FEATURE ARTICLE: Data Analytics

Linkify: Enhancing Text Reading Experience by Detecting and Linking Helpful Entities to Users

Ikuya Yamada
Studio Ousia, RIKEN AIP

Tomotaka Ito
Studio Ousia

Hideaki Takeda
National Institute of Informatics

Yoshiyasu Takefuji
Keio University

We frequently encounter unfamiliar entity names (e.g., a person's name or a geographic location) while reading texts such as newspapers, magazines, and web pages. When this occurs, we typically perform a sequence of tedious actions: select the entity name, submit it to a search engine, and obtain detailed information from websites. In this paper, we present Linkify, a tool that enhances text reading by automatically converting entity names into links and displaying a widget that contains links to several relevant websites. We also propose a novel method for evaluating the helpfulness of entities to users using supervised machine learning with a set of carefully designed features. Experimental results show that our method significantly outperforms existing state-of-the-art methods.

In our daily lives, while reading the text in newspapers, magazines, and web pages, we frequently encounter unfamiliar terms (e.g., a person's name or a geographic location). In such cases, we typically obtain related information from relevant web pages using a search engine. However, this process requires several time-consuming steps: selecting the text, submitting a query to a search engine, and obtaining detailed information from websites.

Entity linking (EL) is the task of linking textual entity mentions in a text to entries in a knowledge base (KB) (e.g., Wikipedia) that contains relevant information regarding the entities. The main difficulty in EL is ambiguity in the meaning of entity mentions. For example, the mention Washington in a

Published by the IEEE Computer Society
1541-1672/18/$33.00 ©2018 IEEE

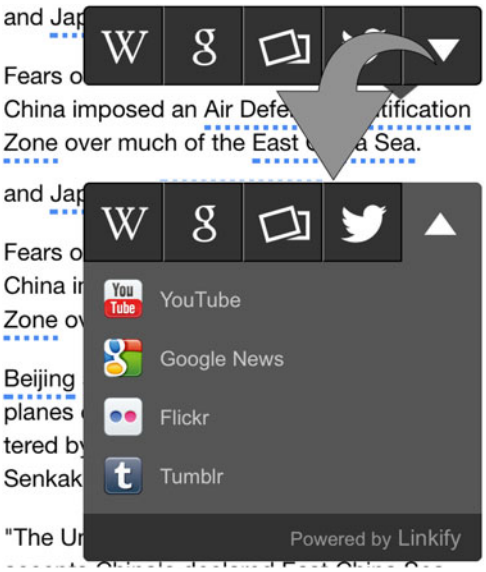

Figure 1. When the user clicks a link, a small widget is displayed. The widget appears small initially and can be expanded to show further links.

document can refer to various entities, such as the state, or the capital of the US, the actor Denzel Washington, the first US president George Washington, etc.

In this paper, we propose a novel tool called Linkify that automatically enriches a document by converting entity mentions within the document into links using EL and displays a widget containing relevant links of the entity when a user selects the link (see Figure 1). Using Linkify, users can retrieve desired information about an unfamiliar entity instantly by clicking the link, instead of having to select the text and submit it to a search engine. As in previous studies, we use Wikipedia as the KB.

One key problem is that Wikipedia contains many entities that are rarely helpful to users. For example, it even contains A, I, and You. Furthermore, general entities, such as Japan, are rarely helpful to users when compared to specific entities, such as Nara Prefecture. Excessive linking can be distracting and typically degrades a user's overall experience. Therefore, we need to ensure the quality of links when we present detected entities to users directly. For example, Wikipedia instructs its contributors to insert links only where they are relevant and helpful in the manual for its contributor (http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style). Unfortunately, this problem has received little attention, since most of the existing EL methods mainly focus on their use as a component of other systems rather than on direct usage by users.

In order to address this problem, we propose a method for evaluating the helpfulness to users of entities detected by EL systems. We define this task as a postprocessing step of EL. We use a supervised machine learning model with a carefully designed set of features to classify whether an entity is likely to be helpful to users. We evaluated the proposed method with a dataset that we developed using a crowd sourcing service. The experimental results show that our method significantly outperformed baseline methods.

The proposed system is available publicly at http://linkify.mobi.

RELATED WORK

EL has lately been extensively studied,[1–4] and used as a fundamental component in various natural language tasks, such as information extraction and semantic search.[5] However, typical EL methods are targeted at obtaining higher recall; in other words, the goal is to extract every entity in the text. Although this approach is ideal when it comes to using EL as a component of other systems, it is less effective for direct consumption by users. Because excessive linking of entities reduces the readability of a document, an additional pruning step is typically required when displaying linked entities directly to users. Two recent studies have been reported to address this problem. The first example is the work of Brzeski *et al*.,[6] who proposed a model to estimate the helpfulness (or interestingness) of mentions using a conventional vector space model based on query logs obtained from the Yahoo! search engine. Similarly, Gao *et al*.[7] proposed a model to measure the interestingness of entity mentions using deep neural networks trained with the massive click logs obtained from a commercial web browser. These methods differ significantly from the one presented here in terms of their dependence on proprietary user logs obtained from commercial services. Our proposed method depends only on publicly available data, thus the experimental results are easily reproducible.

ARCHITECTURE

Figure 2 shows the architecture of Linkify. Linkify can be easily integrated into any web pages by adding a small-sized script coded in JavaScript. The system has the following three steps: when the browser finishes rendering a web page,

(1) the DOM iterator sends the page texts to the server-side system,
(2) the server-side system processes the page text and returns the target entity mentions, and
(3) the link converter converts the mentions in the web page into links.

Then, when the user clicks on the entity name, a small widget is displayed (see Figure 1). The widget contains relevant links leading to Wikipedia, Google, Twitter, and so forth. The Wikipedia link points to its corresponding page of the entity whereas the other links point to the search pages of the other services.

The main components of the server-side system are the entity linker and the helpfulness evaluator, which are described in detail in the remainder of this paper.

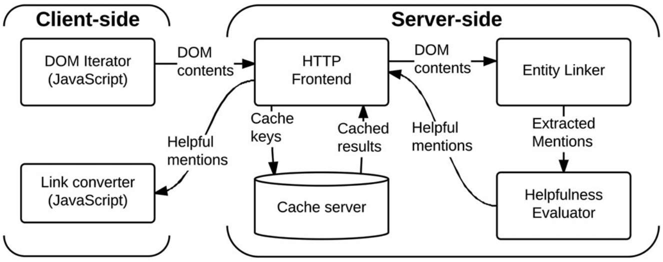

Figure 2. Architecture of the proposed system.

LINKING HELPFUL ENTITY MENTIONS

This section describes our approach that aims to extract entity mentions users are likely to find helpful. Given a document, our goal is to detect a set of entity mentions that are helpful to users. We address this task as a three-step approach consisting of candidate generation, mention disambiguation, and helpfulness evaluation. The first two steps are those that have typically been used to address EL in the past EL literature. The key contribution of this paper is the last step in which our novel method based on supervised machine learning is used with the aim of filtering out entity mentions that are rarely helpful to users.

Before starting the description of our method, we introduce three measures that have frequently been observed in past EL studies. The prior probability of mention m referring to entity e is the probability of Wikipedia anchors with the same surface as m pointing to e, the link probability of m is the probability that the surface text of m appears as an anchor in Wikipedia, and the entity prior of e is defined as log($|A_e|$ + 1), where A_e is the set of KB articles that contain a link to e. These measures were collected directly from the Wikipedia dump described in Section "Knowledge Base".

Candidate Generation

The candidate generation task aims to generate a set of candidate entity mentions with the set of possible referent entities. The system takes all the word n-grams in a document, looks up each n-gram in the mention-entity dictionary described below, and treats an n-gram as a candidate mention if it exists in the dictionary. The output of this step is a list of candidates each of which consists of the mention m and the set of possible referent entities $\{e_1, e_2, \ldots, e_K\}$. For computational efficiency, we sort the referent entities according to the prior probability and use only the top 30 results.

The mention-entity dictionary maps a mention surface (e.g., apple) to the possible referent entities (e.g., Apple Inc., Apple (food)). Here, the possible mention surfaces of an entity are extracted from the following three sources.

(1) The corresponding Wikipedia page title.
(2) The page titles of the Wikipedia pages that redirect to the page of the entity.
(3) Anchor names in Wikipedia articles that point to the page of the entity.

Mention Disambiguation

Given a candidate detected by the previous step, the mention detection task aims to select the referent entity from the set of possible referent entities. Similar to past work,[8–10] we adopt a supervised machine learning model to address this task. In particular, we use random forest (RF) because of its superior performance in a past EL study.[8] Given a candidate (i.e., a mention and possible referent entities), we train the RF model to assign a relevance score to each entity that represents whether the mention refers to the entity. Then, we exclude candidates with relevance scores lower than a threshold σ, and to select the best set of nonoverlapping entity mentions by processing the mentions in the descending order of the relevance scores.

Furthermore, we use machine learning features that are frequently used in the past EL literature. Our features can be grouped into base, *string similarity, and topical coherence.*

Base: As base features, we mainly use the features proposed by Guo *et al*.[9] In particular, we include the link probability, the prior probability, the number of tokens in the mention, and the number of KB anchors that have the same surface as the mention and point to the entity. We also include the entity prior that has commonly been used as a feature in past EL research.

String Similarity: Following,[8,10] we also include the string similarity between the title of the entity and the surface of the mention. We use the edit distance, whether the title of the entity exactly equals or contains the surface of the mention, and whether the title of the entity starts or ends with the surface of the mention.

Topical Coherence: The topical coherence feature represents the extent to which an entity e is related to the topics of the document d. This can be estimated by measuring how e is related to other entities in d. The Wikipedia link-based measure (WLM)[11] is used to measure the semantic relatedness between two entities. This measure is defined as

$$\mathrm{WLM}(e_1, e_2) = 1 - \frac{\log\ \max\left(|A_{e_1}|, |A_{e_2}|\right) - \log|A_{e_1} \cap A_{e_2}|}{\log|KB| - \log\ \min\left(|A_{e_1}|, |A_{e_2}|\right)}$$

where $|KB|$ is the total number of articles in KB and A_e refers to the KB articles that have a link to e. The topical coherence is calculated by averaging the relatedness between e and all the other unambiguous entities in d. Additionally, averaging the relatedness score has been a common practice to model the topical coherence in past EL studies.[1,3,9,12]

Helpfulness Evaluation

Given a mention and its referent entity detected by the mention disambiguation task, the helpfulness evaluation task aims to predict whether linking the mention is likely to be helpful to users and to exclude the mention if it is unlikely to be helpful. We address this task using a supervised machine learning model. Here, we also use the binary RF classifier because its performance was superior to that of the other algorithms (C4.5, SVM, and AdaBoost) in our preliminary experiments. Note that, the training of this model does not depend on the prior EL steps; we simply use each entity mention in the dataset as a training instance of our machine learning model.

Regarding the features of this model, we carefully designed a feature set to evaluate the extent to which users would find the entity mentions helpful. The features can be grouped into four groups: link probability, entity popularity, entity class, and topical coherence. The basic concept of the design of our feature set is to combine the three kinds of relevant signals such as the human accumulated decisions of whether to link a mention (e.g., link probability), the knowledge of the entity (i.e., entity popularity and entity class), and the semantic relevance between the entity and the document (i.e., topical coherence). The details of our features are described as follows.

Link Probability: We first include the link probability to our feature set. The intuition behind using them as features is that they represent the past decisions made by Wikipedia contributors as to whether the mention should be created as a link.

Entity Popularity: Entity popularity features consist of two features, namely the entity prior to the entity, and the average number of page views of the corresponding Wikipedia article of the entity. The former feature represents the popularity of the entity in the Wikipedia articles, and the latter represents the popularity among Wikipedia readers.

Entity Class: Entity class features are obtained from DBpedia (http://wiki.dbpedia.org/). In DBpedia, an entity is assigned one or multiple entity classes (e.g., Actor, Company). We use each entity class of DBpedia as a feature. Note that, because the entities contained in DBpedia are derived directly from Wikipedia, DBpedia entities generally have corresponding entities in Wikipedia.

Topical Coherence: The topical coherence feature, described in the previous section, is also used as a feature of this task. The motivation for adding this feature is based on our simple assumption that entities that are semantically relevant to the topics of the document are likely to attract interest compared to those that would not.

Experimental Setup

In this section, we detail our experimental setup.

Knowledge Base

We use the February 2016 English version of the Wikipedia dump as the KB, and the Wikipedia page view data from January 1, 2016, to March 1, 2016. We also use the April 2015 version of the DBpedia datasets to obtain entity classes.

Dataset

We trained and evaluated our method by developing a dataset based on the IITB dataset,[2] which is a popular dataset that was used in past EL studies. The dataset contains annotations of entity mentions in news articles obtained from various news sites. We add a label to each entity mention to indicate whether the linking entity mention is likely to be helpful to users.

Here, the problem is that an annotation can be biased by the annotators' subjective judgment. In order to avoid this as much as possible, we used various anonymous online annotators recruited through a crowd-sourcing service (i.e., Amazon Mechanical Turk). We posed the following question to the annotators:

> *Do you think converting the keyword into a link is sufficiently helpful to readers of this document and could it improve the readers' overall user experience?*

The annotators were required to decide whether linking the entity mention would be helpful to users. We explained to the annotators that this dataset is intended to be used for improving the user's reading experience by linking entities, and we asked annotators to imagine themselves as actual readers of the document.

We assigned each annotation task to three individual annotators. As a result, we obtained 33 567 annotations consisting of 10 775 positive annotations and 22 792 negative annotations from 60 individual annotators. The annotators processed an average of 559 tasks and a maximum of 5793 tasks, and 72.8% of annotations received the same binary judgments from all three annotators. Additionally, we publicized the dataset at http://www.github.com/studio-ousia/el-helpfulness-dataset for further studies.

We derived the ground-truth binary labels by counting the votes for the three corresponding answers. Furthermore, we randomly selected 73 documents as the training set, 10 documents as the development set, and the remaining 20 documents as the test set. We used the training set for training the machine learning model, the development set for tuning parameters, and the test set for evaluating the performance.

Evaluation Criteria

In this experiment, we used precision, recall, and F1 as performance metrics, all of which were typically employed in past EL studies. F1, which is defined as

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

was used as our primary metric. Note that, we treat an entity mention as being correct only when the mention is linked to the correct entity and also when the mention is annotated as helpful in the dataset.

Hyper-Parameters

As mentioned above, we used an RF in the mention disambiguation and the helpfulness evaluation models. RF can be parameterized by (1) the number of decision trees contained in the model (n_trees) and (2) the maximum depth of the decision tree (max_depth). We tuned the hyper-parameters of RF by using the F1 score on the development set and we used n_trees = 1000 and max_depth = 10 for the mention disambiguation task, and n_trees = 1000 and max_depth = 20 for the helpfulness evaluation task. For the other hyper-parameters, we followed the RF implementation in the Python scikit-learn library. Furthermore, we experimentally set the threshold σ of the mention detection task to 0.3.

Baselines

We compared our system with existing systems by using the following three state-of-the-art EL systems as the baselines of our system.

- *DBpedia Spotlight*[13] is an open-source EL system that extracts candidate entity mentions using a predefined dictionary and resolves the detected mentions to their corresponding entities using a context model based on a vector space model.
- *TAGME*[12] is an EL system with a publicly accessible web API. The system detects candidate entity mentions using a dictionary and disambiguates these mentions using a novel coherence model that is designed to work well for short texts.
- *Illinois Wikifier*[3,14] is an open-source EL system that first detects the entity mentions using its named entity recognition system and disambiguates the detected mentions using a supervised machine learning model with the textual and the coherence features.

Additionally, we used the default parameters for all baseline systems.

Experimental Results

In this section, we present our experimental results. Table 1 shows our main experimental results with the results of our competitors. We tested our method as both a full system and a system without our

Table 1. Comparison of our method with the state-of-the-art methods.

	Precision	Recall	F1
Our Method (Full)	**0.59**	**0.61**	**0.60**
Our Method (EL only)	0.37	0.68	0.48
Illinois Wikifier	0.24	0.69	0.35
DBpedia Spotlight	0.40	0.60	0.48
TAGME	0.10	0.63	0.17

Table 2. Results of our feature study.

	Precision	Recall	F1
Entity linker:			
Base	0.34	0.66	0.45
+ String Similarity	0.35	0.67	0.46
+ Coherence	0.37	0.69	0.48
Helpfulness evaluator:			
Link probability	0.54	0.52	0.53
+ Entity popularity	0.56	0.55	0.55
+ Entity class	0.58	0.60	0.59
+ Topical coherence	0.59	0.61	0.60

helpfulness evaluator (EL only). As a result, our full system clearly outperformed all the baseline systems in terms of F1 scores. The baseline systems typically performed well in recall, but worse in precision, because they extracted many unhelpful entity mentions. Furthermore, compared to our system without the helpfulness evaluator, our full system improved the precision significantly by sacrificing only a small amount of recall. This clearly demonstrates the effectiveness of our helpfulness evaluator model for improving the quality of linking helpful entity mentions. Additionally, the performance of our EL method (i.e., candidate generation and mention disambiguation) also proved to be competitive with that of the state-of-the-art methods.

Next, we conducted a feature study to evaluate the effectiveness of the proposed features by iteratively adding each feature group to our model (see Table 2). As can be seen in the table, all features consistently contributed to the performance.

Error Analysis

In order to investigate the errors made by our proposed system, we conducted an error analysis on our test set. As a result, we observed that the 227 errors were caused by the mention disambiguation step. In particular, 76 mentions were not detected, 119 mentions were wrongly detected, and 32 mentions were resolved into the wrong entities.

Furthermore, the helpfulness evaluator erred on 210 mentions. 135 unhelpful mentions were wrongly detected as helpful, whereas 75 helpful mentions were classified as unhelpful. Moreover, we inspected the errors of the helpfulness evaluator and found that our method generally performed more accurately for mentions referring to named entities (e.g., person, location, and organization), and less accurately for those referring to general entities (e.g., virus, carbon, DVD). We considered that because mentions referring to general entities had relatively low link probabilities and a small number of entity classes compared to mentions referring to named entities, our method neglected to decide whether they were helpful.

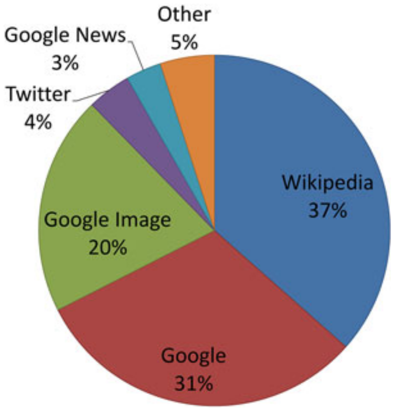

Figure 3. Distribution at external services selected by user.

ANALYSIS OF USER CLICK LOGS

We finally present two empirical results based on user click logs retrieved from our service. We randomly sampled 10 000 user clicks on the entity links and analyze them from two points of view.

We first investigated which service in the widget is likely to be selected by users. As shown in Figure 1, we display eight external services on the widget: Wikipedia, Google, Google Image, Twitter, YouTube, Google News, Flickr, and Tumblr. Figure 3 summarizes the distribution of these services. The most popular service is Wikipedia, followed by Google. This demonstrates the advantage of EL in that it enables direct navigation to the corresponding Wikipedia page by a single click. The results showed that Wikipedia and Google accounted for approximately 68% of the total selections.

Next, we examined what type of entity is likely to be selected by users. For each click log, we retrieved the types of the entity using DBpedia Ontology Classes (http://mappings.dbpedia.org/server/ontology/classes/) and counted the frequencies. Table 3 lists the top 10 common entity types observed in the click logs. (Note that entity types that are direct children of Thing, such as Agent, Work, and Place, have been eliminated). As seen in the table, a variety of entity types received attention from users. In particular, we observed that specific entity types such as Artist, Actor, and Company are more likely to be selected than other entity types.

Table 3. Top 10 common entity types selected by user with frequencies and ratios.

DBpedia Class	Count	Ratio
Person	1928	21.9%
Artist	984	11.2%
Organisation	759	8.6%
Actor	699	8.0%
Company	538	6.1%
Software	476	5.4%
Populated-Place	352	4.0%
City	349	4.0%
Model	307	3.5%
AdultActor	213	2.4%

CONCLUSION AND FUTURE WORK

In this paper, we proposed a tool named Linkify. The tool demonstrates a novel use-case of EL that helps readers to automatically enrich a document by linking entities using EL. We also proposed a novel method for accurately evaluating the helpfulness of entities in a document. We demonstrated that this task can be addressed accurately by using supervised machine learning with a carefully designed set of features.

In future, we plan to improve the performance of our proposed model using state-of-the-art context models designed to address EL such as those of Globerson *et al.*[15] and Yamada *et al.*[10] Moreover, we are also interested in linking entity mentions of which the referent entities do not exist in Wikipedia by leveraging domain-specific open repositories such as Crunchbase (https://www.crunchbase.com) and MusicBrainz (https://musicbrainz.org).

REFERENCES

1. D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proc. 17th ACM Conf. Inf. Knowl. Manage*., 2008, pp. 509–518.
2. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 457–466.
3. L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol. – Vol. 1*, 2011, pp. 1375–1384.
4. X. Ling, S. Singh, and D. S. Weld, "Design challenges for entity linking," *Trans. Assoc. Comput. Linguistics*, vol. 33, pp. 315–328, 2015.
5. R. Blanco, G. Ottaviano, and E. Meij, "Fast and space-efficient entity linking for queries," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 179–188.
6. V. von Brzeski, U. Irmak, and R. Kraft, "Leveraging context in user-centric entity detection systems," in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, 2007, pp. 691–700.
7. J. Gao, P. Pantel, M. Gamon, X. He, and L. Deng, "Modeling interestingness with deep neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 2–13.
8. E. Meij, W. Weerkamp, and M. de Rijke, "Adding semantics to microblog posts," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 563–572.
9. S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 1020–1030.
10. I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn*., 2016, pp. 250–259.
11. D. Milne and I. H. Witten, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *Proc. 1st AAAI Workshop Wikipedia Artif. Intell.*, 2008, pp. 25–30.
12. P. Ferragina and U. Scaiella, "Fast and accurate annotation of short texts with wikipedia pages," *IEEE Softw.*, vol 29 no. 50, pp. 70–75, Jan./Feb. 2012.
13. P. N. Mendes, M. Jakob, A. Garcốa-Silva, and C. Bizer, "DBpedia Spotlight: Shedding light on the web of documents," in *Proc. Proc. 7th Int. Conf. Semantic Syst.*. Graz, Austria, Sep. 2011, pp. 1–8.
14. X. Cheng and D. Roth, "Relational inference for wikification," in *Proc. Conf. Empirical Methods Natural Lang. Process*., 2013, pp. 1787–1796.
15. A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringaard, and F. Pereira, "Collective entity resolution with multi-focal attention," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 621–631.

ABOUT THE AUTHORS

Ikuya Yamada is a Co-Founder and CTO with Studio Ousia Inc, Tokyo, Japan. His technical interests include natural language processing and machine learning. He received the Ph.D. degree from Keio University, Tokyo, Japan, in 2016. His postal address is 4F Otemachi Building, 1-6-1 Otemachi Chiyoda-ku, Tokyo, 100-0004, Japan.

Tomotaka Ito is a Software Engineer with Studio Ousia Inc, Tokyo, Japan. His technical interests include machine learning and human–computer interaction. He received the M.S. degree in media and governance from Keio University, Tokyo, Japan. His postal address is 4F Otemachi Building, 1-6-1 Otemachi Chiyoda-ku, Tokyo, 100-0004, Japan.

Hideaki Takeda is a Professor with the National Institute of Informatics, Tokyo, Japan. His research interests include knowledge-based systems, the semantic web, and web informatics. He received the Ph.D. degree in engineering from the University of Tokyo, Tokyo, Japan, in 1991. His postal address is 2-1-2, Hitotshubashi, Chiyoda-ku, Tokyo, 101-8430, Japan.

Yoshiyasu Takefuji is a Professor with Keio University, Tokyo, Japan. His research interests include cyber security, neural computing, energy harvesting, the internet of things, and artificial intelligence. He received the Ph.D. degree from Keio University, in 1983. His postal address is 5322 Endo, Fujisawa, Kanagawa, 252-0882, Japan.