and (e), are significantly low for some relationships, such as the relationship between petroleum and the USA. Using only the original directions of edges, few paths formed by edges in the direction from the source to the destination only, exist in the network. However, several important paths containing edges in the inverse direction form the destination to the source, are mined by using the doubled network.

Figure 13 presents the average *NDCG*s of the edges of all the 20 relationships obtained by each method. Similarly, Fig. 14 presents the average *NDCG*s of the paths of the 20 relationships. For both paths and edges, methods (g,w/vc) and (g,wo/vc) produce the highest and the second highest average *NDCG*s, respectively. All methods yield high average *NDCG*s of edges, they can mine many edges representing strong explicit relationships. However, such edges do not necessarily constitute a path important for a relationship. For example, the path (*George W. Bush, Yasuo Fukuda, Junichiro Koizumi*) discussed above is constituted by such edges. Most participants think that the path is unimportant because "Yasuo Fukuda" is unimportant in the relationship between "George W. Bush" and "Junichiro

Table 1 *NDCG*s of paths.

method	$NDCG_5$	$NDCG_{10}$	$NDCG_{20}$	$NDCG_5$	$NDCG_{10}$	$NDCG_{20}$
	1. Japan – Russia			11. Global warming – Agriculture		
g,w/vc	0.88	0.85	0.73	0.76	0.67	0.62
g,wo/vc	0.69	0.66	0.49	0.73	0.70	0.61
d	0.76	0.59	0.54	0.64	0.59	0.58
de	0.85	0.72	0.63	0.73	0.67	0.61
o	0.55	0.52	0.45	0.63	0.56	0.52
e	0.70	0.60	0.51	0.65	0.60	0.57
	2. Japan – China			12. Headache – Internal medicine		
g,w/vc	0.73	0.66	0.59	0.48	0.51	0.51
g,wo/vc	0.66	0.56	0.43	0.32	0.35	0.39
d	0.59	0.49	0.41	0.49	0.43	0.36
de	0.70	0.52	0.45	0.49	0.43	0.38
o	0.57	0.56	0.55	0.44	0.48	0.44
e	0.62	0.59	0.61	0.45	0.45	0.47
	3. Petroleum – USA			13. Apple Inc. – IBM		
g,w/vc	0.89	0.74	0.65	0.75	0.59	0.57
g,wo/vc	0.77	0.61	0.56	0.68	0.64	0.61
d	0.33	0.29	0.35	0.64	0.59	0.57
de	0.69	0.54	0.47	0.85	0.78	0.65
o	0.00	0.00	0.00	0.61	0.66	0.61
e	0.00	0.00	0.00	0.69	0.71	0.65
	4. Petroleum – Saudi Arabia			14. Earthquake – Volcano		
g,w/vc	0.87	0.84	0.75	0.79	0.71	0.61
g,wo/vc	0.80	0.73	0.61	0.78	0.64	0.57
d	0.80	0.68	0.54	0.55	0.53	0.52
de	0.80	0.71	0.58	0.69	0.62	0.55
o	0.83	0.60	0.55	0.56	0.67	0.55
e	0.83	0.71	0.56	0.75	0.64	0.61
	5. Buddhism – Sri Lanka			15. Japanese – Chinese		
g,w/vc	0.98	0.84	0.77	0.32	0.34	0.33
g,wo/vc	0.86	0.76	0.65	0.32	0.34	0.33
d	0.81	0.80	0.73	0.19	0.23	0.23
de	0.81	0.80	0.73	0.22	0.25	0.24
o	0.93	0.84	0.77	0.40	0.36	0.29
e	0.93	0.86	0.77	0.48	0.38	0.30
	6. Japanese cuisine – Chinese cuisine			16. Informatics – Mathematics		
g,w/vc	0.77	0.65	0.62	0.54	0.50	0.43
g,wo/vc	0.78	0.68	0.50	0.53	0.41	0.39
d	0.53	0.35	0.23	0.47	0.43	0.42
de	0.56	0.38	0.28	0.53	0.46	0.44
o	0.60	0.58	0.51	0.50	0.46	0.42
e	0.60	0.63	0.57	0.69	0.57	0.54
	7. Yoshiro Mori – China			17. Greenhouse gas – German		
g,w/vc	0.30	0.34	0.34	0.67	0.63	0.56
g,wo/vc	0.29	0.25	0.27	0.49	0.48	0.50
d	0.49	0.44	0.33	0.60	0.58	0.58
de	0.50	0.44	0.34	0.60	0.58	0.62
o	0.29	0.23	0.20	0.23	0.23	0.21
e	0.32	0.26	0.21	0.23	0.23	0.21
	8. Yasuo Fukuda – USA			18. Olympic Games – Greece		
g,w/vc	0.49	0.46	0.30	0.83	0.77	0.64
g,wo/vc	0.64	0.47	0.33	0.83	0.77	0.64
d	0.15	0.23	0.23	0.77	0.63	0.47
de	0.16	0.29	0.26	0.82	0.64	0.47
o	0.00	0.00	0.00	0.81	0.73	0.60
e	0.00	0.00	0.00	0.81	0.71	0.57
	9. George W. Bush – Junichiro Koizumi			19. Human – Common chimpanzee		
g,w/vc	0.75	0.59	0.51	0.30	0.40	0.44
g,wo/vc	0.62	0.54	0.43	0.39	0.32	0.37
d	0.57	0.47	0.39	0.50	0.43	0.40
de	0.57	0.49	0.41	0.44	0.41	0.39
o	0.63	0.52	0.45	0.47	0.53	0.48
e	0.72	0.52	0.46	0.49	0.51	0.49
	10. Ichiro Ozawa – Taro Aso			20. Cao Cao – Zhuge Liang		
g,w/vc	0.28	0.27	0.27	0.52	0.50	0.46
g,wo/vc	0.31	0.30	0.26	0.58	0.53	0.47
d	0.42	0.31	0.28	0.38	0.37	0.32
de	0.41	0.32	0.27	0.38	0.35	0.29
o	0.33	0.25	0.24	0.14	0.15	0.29
e	0.27	0.28	0.25	0.14	0.29	0.31

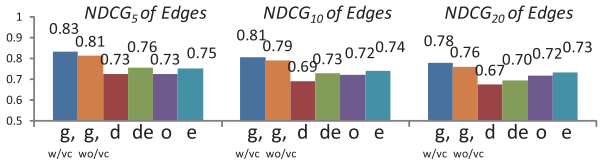

Fig. 13 Average *NDCG*s of edges of all the 20 relationships.

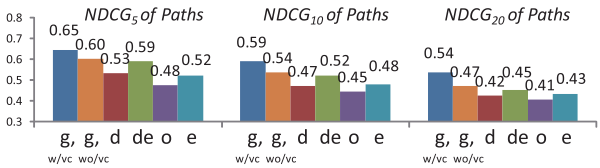

Fig. 14 Average *NDCG*s of paths of all the 20 relationships.

Koizumi." With respect to paths, method (g,w/vc) produces significantly higher average *NDCG*s than the other methods.

The vertex capacity improves the average *NDCG*s of the generalized flow based method, although the vertex capacity is originally proposed to mine more disjoint paths. As discussed later in Sect. 5.2.3, method (g,w/vc) with the vertex capacity mines more disjoint paths than method (g,wo/vc). k disjoint paths include more information than k dependent paths containing duplicate concepts do. Consequently, disjoint paths produce higher *NDCG*s than dependent paths, in most cases. The experimental results are similar to that removing redundancies in search results improves recall and precision in information retrieval system [5]–[7].

With respect to paths, the methods using the doubled network, (d) and (de), produce higher *NDCG*s than methods without using the doubled network, (o) and (e), respectively. Without using the doubled network, the naive method mines paths formed by edges in the direction from the source to the destination only. By using the doubled network, the naive method can mine paths formed by edges of any direction, and estimate paths formed by edges of different directions to be relatively less important, as discussed in Sect. 4.2. Therefore, the doubled network improves *NDCG* for the naive method. The methods without the edge weight function, (o) and (d), produce lower *NDCG*s for paths than those using the edge weight, (e) and (de), respectively.

As a result of the above discussion, we conclude that the generalized flow based method with the vertex capacity is the best for mining many paths important for relationships, and that setting vertex capacity, the doubled network, and the edge weight function are effective.

The generalized flow based method with the vertex capacity and the naive method took 1063 seconds and 1217 seconds, respectively, to mine largest-100 paths for the selected 200 relationships. We focus on the accuracy problem in this paper. As one of our future work, we plan to implement the generalized flow based method using parallel processing to accelerate computation. We also plan to construct a relationship search system to visualize the largest-10 paths for a given relationship.

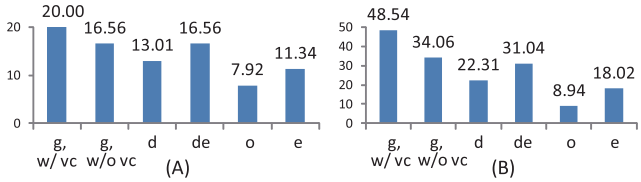

Fig. 15 Number of disjoint paths mined by each method.

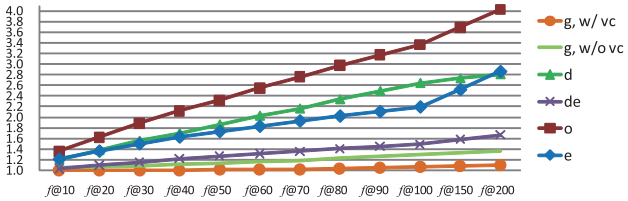

Fig. 16 Average concept frequency in the largest-k paths of each method.

5.2.3 Evaluation of Disjoint Paths

We have proposed a technique for avoiding confluences using the vertex capacity function in Sect. 4.4. We examine how many disjoint paths were mined by each method, and how effective the technique is. For the selected 200 relationships, we first mine the top 20 and the top 50 paths by each of our six methods and the generalized flow based method without the technique. We then count the disjoint paths in the mined paths for each relationship. Figure 15 (A) and Fig. 15 (B) depict the average number of disjoint paths in the largest-20 and that in the largest-50 paths, respectively. The symbol (g, w/ vc) denotes the generalized flow based method with the technique, and (g, w/o vc) denotes that without the technique. Method (g, w/ vc) produced the highest average number for both the largest-20 and largest-50 paths; especially, all the largest-20 paths are disjoint for all the 200 relationships. The naive methods without the doubled network, (o) and (e), produced the lowest average numbers. Consequently, we observed the following three facts: (1) Our technique is effective in mining disjoint paths. (2) The naive method is inadequate for mining disjoint paths. (3) The doubled network is effective in mining disjoint paths formed by edges of different directions.

As discussed in Sect. 2, we mine disjoint path to prevent a concept appearing multiple times in the mined paths. Therefore, we also evaluate how frequent a concept appears in the largest-k paths. We compute the average concept frequency $f@k$ in the largest-k paths important for a relationship between s and t mined by each method. Let O_k be the set of concepts in the largest-k paths, except s and t, and let $n_{k,j}$ denote how many times the j-th concept $o_{k,j} \in O_k$ appears in the largest-k paths. Then, the average concept frequency is defined as

$$f@k = \frac{\sum_j n_{k,j}}{|O_k|}.$$

Note that $f@k$ is at least 1. If $f@k = 1$, then every concept

Table 2 concepts in the paths for relationship between "Japanese cuisine" and "Chinese cuisine".

The generalized flow base method (g)
Karaage (269), Chili pepper (402), Soy milk (103), Sesame oil (95), Mochi (345), Dashi (305), Ginger (344), Donburi (119), Tonkatsu (215), Sashimi (477), Fried vegetables (87), Jiaozi (341), Jellied fish (45), Yatai (412), Chazuke (164), Kenchin soup (47), Western Cuisine (77), Crab stick (58), Japanese noodle (1038)
The naive method (de)
Nouvelle Chinois (12), Three major world cuisine (10), Seafood (12), Wynn Macau (13), Cooking School of West Japan (15), The Family Restaurant (15), Kazuhiko Cheng (31), Radisson Hotel Bangkok (21), Hotel Laforet Tokyo (17), Banyan Tree Bangkok (18), Jellied fish (45), West (Japanese restaurant chain) (19), Soup spoon (38), Grand Hyatt Fukuoka (20), Grand Hyatt Singapore (16), Ship dish person (20), Resort Okinawa Marriott & Spa (21), Hyatt Regency Osaka (22)

appears only once in the largest-k paths; if a number of concepts appear many times in the largest-k paths, then $f@k$ becomes larger than 1. Figure 16 illustrates the average value of $f@k$ in the largest-k paths mined by each method, for the selected 200 relationships. The method (g, w/ vc) has the lowest $f@k$ among all methods; especially, $f@100 = 1.06$ is almost equal to 1. That is, almost all concepts appear only once in the largest-100 paths mined by the method (g, w/ vc). The method without setting the vertex capacity (g, w/o vc) has higher $f@k$ than the method (g, w/ vc). The naive method without using the doubled network and the weight function, (o), produced the highest $f@k$; the values of $f@k$ increase dramatically as k increases. Consequently, we conclude that our technique of using the vertex capacity function is effective for avoiding redundant concepts in the mined paths. Many concepts appear frequently in the paths mined by the naive methods, although the doubled network is helpful for alleviating the redundancy issue.

5.2.4 Case Studies for Understanding Relationships

Table 2 presents the elucidatory concepts in the largest-20 paths important for the relationship between "Japanese cuisine" and "Chinese cuisine," mined by methods (g,w/vc) and (de), respectively. Both methods (g,w/vc) and (de) use the doubled network. The number in the parentheses behind each concept is the number of links going from or to the page representing the concept in Wikipedia. Each concept shown in Table 2 constitutes a mined path, e.g. "Karaage" constitutes (*Japanese cuisine, Karaage, Chinese cuisine*). Method (g,w/vc) mines many Japanese foods originated in China, such as karaage, mochi, fried vegetables, jiaozi, Japanese noodle, and soy milk. Method (g,w/vc) also mines some cooking ingredients used in both cuisines, such as chili pepper, sesame oil, and ginger. On the other hand, most elucidatory concepts mined by method (de) are hotels or restaurants purveying both cuisines. The elucidatory concepts in the mined paths for a relationship should play important roles in the relationship. In Japan or other Asian countries, it is common that hotels supply both Japanese and Chinese cuisines. These hotels and restaurants are not important parts constituting the relationship between Japanese cuisines and Chinese cuisines. As discussed in Sect. 3, random walk based methods always underestimate popular concepts; inversely, paths constituted by concepts having few links are always overestimated. As shown in Table 2, these hotels and restaurants have few links in Wikipedia. Therefore, the naive methods based on CFEC overestimate paths constituted by concepts corresponding to these pages. However, method (g,w/vc) mined many important concepts regardless of how many links the concepts have.

Path (*number of links*)	Brief Explanation	Score	Ranking	
			g,w/vc	de
Cao Cao, Liu Bei (*620*), **Zhuge Liang**	Zhuge Liang helped Liu Bei to fight Cao Cao during the Three Kingdoms period in China.	1.75	8	>100
Petroleum, Texas (*2620*), **USA**	Texas is a U.S. state whose economy has been heavily dependent on petroleum.	1.8	9	36
Apple Inc., Power Macintosh (*157*), **IBM**	Power Macintosh was a line of personal computers from Apple Inc. whose CPU is developed by Apple Inc., IBM and Motorola.	2.0	2	33
Chinese, Japanese orphans in China (*141*), **Japanese**	Japanese orphans in China consist of children left behind by Japanese families repatriating to Japan in the aftermath of World War II.	2.0	6	40
Japan, Russo-Japanese War (*2062*), **Russia**	The Russo-Japanese War (8 Feb. 1904 – 5 Sep. 1905) grew out of rival imperial ambitions of the Russian Empire and Japanese Empire.	2.0	7	42

Fig. 17 Five paths with rankings obtained by (g,w/vc) and (de).

Group	Industries	Label
0	Retailing, Traffic, information and communication industry , Service, Finance industry	Convenience store, Vending machine, Emissions Trading, Transportation, Niigata, Industry, Senko Co.,Ltd., Shima Spain Village
1	Construction industry, Manufacturing	Fuel cell, Biofuel, Carbon footprint, Convenience store, Emissions Trading, Industrial evolution, USA
2	Agriculture, Forestry, Fishing industry	Afforestation, Local production for local consumption, Biodiesel, Biomass, Biofuel

Fig. 18 Classification for relationships between CO_2 and industries.

We present five paths which are underestimated by the naive method (de), in Fig. 17. The column "Score" represents the average scores given by the participants for each path. Every one of the five paths is judged as important path for its relationship by the participants. By reading the "Brief Explanation," we could understand the importance of every path. The generalized flow based method (g,w/vc) correctly ranks the five paths high. By contrast, method (de) inappropriately ranks the five paths very low just because elucidatory concepts in the five paths have many links. The number in the parenthesis following each elucidatory concept represents its number of links. Again, we verified that the naive method always inappropriately degrades paths containing popular concepts.

From the above case studies, we ascertain that the generalized flow based method is more appropriate than the naive methods for mining paths important to a relationship in Wikipedia.

5.3 Case Study: Classification for Relationships

We present an example of our classification for relationships from carbon dioxide, CO_2, to the top-10 industries strongly related to CO_2. Our method discussed in Sect. 4.7 then classifies the 10 industries into three groups. Figure 18 presents

the groups and the label for each group. By investigating the groups and the labels, a user could understand the classification. In fact, the groups 0, 1, and 2 respectively correspond to the tertiary sector, the secondary sector, and the primary sector of the economy. The label for group 2 includes "Afforestation," "Local production for local consumption," "Biodiesel," "Biomass," and "Biofuel," which are approaches performed in "Agriculture," "Forestry," or "Fishing industry," for decreasing CO_2 emissions. The label for group 0 also contains concepts related to CO_2 emitted by the industries in the group. For example, "Shima Spain Village" is a famous amusement park in Japan, and "Senko Co. Ltd." is a Japanese Logistics company, both of which use renewable energy; "Niigata" is one of the top three cities having high CO_2 emissions per capita in the transportation industry of Japan. Similarly, the label for group 1 is helpful for understanding the relationships between CO_2 and the industries in group 1. Consequently, we confirmed that our classification method could give a user a better understanding of relationships.

6. Conclusion

We proposed a new approach for explaining a relationship between two concepts by mining disjoint paths connecting the concepts on Wikipedia. We achieved the approach by proposing the naive method and the generalized flow based method. Our experiments revealed that the generalized flow based method is more appropriate than the naive methods for mining paths important to a relationship. We confirmed that the proposed technique of setting vertex capacity is very effective in improving the generalized flow based method for mining many disjoint paths. We ascertained that our classification, proposed as an application of our approach, is also helpful for understanding relationships.

In the future, we plan to implement the generalized flow based method using parallel processing to accelerate computation. We also plan to apply the mined paths for a relationship to Web search of images and texts explaining the relationship.

Acknowledgements

This work was supported by Grant-in-Aid for Scientific Research(B) (20300036) and Grant-in-Aid for Young Scientists (B) (23700116).

References

[1] Y. Koren, S.C. North, and C. Volinsky, "Measuring and extracting proximity in networks," Proc. 12th ACM SIGKDD Conference, pp.245–255, 2006.
[2] X. Zhang, Y. Asano, and M. Yoshikawa, "A generalized flow based method for analysis of implicit relationships on wikipedia," IEEE Trans. Knowl. Data Eng., Nov. 2011. IEEE Computer Society Digital Library.
[3] K. Nakayama, T. Hara, and S. Nishio, "Wikipedia mining for an association web thesaurus construction," Proc. 8th WISE, pp.322–334, 2007.
[4] G. Jeh and J. Widom, "Simrank: A measure of structural-context similarity," Proc. 8th ACM SIGKDD Conference, pp.538–543, 2002.
[5] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.Y. Ma, "Improving web search results using affinity graph," Proc. 28th SIGIR, pp.504–511, 2005.
[6] H. Chen and D.R. Karger, "Less is more: Probabilistic models for retrieving fewer relevant documents," Proc. 29th SIGIR, pp.429–436, 2006.
[7] C.L. Clarke, M. Kolla, G.V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," Proc. 31st SIGIR, pp.659–666, 2008.
[8] X. Zhang, Y. Asano, and M. Yoshikawa, "Mining and explaining relationships in Wikipedia," Proc. 21st DEXA, Part II, pp.1–16, 2010.
[9] C. Faloutsos, K.S. McCurley, and A. Tomkins, "Fast discovery of connection subgraphs," Proc. 10th ACM SIGKDD Conference, pp.118–127, 2004.
[10] H. Tong and C. Faloutsos, "Center-piece subgraphs: Problem definition and fast solutions," Proc. 12th ACM SIGKDD Conference, pp.404–413, 2006.
[11] J. Cheng, Y. Ke, W. Ng, and J.X. Yu, "Context-aware object connection discovery in large graphs," Proc. 25th ICDE, pp.856–867, 2009.
[12] P.G. Doyle and J.L. Snell, Random Walks and Electric Networks, Mathematical Association America, New York, 1984.
[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a web of open data," in The Semantic Web, ed. K. Aberer, K.S. Choi, N. Noy, D. Allemang, K.I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudre-Mauroux, pp.722–735, Springer Berlin / Heidelberg, 2007.
[14] J. Lehmann, J. Schüppel, and S. Auer, "Discovering unknown connections - the DBpedia relationship finder," Proc. 1st Conference on Social Semantic Web, pp.99–110, 2007.
[15] G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum, "Naga: Searching and ranking knowledge," Proc. 24th ICDE, pp.953–962, 2008.
[16] F.M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," Proc. 16th WWW, pp.697–706, 2007.
[17] K. Anyanwu, A. Maduko, and A.P. Sheth, "Semrank: ranking complex relationship search results on the semantic web," Proc. 14th WWW, pp.117–127, 2005.
[18] B. Aleman-Meza, C. Halaschek-Wiener, I.B. Arpinar, and A.P. Sheth, "Context-aware semantic association ranking," Proc. 1st SWDB, pp.33–50, 2003.
[19] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," Proc. 18th WWW, pp.101–110, 2009.
[20] SPYSEE. http://spysee.jp/
[21] C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, 1999.
[22] K.D. Wayne, Generalized Maximum Flow Algorithm, Ph.D. Thesis, Cornell University, New York, U.S., Jan. 1999.
[23] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, New Jersey, 1993.
[24] E.Q.V. Martins and M.M.B. Pascoal, "A new implementation of yen's ranking loopless paths algorithm," 4OR: A Quarterly Journal of Operations Research, vol.1, pp.121–133, 2003. 10.1007/s10288-002-0010-2.
[25] B.C.M. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets," Proc. 3rd SDM, 2003.

Xinpeng Zhang received B.S. degree from the School of Software Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2004, the M.S. degree in Information Science from Graduate School of Informatics, Kyoto University, Japan, in 2009. He is currently a Ph.D. student at Graduate School of Informatics, Kyoto University. His research interests include graph data mining, information retrieval, and natural language process. He is a student member of the IEEE, IEEE Computer Society, and DBSJ.

Yasuhito Asano received B.S., M.S. and D.S. in Information Science, the University of Tokyo in 1998, 2000, and 2003, respectively. In 2003-2005, he was a research associate of Graduate School of Information Sciences, Tohoku University. In 2006-2007, he was an assistant professor of Department of Information Sciences, Tokyo Denki University. He joined Kyoto University in 2008, and he is currently an associate professor of Graduate School of Informatics. His research interests include web mining, network algorithms. He is a member of IEEE, IPSJ, DBSJ, OR Soc. Japan.

Masatoshi Yoshikawa received the B.E., M.E. and Ph.D. degrees from Department of Information Science, Kyoto University in 1980, 1982 and 1985, respectively. From 1985 to 1993, he was with Kyoto Sangyo University. In 1993, he joined Nara Institute of Science and Technology as an Associate Professor of Graduate School of Information Science. From April 1996 to January 1997, he has stayed at Department of Computer Science, University of Waterloo as a visiting associate professor. From June 2002 to March 2006, he served as a professor at Nagoya University. From April 2006, he has been a professor at Kyoto University. His current research interests include XML database, databases on the Web, and multimedia databases. He is a member of ACM and IEEE Computer Society and IPSJ.