

# Measuring the Similarity between TV Programs Using Semantic Relations

*Ichiro YAMADA Masaru MIYAZAKI Hideki SUMIYOSHI  
Atsushi MATSUI Hironori FURUMIYA Hideki TANAKA*

Japan Broadcasting Corporation, 1-10-11 Kinuta, Setagaya-ku, Tokyo, JAPAN

yamada.i-hy@nhk.or.jp, miyazaki.m-fk@nhk.or.jp,  
sumiyoshi.h-di@nhk.or.jp, matsui.a-hk@nhk.or.jp,  
furumiya.h-ke@nhk.or.jp, tanaka.h-ja@nhk.or.jp

## ABSTRACT

This paper presents a novel method of measuring the similarity between TV programs by using summaries of the Electronic Program Guide (EPG). Most previous methods use statistics such as the TFIDF based cosine measure of word vectors, whose elements are words appearing in the summaries. However, these approaches are not effective because TV program summaries, especially short ones, do not necessarily share many words even when they have similar meanings. The proposed method generates a graph structure whose nodes are TV programs and nouns. These nouns are connected by semantic relations that are extracted from the Web automatically. The similarity between two TV programs is measured in terms of the relativeness of two TV program's nodes in the graph structure by using a random walk algorithm. Experiments showed that our method is better at measuring similarities between two TV programs compared with baseline methods.

---

KEYWORDS : Measuring similarity, Semantic relation, Recommendation system, Graph structure, Random walk

---

# 1 Introduction

Japanese broadcasting stations have started services to provide viewers with previously broadcast TV programs on demand. Since these services are becoming popular, it is important for them to have an efficient way to find programs that a viewer wants to watch amidst huge program archives. Many on-demand services have the ability to present TV programs related to a selected program and rank them. This function makes it possible for viewers to find programs they would be interested in but would not have known about in advance. The TV program that the viewer selects from such a list is highly dependent on its presentation rank given by the on-demand service. FIGURE 1 shows how the number of selected programs depends on the presentation rank in a major Japanese on-demand service, NHK on demand. The higher ranked TV programs are selected more frequently than the lower ranked ones. For this reason, it is important to find out how to select related programs in huge program archives and how to rank them for better on-demand services. In order to select the related programs and rank them, NHK on-demand system measures the similarity between TV program summaries in Electronic Program Guides (EPGs) in advance. The system ranks TV programs according to their similarities to the selected program by a viewer. Since the viewer might be interested in watching TV programs which are relevant, but not exactly similar in content, the technique for measuring the similarity between programs can exclude exactly similar content and select only related programs and rank them<sup>1</sup>.

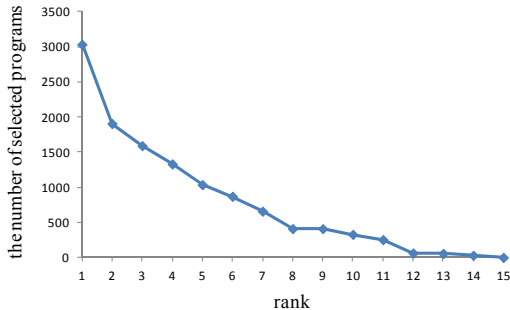


FIGURE 1 – The number of selected programs depends on the presentation rank in the NHK on demand service. (2010/9-2011/5)

NHK on-demand adopts a method of Goto et al. (2010) for measuring the similarity between TV program summaries. This method is based on word co-occurrences. Each summary is represented using words or n-grams of words as a vector, and the similarity between two summaries is calculated from their corresponding vectors. However, this method sometimes gives inappropriate similarity values for summaries (especially short ones) that have similar meanings but do not share many words. For example, the following sentences in program summaries were judged as dissimilar.

- (1) This TV program conveys aspects about the treatment of diabetes.
- (2) The doctor describes measures to alleviate hypertension, such as low-salt diet and drug therapy.

<sup>1</sup> A function for excluding exactly similar content has not been implemented on the NHK on-demand system yet.

If we knew that *low-salt diet* and *drug therapy* are *treatments* and *diabetes* and *hypertension* have the same hypernym *lifestyle-related diseases*, these sentences could be judged to have some semantic relevance.

This paper proposes a method for measuring such semantic relevance, that is, similarity between two TV program summaries by using semantic relations between nouns, such as causality and hyponymy. These semantic relations are extracted from the Web automatically and make it possible to judge the similarity of sentences appropriately even if they do not share many words. Our method generates a graph structure whose nodes are TV programs and nouns. These noun nodes are connected by the semantic relations. The similarity between two TV program summaries is calculated in terms of the relativeness of two TV program's nodes in the graph structure by using a random walk algorithm. Through experiments on ranking related TV programs provided by the NHK on demand service, we found that our method provides a proper similarity measure that shows a better correlation with human intuition than the baseline approaches we tested for comparison.

In the remainder of this paper, section 2 reviews related work on recommender systems. Section 3 introduces the methods of extracting semantic relations from Web data that we use for measuring the similarities between TV program summaries. Section 4 describes our method of measuring the similarities and ranking related TV programs. We present experimental results in section 5 before concluding the paper.

## 2 Related work

Many studies have been conducted on recommendation systems which offer relative products in relation to the user selected one. These systems are classified into content-based filtering and a user based approach known as collaborative filtering.

The content-based filtering is based on word co-occurrences in text relating to products, and it is commonly used in traditional information retrieval systems. Goto et al. (2010) proposed a method of TV program recommendation using content-based filtering. They used a score based on Okapi BM25 (Robertson 1999) and put weights on semantically significant words, such as named entities and compound words. Oku et al. (2007) proposed a context-aware recommendation system. Their system suggested products according to the user's context, such as time, weather, accompanying persons, and price, which change according to the situation. However, these methods rely on the assumption that more similar documents share more of the same words, even though it is true that similar pieces of content do not necessarily share the same words. Moreover, summaries in the EPG use a great variety of expressions; such summaries would not use the same word frequently to express a similar meaning.

Statistical models are useful for measuring the similarities between documents that do not share words but express similar meanings. Latent Semantic Indexing (LSI) represents terms in a latent semantic space by using the SVD of the corresponding term-document matrix (Deerwester et al. 1990). Hofmann et al. (1999) proposed PLSI which uses a latent variable model. However, these models are not good for measuring the similarities of TV program summaries, as determined in the experiment (Section 5).

Collaborative filtering is now used by several real-world recommendation systems. Amazon.com proposed item-to-item collaborative filtering (Linden et al. 2003). They produce

recommendations from customers who bought similar items to the ones in your shopping cart. Their algorithm works online and uses computation scales independent of the number of customers and number of items. Koren et al. (2009) proposed an approach based on matrix factorization; it characterizes both items and users by using vectors of factors inferred from an item rating pattern. Their system won the Netflix Prize, which was an open competition to predict user ratings for films. However, collaborative filtering suffers from the cold start problem wherein it cannot estimate the rating of new items and users, and recommendation systems for TV programs must handle new items (TV programs) frequently.

The proposed method is based on content-based filtering. We can put a high similarity value on documents that have similar meanings even if they do not share many words. These similarities between two items can be applied to the collaborative filtering approach by using not only the user's preference but similar items to the user's preference. Melville et al. (2010) proposed hybrid techniques which use both content-based and collaborative filtering. The proposed method can be used with such hybrid techniques, and it is expected to have promising results.

### 3 Acquisition of relations between nouns

This section describes three methods for acquiring semantic relations between nouns. The acquired relations are then used for measuring similarities between TV program summaries. We used the following four semantic relations.

- Hyponymy: A is a hypernym of B.  
Ex.) *lifestyle-related diseases / hypertension*
- Causality: B is caused by A.  
Ex.) *stroke / hypertension*
- Specialty: A is famous for B.  
Ex.) *Kyoto / temples*
- Material: A is made from B.  
Ex.) *beer / wheat*

We selected these relations because they are effective at capturing the VOD user's attention when it comes to TV program suggestions. For example, someone who is interested in *Kyoto* would probably like TV programs concerning *temples*.

We also use the entity-attribute-value relation. It can be considered that the entity word and the value word have the relation of attribute. For example, in the relation "*hypertension / management / weight loss*," the relation between *hypertension* and *weight loss* can be considered to be *management*.

In the following subsection, we describe the methods of acquiring these semantic relations.

### 3.1 Relation acquisition from Wikipedia

For the hyponymy relation and entity-attribute-value relation acquisition, we used an open-source software<sup>2</sup> based on the extraction methods of Sumida et al. (2008) and Yamada et al. (2010). Sumida et al. (2008) proposed a method of automatically acquiring hyponymy relations of nouns from Wikipedia. They focused on the hierarchical layout of articles in Wikipedia, which is made of titles, sections, sub-sections, itemizations, and so on. For example, in the article titled *lifestyle-related diseases*, there are itemizations *hypertension*, *diabetes*, and *history*. Relations such as the one between *lifestyle-related diseases* and *hypertension* and the one between *lifestyle-related diseases* and *diabetes* can be considered to be hyponymy relations, but the one between *lifestyle-related diseases* and *history* cannot be considered a hyponymy relation. Their method first extracts hyponymy relation candidates from the hierarchical structure of Wikipedia. The candidates are then classified into plausible and implausible ones by using a support vector machine (SVM) classifier.

Sumida et al. (2008) also proposed another method for hyponymy acquisition that exploits other information sources: the first sentence of Wikipedia articles, which is regarded as the article's definition, and category names. This method generates hyponymy relation candidates in which the hyponymy corresponds to the article titles and the hypernym comes from either of the information sources. The candidates are classified in the same process of analyzing a hierarchical layout.

We expanded their hyponymy acquisition method to generate entity-attribute-value relations (Yamada et al. 2010). We confirmed our assumption that if two words located in the layout structure can be regarded as a hyponymy relation, the article title, hypernym word and hyponym word can be interpreted as an entity, the attribute, and its attribute value independently. Take, for example, the hyponymy relation *management* and *weight loss* from the Wikipedia article *hypertension*; it can be interpreted that the entity *hypertension's management* (attribute) is *weight loss* (value).

### 3.2 Relation acquisition from Web text

Causality, specialty, and material relations are extracted by using a semantic relation acquisition service provided by the ALGIN forum<sup>3</sup> in Japan. This service is based on a method proposed by Stijn et al. (2009) and can extract large-scale relations between nouns from 6 million Japanese Web pages by inputting a small number of seed patterns. The service learns linguistic patterns that express each relation such as “X gives rise to Y” for causality with semantic word classes of X and Y acquired by large-scale clustering (Kazama et al. 2008). For example, if we know that the pattern “X gives rise to Y” expresses causality and the phrase “*hypertension* gives rise to *stroke*” appears frequently on the Web, the relation between *hypertension* and *stroke* will be regarded as causal. Moreover, *heart attack*, which belongs to the same class as *stroke*, can be also considered to have a causality relation with *hypertension*.

However, the relations acquired by this method include some obvious errors and ambiguities. For example, the method erroneously regards the relation between *disease* and *stroke* as being causal from the pattern “The *disease* gave rise to the *stroke*”. This is because the word *disease* belongs to the same class of *hypertension*. To avoid this error, we generated a stop-word list manually to

---

<sup>2</sup> <http://alaginrc.nict.go.jp/hyponymy/index.html>

<sup>3</sup> <http://alaginrc.nict.go.jp>

exclude these erroneous relations from the results of the semantic relation acquisition service.

### 3.3 Hypernym acquisition using compound nouns

A suffix of a compound noun sometimes becomes a hypernym of the original noun in a head-final language (Kuroda et al. 2009). For example, the suffix *disease* is considered to be the hypernym of *lifestyle-related diseases*. Since our target language in the experiment was Japanese, we first decomposed a compound noun into a sequence of nouns using a morphological analyzer and then checked whether the suffix sequence was a valid hypernym of the compound noun. We judged that the suffix is a valid hypernym if it is registered in a dictionary. In the experiment described in section 5, we used the Japanese WordNet (Bond et al. 2012) as the dictionary.

### 3.4 Acquired relations

We acquired hyponymy and entity-attribute-value relations by using the relation extraction software mentioned in section 3.1, targeting 5 years of Wikipedia dump data from 2007 to 2011. We also extracted causality, specialty, and material relations by using the semantic relation acquisition service mentioned in section 3.2, by inputting a few seed patterns, and acquired reliable patterns. We extracted hypernyms by using the suffixes of nouns appearing in the extracted relations and TV program summaries. TABLE 1 shows examples of the acquired relations.

We randomly sampled 200 of the automatically acquired relations respectively, and one of the authors checked whether the relations were valid or not. TABLE 2 shows the number of acquired

Word X	Relation	Word Y
hikkigu (writing material)	hyponymy	shapu-pen (mechanical pencil)
eiga (movie)	hyponymy	Star Trek
ice cream	hyponymy	vanilla ice cream
allergen	causality	kikanshi zensoku (asthma bronchiale )
El Nino	causality	ondo joshu (rising water temperatures)
Canada	specialty	winter sports
Chiang Mai	specialty	bukkyo-jiin (Buddhist temple)
miso	material	soybean
pannacotta	material	coconut
John Woo	directed film	Red Cliff
J. D. Salinger	work	A boy in France

TABLE 1 – Examples of acquired relations between nouns.

Relation	Number of relations	Accuracy
hyponymy(Wikipedia)	8,591,469	90.0%*
hyponymy(suffix)	1,347,382	82.5%
entity-attribute-value	5,213,455	94.0%*
causality	77,636	75.0%
specialty	183,093	49.0%
material	49,711	73.0%

TABLE 2 – Number of acquired relations and accuracies. \* indicates accuracies obtained from the original literature.

Relation	Coverage
All relations	72.8% (68,726/94,456)
Relations other than the hyponymy acquired using suffix information.	47.7% (45,042/94,456)

TABLE 3 – Coverage rate of the acquired relation relative to nouns in the TV program summaries.

relations and the evaluation results.

The total number of relations was 15,462,746, and 3,458,913 nouns appeared in the relations. We checked how these nouns covered the TV program summaries. We picked 25,769 summaries containing 94,456 nouns whose TV programs were available from NHK on demand. TABLE 3 shows the coverage of the acquired relation.

The acquired relations contained 6% ~ 51% errors (TABLE 2). TABLE 3 indicates the nouns in all relations cover 72.8% of the TV program summaries, but that is overstated because the hyponymy relations acquired by using suffix information target the nouns of the summaries. Relations other than the hyponymy relations by suffix cover 47.7% of nouns of the summaries. Although there is room for improvement in accuracy and coverage, we can acquire a huge number of relations and this holds promise for measuring similarities between summaries.

#### 4 Proposed method for measuring similarities between summaries

Here, we describe the proposed method for measuring the similarity between the TV program summaries in NHK on demand using the acquired relations. The average number of characters in the summaries is about 170, and the average number of nouns included in the summaries is 26. These nouns are used as a clue to measure the similarities. The proposed method first generates graph structures which include the TV programs as their nodes. Then it measures the similarity between summaries by measuring the strength of binding of two TV program nodes on the graph. The following subsections describe each step of measuring similarities.

##### 4.1 Generating graph structures

Graph structures are generated from TV program summaries as follows:

1. The TV program names and each noun appearing in the TV program summaries are put on graph nodes.
2. The TV program node and nodes of summary nouns are connected by non-directional edges. The weight to put on each edge between program node  $p_i$  and noun node  $n_j$  is defined as follows.

$$e(p_i, n_j) = tf(n_j)idf(n_j) / Z_{p_i} \quad (1)$$

Here,  $tf(n_j)$  represents the frequency of noun  $n_j$ ,  $idf(n_j)$  is the inverse document frequency of noun  $n_j$  in the summary of TV program  $p_i$ , and  $Z_{p_i}$  is a normalization factor calculated as the sum of the weights of edges which start from node  $p_i$ .

3. The non-directional edges between noun nodes are constructed from the automatically acquired relations between nouns. If two nouns are related, their nodes are connected. For example, if we acquired the hyponymy relation “*lifestyle-related diseases / hypertension*”, the causality relation “*hypertension / smoking*”, and the material relation “*smoking / Tabaco*”,

the edges “*lifestyle-related diseases ↔ hypertension ↔ smoking ↔ Tabaco*” are constructed. The weights for the edges are defined as

$$e(n_i, n_j) = 1/Z_n \quad (2)$$

Here,  $Z_n$  is the normalization factor calculated as the sum of the weights of edges which start from node  $n$ .

## 4.2 Measuring similarities

The next step measures the similarities between summaries by estimating how much one TV program node is related to another program node on the graph. We use a Markov chain theory, called Green Measures (Kemeny et al. 1966), which is a random walk algorithm to measure the similarity of nodes. This method uses a matrix  $M$  whose element  $m_{ij}$  corresponds to the transition probability from node  $i$  to node  $j$ . Here,  $\sum_j m_{ij} = 1$ . We use the normalized weight for edge defined by equation (1) or (2) for the element  $m_{ij}$ . That is,  $m_{ij} = e(p_i, n_j)$  if there is an edge between program node  $p_i$  and noun node  $n_j$ ,  $m_{ij} = e(n_i, n_j)$  if there is an edge between noun node  $p_i$  and noun node  $n_j$ , and  $m_{ij} = 0$  if there is no edge between node  $i$  and node  $j$ . The Green matrix  $G$  is defined as

$$G = \sum_{t=0}^{\infty} (M^t - M^{\infty}) \quad (3)$$

Here,  $M^t$  corresponds to the transition matrix of  $t$  steps in the random walk. It is known that the Markov chain converges if the chain is both irreducible and aperiodic. Because the chain of our generated graph structure satisfies those conditions,  $M^t$  converges exponentially to  $M^{\infty}$ . The value of element  $g_{ij}$  in  $G$  indicates how much node  $i$  is related to node  $j$ . Using the Green matrix  $G$ , Yann et al. (2007) proposed a score  $S(i, j)$  to indicate the relativeness between two nodes  $i$  and  $j$ :

$$S(i, j) = g_{ij} \log(1/v_j) \quad (4)$$

Here,  $v_j$  is the  $j$ -th element of vector  $v$ , which is a unique invariant probability measure of the matrix  $M$  where  $vM = v$ . Moreover, against any measure  $\mu$  where  $\sum \mu = 1$ ,  $\mu M^n$  converges to  $v$  as  $n \rightarrow \infty$ . The logarithmic term  $\log(1/v_j)$  can modify the  $g_{ij}$  which corresponds to the relativeness score between node  $i$  and node  $j$  when the value  $v_j$  is high. This logarithmic term works like the *idf* value which is used in information retrieval.

We devised two methods for measuring the similarity between two TV program summaries using the relativeness score (eq. (2)) between two nodes. The first method directly uses the relativeness score of the target node directory. The similarity between program nodes  $p_i$  and  $p_j$  is defined as follows.

$$S_{direct}(p_i, p_j) = S(p_i, p_j) \quad (5)$$

This is the relativeness score of node  $p_j$  when the random walk starts from node  $p_i$ .



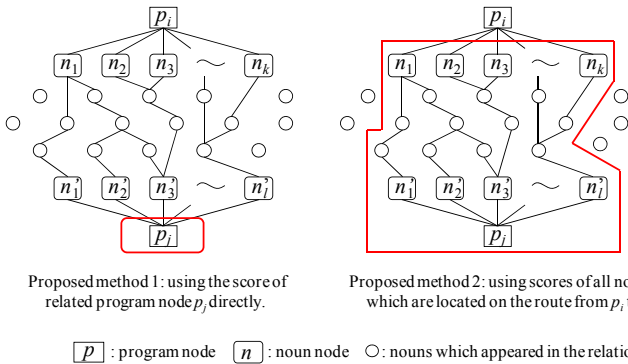


FIGURE 2 – Two methods of measuring similarities between TV program summaries.

The second method uses the scores of all nodes on the route from  $p_i$  to  $p_j$ . The similarity is defined as follows.

$$S_{related}(p_i, p_j) = \sum_{v \in \text{node}(p_i, p_j)} S(p_i, v) \quad (6)$$

Here,  $\text{node}(p_i, p_j)$  is the set of nodes on the route from  $p_i$  to  $p_j$ . FIGURE 2 shows the concept of measuring similarities between the summaries of TV programs by the two methods.

## 5 Experiments

To confirm the effectiveness of the proposed methods, we conducted experiments on measuring the similarities of TV program summaries by using the two proposed methods and four baseline methods. To make the test data, we sampled 352 summaries with the following restrictions.

- All the summaries had different TV program titles
- More than one related TV program were presented for the selected summary in the NHK on-demand service.

The average number of related TV programs to the sampled 352 TV programs was 10.4.

Next, three judges who were not authors ranked the relativeness of the related TV programs against 352 samples. We used Spearman's rank correlation to confirm whether the three judge's rankings were in reasonable agreement. The correlation, which takes into account tie scores of ranks is defined as follows:

$$\rho = T_x + T_y - \sum D^2 / 2\sqrt{T_x T_y} \quad (7)$$

$$T_x = (N^3 - N - \sum_{i=1}^n (t_i^3 - t_i)) / 12$$

$$T_y = (N^3 - N - \sum_{j=1}^n (t_j^3 - t_j)) / 12$$

Here,  $D$  and  $N$  indicate the difference between two ranks and the number of related programs.  $n_x$  and  $n_y$  are the number of tie ranks, and  $t_i$  and  $t_j$  are the ranks of  $n_x$  and  $n_y$ . The average correlation between the ranks by the three judges was 0.565, which indicates moderate agreement. We generated the data by arranging these related programs in descending order of average rank and regarded this manually-generated data as the gold-standard ranking data.

We used his gold-standard ranking data as a reference to evaluate the ranking results.

## 5.1 Baseline methods

### Baseline method 1: Okapi BM25

Goto et al. (2010) proposed a method for measuring the similarity between two summaries of TV programs by using Okapi BM25. Okapi BM25 ranks documents according to the relevance to a given query. They substituted the documents and query with summaries. The similarity score  $S_{BM}(p_1, p_2)$  between two summaries  $p_1$  and  $p_2$  is defined by the following equation.

$$S_{BM}(p_1, p_2) = \sum_{n \in p_1} idf(n) \cdot \frac{tf_{p_2}(n) \cdot (k+1)}{tf_{p_2}(n) + k \cdot (1-b + \frac{|p_2|}{avgdl})} \cdot \frac{(k'+1)tf_{p_1}(n)}{k' + tf_{p_1}(n)} \quad (8)$$

Here,  $tf_p(n)$  is a frequency of noun  $n$  in the summary  $p$ ,  $idf(n)$  is the inverse document frequency of  $n$ ,  $|p|$  is the length of the summary  $p$ ,  $avgdl$  is the average length of the summary, and  $k$  and  $k'$  and  $b$  are parameters for which we used  $k=3.0$ ,  $k'=100.0$ , and  $b=0.75$  from the original literature.

### Baseline method 2: Cosine with nouns appearing in the summary

Each summary is represented by a vector whose elements are nouns appearing in the summary. The weight  $w_{TFIDF}(n)$  for each element is defined by TFIDF.

$$w_{TFIDF}(n) = tf_p(n) \cdot idf(n) \quad (9)$$

The similarity between two summaries  $p_1$  and  $p_2$  is defined by the cosine value of these vectors.

$$S_{TFIDF}(\vec{p}_1, \vec{p}_2) = \frac{\vec{p}_1 \cdot \vec{p}_2}{|\vec{p}_1| |\vec{p}_2|} \quad (10)$$

### Baseline method 3: Cosine with related nouns

This method also represents each summary by a vector. The elements of the vector are composed by nouns appearing in the summary and nouns that are related with any nouns in the summary. For example, if *stroke* appears in a summary, we will add *hypertension* as an element of the vector because *stroke* and *hypertension* have a causality relation. The weight for each noun appearing in the summary is calculated by equation (9). The weight for the expanded noun  $n_{rel}$  is defined as follows.

$$w_{rel}(n_{rel}) = w_{TFIDF}(n) / N_{rel}(n) \quad (11)$$

Here,  $N_{rel}(n)$  is the number of relations of noun  $n$ . The similarity between two summaries is calculated by equation (10).

#### **Baseline method 4 : PLSI**

Baseline method 4 uses a statistical latent class model, called PLSI (Hofmann et al. 1999), which associates an unobserved class variable  $z \in Z = \{z_1, \dots, z_k\}$  with each observation of noun  $w \in W = \{w_1, \dots, w_M\}$  in a document  $d \in D = \{d_1, \dots, d_N\}$ . The distribution of probability  $P(z|d)$  is estimated for each class  $z$  and document  $d$  by using the EM algorithm. The similarity between two summaries is calculated by computing the distance (Jensen-Shannon divergence) between two probability distributions. The Jensen-Shannon divergence between two probabilities,  $P(z|d_1)$  and  $P(z|d_2)$ , can be calculated as follows.

$$D_{JS}(P(z|d_1) \| P(z|d_2)) = \frac{1}{2}(D_{KL}P(z|d_1) \| \frac{P(z|d_1)+P(z|d_2)}{2}) + D_{KL}P(z|d_2) \| \frac{P(z|d_1)+P(z|d_2)}{2}) \quad (12)$$

Here,  $D_{KL}$  indicates the Kullback-Leibler divergence:

$$D_{KL}(P(z|d_1) \| P(z|d_2)) = \sum P(z|d_1) \log \frac{P(z|d_1)}{P(z|d_2)} \quad (13)$$

By using the latent class, it becomes possible to put a non-zero similarity value on summaries that express similar meanings even if they do not share words.

## **5.2 Experimental results**

Targeting the selected summaries for 352 TV programs, we conducted an experiment on ranking related TV programs using the proposed methods and four baseline methods. TABLE 4 shows Spearman's rank correlation with the gold-standard data. In baseline method 4, we tried the following parameter values for the number of unobserved classes  $z$  and the temperature parameter  $\beta$  in the process of the EM algorithm and selected  $z=200$  and  $\beta=0.75$ , which gave the best result.

Methods	Rank correlation
Baseline 1 (Okapi-BM25)	0.370
Baseline 2 (cosine with nouns appearing in the summary)	0.350
Baseline 3 (cosine with related words)	0.371
Baseline 4 (PLSI)	0.190
Proposed 1 (using the score of related program node)	0.351
Proposed 2 (using the scores of all nodes on the route to the related program node)	<u>0.423</u>

TABLE 4 – Evaluation result for each method.

$$z = \{10, 50, 100, 200, 300, 400, 500, 1000\}$$

$$\beta = \{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0\}$$

The results for proposed method 2, which uses the scores of all nodes on the route between the program nodes, were far better than those of the other methods (TABLE 4). On the other hand, proposed method 1 was not better than baseline methods 1 ~ 3. The score of the node which is connected directly with the start node of the random walk is much larger than one of the nodes which are connected indirectly with the start node. The scores for the related program nodes, which are far from the target program node, are too small to compare with each other. This arises from the shortage of noun relations. If we can acquire enough noun relations, we could avoid this problem. The correlation of baseline method 4 which uses PLSI is much lower than the other methods. This is because the results of the clustering by PLSI were not useful for making TV program recommendations. For example, the words *politics*, *economics*, and *international* belong

[Target TV program title] Marutoku-magazine, exercise ~ hip joint and legs

[Nouns appearing in the corresponding summary] hip joint, stretch, movement, you, easily, range of movement, up, posture, around hip joint, five minutes, velocity, length of stride, exercise, muscle, coverage, one day

Related TV program titles and nouns appearing in the corresponding summary	Gold-standard data (score)	Proposed method 2 (score)	Baseline method 2 (score)
Marutoku-magazine, exercise ~ arm and back [Nouns] back, body, arm, five minutes, condition, tension, everyday, you, head, exercise, posture, muscle, one day	1 (1.333)	1 (0.913)	1 (0.328)
Tameshite-Gatten, Banana revolution ~ declaration of new ingredient [Nouns] banana, fruit, vegetable, cooking, taste, exposure, nourishment, full marks, hand, product, No. 1 of consumption, clear, world, shipping, I, easy, expensive ingredient, ability, ingredient, Gatten's way of cooking bananas, majority, method	2 (1.667)	2 (0.582)	2 (0.0)
Fudangi-no-onsen, Aomori Shimofuro hot spring [Nouns] therapeutic bath, body, bitter cold, Muromachi era, scene, strait, role, people, two, shared hot spring, fishing herring, before World War II, large spa, Yasushi Inoue, place, home town, fist, exchange, information, fisherman, friend, Tsugaru Straits, novel, variety, famous, importance, Shimofuro hot spring, hot spring, core, new hot water	3 (3.333)	3 (0.561)	2 (0.0)
Asaichi, Japan navigation ~ Kyoto [Nouns] Kyoto, Daihachi car, talk, huge, Daikaku temple, autumn, surface of water, Osawa pond, travel, kimono, Maho, vegetable farmer, autumn leaves, fantastic, together, meeting, vegetarian dish, popular, Kyoto vegetable, lighting-up, scene, Hisako Noguchi, 82 years old, temple, Sagano, actor, finding antique kimono, full of autumn leaves, 13th, airiness, world, sight to see, temple master, ancient city	4 (3.667)	4 (0.203)	2 (0.0)

TABLE 5 – Ranking results of proposed method 2 and baseline method 2. All titles and nouns are translated from Japanese nouns.

to the same class, which resulted in miss-selection of the related program for *politics*. The clustering based approach sometimes is affected by from the granularity of each cluster.

TABLE 5 shows a sample of the rankings of proposed method 2 and baseline method 2. The scores of baseline method 2 were zero for three programs, whose summaries did not share words with that of the target program. In contrast, proposed method 2 properly scored these programs. This caused its results to have a higher correlation with the gold-standard data.

### 5.3 Effectiveness of the using relations

We randomly sampled from all acquired relations in order to investigate the effectiveness of the relations. FIGURE 3 shows the correlation versus the number of sampled relations. Here, the correlation is 0.400 when the number of relations is zero. This value is higher than those of the baseline methods. In the case that we do not use any relations, the edges in the graph structure are composed of one between the program node and the noun node appearing in the program summary. This arises from a heuristic that “the words in the same program summary are similar to each other”. This result shows that the more relations we used, the higher the correlation became. We used about 15.5 million relations in this experiment, and we hope we can get higher correlations if we use more relations.

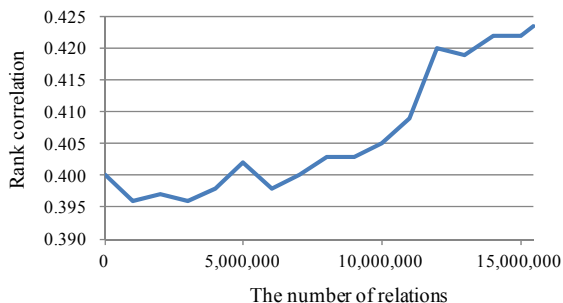


FIGURE 3 – The correlation depended on the number of sampled relations.

### 5.4 Considerations regarding the edge weighting

In the process of generating graph structures, each edge between the program node and the noun node is weighted by the TFIDF value in equation (1). A random walk tends to move to the node whose TFIDF value is high and which is considered important. However, if a noun node has links to the other nouns appearing in the same summary, the random walk will move on the noun node with higher probability because there are several routes to get to the node.

We experimented with ranking related TV programs by using proposed method 2 without an edge weighting by TFIDF. As a result, the rank correlation value was 0.427, which is comparable to the result of using the weighting. This means that the graph structure implicitly defines the node’s importance.

## Conclusion

We proposed a method for measuring the similarity between two TV program summaries. The method generates a graph structure whose nodes are composed of TV programs and nouns appearing in the corresponding summary and whose edges are generated from four kinds of relations automatically acquired from the Web and Wikipedia. The similarity between two TV program summaries is calculated on the basis of the relativeness of the two TV program's nodes in the graph structure by using a random walk algorithm. Experiments confirmed that our method provided a proper similarity measure that showed a better correlation with the gold-standard data than the baseline approaches. The experiments using several relations indicated that we would get a higher correlation if we used more relations. Furthermore, we confirmed that we did not need to use the edge weighting by TFIDF because the graph structure implicitly defines the importance for each node.

We used four types of relations in the experiments. Enlarging the number of relations will be of further help in measuring the similarity. Moreover, the relations in the experiments contained 6% ~ 51% errors and covered 47.7% of the nouns appearing in the TV program summaries. In the future, we will determine whether using more relations with higher accuracy and broader coverage.

## References

- Bond, F., Baldwin, T., Fothergill, R. and Uchimoto, K. (2012). Japanese SemCor: A Sense-tagged Corpus of Japanese, In *Proceedings of the 6th International Conference of the Global WordNet Association (GWC)*.
- De Saeger, S., Torisawa, K., Kazama, J., Kuroda, K. and Murata, M. (2009). Large Scale Relation Acquisition using Class Dependent Patterns, In *Proceedings of the IEEE International Conference on Data Mining (ICDM'09)*, pp.764-769.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, 391-407.
- Goto, J., Sumiyoshi, H., Miyazaki, M., Tanaka, H., Shibata, M. and Aizawa, A. (2010). Relevant TV Program Retrieval using Broadcast Summaries, In *Proceedings of ACM on Intelligent User Interfaces(IUI)*, pp.411-412.
- Hofmann, T. (1999). Probabilistic Latent Semantic Indexing, In *Proceedings of SIGIR'99*, ACM Press, pp.50-57.
- Kazama, J. and Torisawa, K. (2008). Inducing Gazetteers for Named Entity Recognition by Large-scale Clustering of Dependency Relations, In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pp. 407-415.
- Kemeny, J. G., Snell, J. L. and Knapp, A.W. (1966). Denumerable Markov chains, Van Nostrand Company.
- Koren, Y., Bell, R. and Volinsky, C. (2009) Matrix Factorization Techniques for Recommender

Systems, *IEEE Computer*, pp.42-49.

Kuroda, K., Murata M. and Torisawa, K. (2009). When nouns need co-arguments: A case study of semantically unsaturated nouns, In *Proceedings of the 5th International Conference on Generative Approaches the Lexicon 2009*.

Linden, G., Smith, B. and York, J. (2003). Amazon.com recommendations, *IEEE Internet Comput.*, vol7, no.1, pp. 76-80.

Melville, P. and Sindhvani, V. (2010). Recommender Systems, *Encyclopaedia of Machine Learning*, Springer.

Oku, k., Nakajima, S., Miyazaki, J. and Uemura, S. (2007). Context-Aware Recommendation System Based on Context-Dependent User Preference Modeling, *IPSJ journal*, vol. 48, SIG11 (TOD\_34), pp. 162-176. (in Japanese)

Robertson, S. and Walker, S. (1999). Okapi/ Keenbow at TREC-8, In *Proceedings of TREC-8*, pp151-162.

Sumida, A., Yoshinaga. N. and Torisawa, K. (2008). Boosting Precision and Recall of Hyponymy Relation Acquisition from Hierarchical Layouts in Wikipedia, In *Proceedings of the sixth Language Resources and Evaluation Conference (LREC 2008)*, pp.2462-2469.

Yamada, I., Hashimoto, C., Oh, J.-H., Torisawa, K., Kuroda, K., De Saeger, S., Tsuchida, M. and Kazama, J (2010). Generating Information-Rich Taxonomy from Wikipedia, In *Proceedings of the 4th International Universal Communication Symposium (IUCS 2010)*, pp. 96-103.

Yann, O. and Pierre S. (2007). Finding Related Pages Using Green Measures: An Illustration with Wikipedia, In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pp.1427-1433.

